2nd MATHMOD
VIENNA

2nd MATHMOD
VIENNA

2nd MATHMOD
VIENNA

2nd MATHMOD
VIENNA

PROCEEDINGS

IMACS Symposium on
MATHEMATICAL MODELLING
February 5-7, 1997
Technical University Vienna, Austria

I. Troch
F. Breitenecker
editors

ARGESIM Report 15

ASIM Mitteilung 57B

ISBN print: 978-3-901608-11-7
ISBN e-book: 978-3-901608-09-4

ISBN 3-901608-11-7

ARGESIM Report No. 15

ARGE Simulation News (ARGESIM)
c/o Technical University Vienna
Wiedner Hauptstraße 8-10
A-1040 Vienna, Austria
Tel: +43-1-58801 5386, 5374, 5484
Fax: +43-1-5874211
Email: argesim@argesim.tuwien.ac.at
WWW: <URL:http://argesim.tuwien.ac.at>

Printed by *CA Druckerei*, Vienna, Austria

ARGESIM Report 15

ASIM Mitteilung 57B

ISBN print: 978-3-901608-11-7
ISBN e-book: 978-3-901608-09-4

Preface

The possibility to solve a certain problem and the quality of a solution of a certain task depend essentially on appropriate modelling of the question and of all available information. In some cases, the system under investigation and its behaviour are understood rather well. In such cases an appropriate model will assist in finding a good solution of the problem to be solved. In other situations such a model is primarily intended to help for a better understanding of what is going on in the system. Examples for the first case are many types of design problems being encounterd in typical engineering sytems, such as controller design, design of a production line etc. whereas the request for an improved understanding is often found in connections with non-engineering systems such as biological or medical systems, economic or environmental systems and their control etc.

There is a rather wide consensus that mathematical modelling i. e. abstraction and formalization, is of intrinsic importance. Moreover, most engineers and scientists know quite well that appropriate modelling is far from being easy and that the quality of a design depends strongly on the quality of the model. One of the most important challenges connected with proper modelling is the request to model indeed the given task i. e. all relevant information, restrictions, demands etc. In control engineering not only a model of the plant and constraints on relevant physical variables must be put in a mathematical form but also other requests such as that the resulting mathematical control law must allow for implementation witb a certain type of equipment etc.

By now, considerations such as these are accepted by practically all people involved in solving problems by using mathematical methods, no matter whether they work at a scientific institution or in an industrial environment.

However, the area of application determines to a certain extent the knowledge of basic modelling principles, preferences of modelling approaches, of methods for model simplification or for parameter estimation etc. Moreover, many things are discovered repeatedly. Therefore, a conference having mathematical modelling as its center will allow for a fruitful and stimulating exchange of ideas. Consequently, the second IMACS symposium on Mathematical Modelling (2nd MATHMOD) is devoted to the mathematical (or formal) modelling of all type of systems no matter whether the system is

* dynamic or static
* lumped parameter or distributed parameter
* deterministic or stochastic
* linear or nonlinear
* continuous or discrete
* or of any other nature.

Consequently, a wide variety of formal models is to be discussed and the term "mathematical model" includes classical models such as differential or difference equations, Markov processes, ARMA models as well as more recent approaches such as bond graphs or Petri nets.

The written versions of the contributions to 2nd MATHMOD Vienna are collected in these Proceedings starting with the manuscripts of the invited lectures. The first survey to be presented deals with recent algorithms for the generation of good random numbers needed for certain modelling approaches. Such numbers will be of interest especially when prediction of a systems behaviour – usually called simulation – is the main goal and when the practitioner has to decide which random number generator will suit his needs best. A survey follows on hybrid systems such as arise in technical systems and where continuous and discrete event dynamics interact. The third invited lecture is concerned with qualitative modelling and surveys the principle lines of current research and explains the main ideas of an automata-theoretic approach being successfully used in supervisory control.

Then follow contributed papers – which were selected for presentation by a reviewing process based on submitted extended abstracts in which the members of the IPC took active part together with groups of papers contributed upon invitation of a session organizer. The volume concludes with the abstracts of posters being on display during the conference. All these contributions were colleted and arranged in sessions according to their main thematic point:

* Fuzzy and Qualitative Modelling
* Automation of Modelling and Bond Graphs
* Petri Nets and Discrete Event Modelling
* Identification
* Software Tools
* Modelling in Practice
* General Engineering Applications
* Traffic Modelling
* Electrical Systems
* Mechanics and Mechatronics, incl. Robotics
* Automatic Control
* Physical Applications
* Environmental Systems
* Biology and Biotechnical Engineering
* Economic and Social Systems
* Theoretic Aspects

Such a grouping is by no means easy because many contributions address several different aspects in a balanced manner. Therefore, the arrangement chosen for this volume follows rather closely the one of the conference where also time limitations had to be observed.

The editors wish to express their sincere thanks to all who have assisted them by making the idea of this symposium known within the scientific community by acting as sponsor or cosponsor, who have assisted them in the reviewing process and have done a good job by putting together special sessions devoted to one main theme. Last but not least the editors thank Creditanstalt-Bankverein for their generous support for the production of these Proceedings.

Vienna, January 1997 I.Troch, F. Breitenecker

Contents

Invited Lectures

Good random number generators are (not so) easy to find 1
P. Hellekalek (Salbzurg, A)

Modelling and analysis of hybrid systems 17
S. Engell (Dortmund, D)

Qualitative modelling of dynamical systems - motivation, methods, and prospective applications 33
J. Lunze (Hamburg, D)

Fuzzy and Qualitative Modelling

Multifaceted Modelling

Special Session organized by S. Strmcnik and D. Juricic (Ljubljana, SI)

Tbe structure and consistency of the model-base for qualitative model-based diagnosis 45
K. M. Hangos, I. Gal (Budapest, H), D. Juricic (Ljubljana, SI)

Complexity reduction in fuzzy modeling 51
M. Setnes, R. Babuska, H. B. Verbruggen (Delft, NL)

Analytical solution approaches of non uniform temperature profiles. Application to heat exchanger simulations 57
J. Castillo, Ph. Bogaerts (Brussels, B)

Combination of two approaches to modelling of pressure-level pilot plant 63
S. Milanic, M. Atanasijevic-Kunc, R. Karba, B. Zupancic (Ljubljana, SI)

Is it reasonable to tbink of multiple-model-based flexible recipe control? 69
N. Hvala, S. Strmcnik, D. Sel, S. Milanic (Ljubljana, SI)

Contributed papers:

Fuzzy differential equations 75
M. Oberguggenberger (Innsbruck, A)

Finite element metbod with fuzzy parameters 81
Th. Fetz (Innsbruck, A)

Fuzzy decisions in discrete event simulation 87
M. Lingl, F. Breitenecker (Wien, A)

Advanced neural network training for discrete time nonlinear dynamic system modeling 91
J. Si, G. Zhou (Tempe, USA)

Uncertainties in modelling. Application to heat transfer in buildings, coupling radiative and convective excbanges 97
P. Aude, G. Rusaouen, P. Depecker (Villeurbanne, F)

Fuzzy modeling for the biosynthesis control 103
L. Rusinov, V. Holodnov, G. Panov (St. Petersburg, Russia)

Automation of Modelling and Bond Graphs

Contributed papers:

Tensor analysis based symbolic computation methods for mechatronic systems 111
K. Schlacher, A. Kugi, R. Scheidl (Linz, A)

Bond graph approacb for relative degrees and zero dynamics analysis of linear systems 117
R. Fotsu Ngwompo, S. Scavarda, D. Thomasset (Lyon, F)

Bond-graph modelling for geometric decoupling control 123
J. M. Bertrand, C. Sueur, G. Dauphin-Tanguy (Villeneuve d'Ascq, F)

Study of causal loops using bond graphs and the scattering formalism 129
A. Kamel, G. Dauphin-Tanguy (Villeneuve d'Ascq, F)

Petri Nets and Discrete Event Modelling

Petri Nets for Modelling, Verification and Synthesis of Systems

Special Session organized by H.-M. Hanisch (Magdeburg, D)

Distributed deadlock avoidance control: Petri net approach 135
R. Wojcik (Wroclaw, PL), K. Hasegawa, M. Sugisawa (Yokohama, J), Z. Banaszak (Zielona Gora, PL)
Action logical correctness proving 141
K. Lautenbach (Koblenz, D)
Quantities of Petri systems 149
H. J. Genrich (St. Augustin, D)
A comparison of Petri net supervisory approaches for state specifications 159
A. Giua (Cagliari, I)
Controller synthesis by means of symbolic backward search 165
H.-M. Hanisch, A. Lüder, J. Thieme (Magdeburg, D)
On exploiting the analysis power of petri nets for the validation of discrete event systems 171
M. Heiner (Cottbus, D)
Extended net condition/event systems for modelling of manufacturing and process technology 177
M. Rausch (Pittsburgh, USA)
Invariance of predicates and state feedback logic synthesis of controlled time petri nets 183
C. Haoxun (Xi'an, PRC)

Discrete Event Models of Continuous Systems

Special Session organized by S. Engell (Dortmund, D)

A discrete modelling procedure for continuous processes based on state-discretization 189
H. A. Preisig, M. J. H. Pijpers, M. Weiss (Eindhoven, NL)
Nondeterministic automata as approximations for continuous systems - an approach with an adjustable degree of accuracy 195
J. Raisch (Stuttgart, D)
Generating timed discrete models of continuous systems 203
O. Stursberg, St. Kowalewski, S. Engell (Dortmund, D)
Hybrid flow nets for batch process modelling and simulation 211
J.-M. Flaus (St. Martin d'Heres, F)
A formal methodology for representing qualitative equations with Petri nets 217
A. Fanni, A. Giua (Cagliari, I)
Discrete event models in batch control 223
M. Tittus, K. Akesson (Gothenburg, S)

Contributed papers:

Using general purpose discrete simulators for microscopic modelling and simulation of traffic systems 229
C. Kiss, M. Klug, F. Breitenecker (Wien, A)
From Petri-net simulations to effective insentive systems 235
M. Hutter, E. Schmid (Bülach, CH)
Petri net simulation of a distributed automation system based on Profibus 241
G. Marschall (Hamburg, D)
Modelling and simulation of supervisory process control systems 247
G. Music, D. Matko (Ljubljana, SI)
A decentralised algorithm to determine invariants in Petri Nets 253
M. Boutayeb, A. Bourjij, M. Darouach, S. Nowakowski (Nancy, F)

Identification

Fitting Mathematical Models to Real Processes

Special Session organized by H. Hahn (Kassel, D)

Identification of damage in reinforced concrete structures 259
T. Jahn (Kassel, D)

Modelling of a two link flexible robot using the multibody system MOBILE 265
W. Bernzen, B. Riege, S. Hartmann (Duisburg, D)
Fitting mathematical models of a planar servo-pneumatic test facility to the laboratory experiment by using exact linearization techniques 271
F. Hecker (Stuttgart, D), H. Hahn (Kassel, D)
Fitting mathematical models of a servo-pneumatic actuator to the laboratory experiment using exact linearization techniques 281
A. Piepenbrink (Friedrichshafen, D), H. Hahn (Kassel, D)
Mathematical modeling, computer simulation and dynamical analysis of a single-point-drive eccentric press 291
M. Neumann, H. Hahn, K.-D. Leimbach (Kassel, D)

Contributed papers:

Nonlinear model structure identification using genetic programming 301
G. J. Gray, D. J. Murray-Smith, Y. Li, K. C. Sharman (Glasgow, UK)
Modelling and identification for highly nonlinear processes 307
M. Boutayeb, M. Darouach (Nancy, F), P. M. Frank (Duisburg, D)
Towards less sensitive identification of a submarine nonlinear dynamic model 313
S. Ziani-Cherif, G. Lebret (Nantes, F)
Probability model identification 319
L. Kuznetsov (Lipetsk, Russia)
New technologies design of experiment 323
A. A. Naumov, A. V. Zemnitsky (Novosibirsk, Russia)

Software Tools

Contributed papers:

Simulation of modular dynamic systems 329
A. Rükgauer, W. Schiehlen (Stuttgart, D)
Data models for object-oriented modeling and simulation 335
R. Girelli, C. Maffezzoni (Milano, I)
PROMOT/DIVA: a prototype of a process modeling and simulation environment 341
F. Tränkle, A. Gerstlauer, M. Zeitz, E. D. Gilles (Stuttgart, D)
NETSIM-NETCALC - a complete papermill simulation package 347
E. Brenner (Graz, A), J. K. Weinhofer (Linz, A)
Computer aided analysis of some nonlinear phenomena in relay systems 353
A. Moeini, D. P. Atherton (Brighton, UK)
Modeling of looper, AGC system and development of a dynamic simulator of the hot strip steel mill 359
C. J. Park, J. H. Kwak, W. H. Lee, K. T. Lee (Pohang, Korea)
GMS - A graphical-based modeling system 365
M. Yongguang, W. Bingshu (Baoding, P. R. China)
On the supremal controllable Grafcet of a given Grafcet 371
J. Zaytoon, C. Ndjab (Reims, F), J. M. Roussel (Cachan, F)

Modelling in Practice: Common Roots and Differences

Special Session organized by F. Kraus (Zürich, CH)

Modelling of rivers for level control 377
J. Chapuis, (Zürich, CH), R. Sachs (Zug, CH)
Mathematical modeling of electrically stimulated muscle 383
J. Schultheiss, L. del Re, F. Kraus (Zürich, CH)
The mechanometrics approach to monitoring and fault detection of a hydro-mechanical system 389
J. Milek, H. Güttinger (Winterthur, CH)
Modeling for control of mean arterial blood pressure (MAP) during anesthesia 395
C. W. Frei, M. Derighetti (Zürich, CH), A. M. Zbinden (Bern, CH)
Continuous and discrete time representations in modelling and identification 401
F. Kraus, W. Schaufelberger (Zürich, CH)

General Engineering Applications

Application of Mathematical Methods in Metal Industry

Special Session organized by U. Epple (Aachen, D)

Thin-strip casting - modeling of the thickness profile with neural networks 407
U. Albrecht-Früh, R. Kopp (Aachen, D)

On-line process modeling and optimization for a 20high cold rolling mill 413
A. Schneider (Wuppertal, D), R. Werners (Ratingen, D)

Integration of model-based methods in process control systems 419
D. Meyer (Aachen, D)

Contributed papers:

Simulation of the regeneration of dust filters 425
Ch. Stöcklmayer, W. Höflinger (Wien, A)

Direct and adjoint oil spill estimates 431
Yu. N. Skiba (Mexico)

Wavelet modelling of high resolution radar imaging and clinical magnetic resonance tomography 437
W. Schempp (Siegen, D)

Application of mathematical modeling for a substantiation of the computer theory of one class of strapdown inertial systems of orientation and navigation 443
P. K. Plotnikov, V. B. Nikishin, A. S. Plotnikov (Saratov, Russia)

An application in postal automation: two ways of modeling a transport process 449
B. Lohmann (Konstanz, D)

In-gronnd effect inflow models for lifting rotors using a finite state modeling approach 455
J. V. R. Prasad, H. Zhang (Atlanta, USA), D. A. Peters (St. Louis, USA)

Discontinuous-continuous mining system operation as Markov process 461
V. Pavlovic (Belgrade, YU)

Discrete probabilistic modelling of mineral processing 467
A. Walaszek-Babiszewska (Gliwice, PL)

Traffic Modelling

Special Session organized by K.-O. Proskawetz (Braunschweig, D)

Modelling and simulation of the new european train control system 473
A. Janhsen, K. Lemmer, B. Ptok, E. Schnieder (Braunschweig, D)

Petrinet based model of container terminal logistics 479
K. Müller, E. Schnieder (Braunschweig, D)

Simulation and virtual reality in requirements engineering 485
A. Janhsen, M. Krone (Braunschweig, D)

MSO: Modelling - Simulation - Optimisation in large scale traffic system 491
A. Graber (Zürich, CH)

The impact of dynamic speed adaption on macroscopic traffic flow simulation 497
F. Meißner (Hamburg, D)

Distributed traffic simulation in Java - first findings 503
S. Schmittger (Ulm, D)

Electrical Systems

Mobile Communication Networks

Special Session organized by C.-H. Rokitansky (Aachen, D)

New wide-band propagation channel model for the mm-wave band 509
J. Fernandes, J. Neves (Aveiro, P)

Fast collision resolution in wireless ATM networks 517
D. Petras, A. Krämling (Aachen, D)

Contributed papers:

Bottleneck analysis for computer and communication systems with workload variabilities and uncertainties 525
J. Lüthi, G. Haring (Wien, A)

Modelling transmission lines effects in integrated circuits by a mixed system of DAEs and PDEs 535
M. Günther (Darmstadt, D)

An improved iron loss model for rotating electrical machines 541
L. R. Dupre, R. Van Keer, J. A. A. Melkebeek (Gent, B)

Prediction in power systems with non stationary load patterns 547
K. Voigtländer, H.-H. Wilfert (Dresden, D)

Some considerations about power systems modeling concerning transient stability analysis 553
N. G. Bretas, L. F. Costa Alberto (Sao Paulo, Brazil)

Modelling of unit-transformers for nonlinear analysis 561
M. Fette, R. Sisterhenn, J. Voss (Paderborn, D)

On the solvability of nonlinear differential-algebraic models of electrical power systems 569
M. Fette, D. Hisseine, J. Voss (Paderborn, D)

Mathematical modelling of failure states on induction machines 577
R. Fiser, S. Ferkolj (Ljubljana, SI)

Simulation for control of interconnected power system 583
M. U. Bogatyriev (Tula, Russia)

Mechanics and Mechatronics, incl. Robotics

Modelling in the Design of Mechatronic Systems

Special Session organized by J. Lückel (Paderborn, D)

Functional modelling and simulation in mechanical design and mechatronics 589
J. Lückel, J. Wallaschek (Paderborn, D)

Parametric bydraulic valve model including transitional flow effects 599
N. Mittwollen, T. Michl, R. Breit (Stuttgart, D)

Integrated product modeling - a contribution to computer aided product development 605
J. Gausemeier, D. Brexel, M. Flath, F. Kallmeyer, M. Miksic (Paderborn, D)

On the optimal design of (engine mounts and) vibration absorbers in cabriolets based on an experimentally identified car-body model 611
K. Reinke, J. Wallaschek (Paderborn, D), M. Heidrich, A. Paul (Weinheim, D)

Modelling and simulation of complex mechatronic systems for real-time simulation on parallel computers 617
V. Enderlein, J. Obermüller (Chemnitz, D), M. Hahn, U. Honekamp (Paderborn, D)

Advanced Robot and Model-Based Control Techniques

Special Session organized by S. G. Tzafestas (Athens, GR)

Incremental control of three cooperating robots in large-object transfer operations 623
C. S. Tzafestas (Paris, F), S. G. Tzafestas, P. Prokopiou (Athens, GR)

A case of intelligent autonomous robot managing uncertainty by means of multisensor fusion 631
F. Matia, A. Jimenez (Madrid, E)

Application of the model reference adaptive control to robot impedance control 637
D. Matko, R. Kamnik, T. Bajd (Ljubljana, SI)

On the applicability of neural-net and fuzzy models in model based control 643
D. Matko, K. Kavsek-Biasizzo, S. Milanic, R. Karba (Ljubljana, SI), O. Hecker (Darmstadt, D)

Pseudo-link-based control of a flexible manipulator 649
G. Fernandez, J. C. Grieco (Caracas, Venezuela), M. Armada, P. Gonzalez de Santos (Madrid, E)

Contributed papers:

A dynamic interpretation of the classical friction model 655
G. Ferretti, G. Magnani, A. Bonsignore (Milano, I)

Model order reduction by means of a continuous friction law for a CVT chain drive simulation 661
J. Srnik, F. Pfeiffer (München, D)

Experimental estimation of dynamic parameters for an industrial manipulator 667
G. Antonelli, F. Caccavale, P. Chiacchio (Napoli, I)

Singular perturbation model of robots with elastic joints and elastic links constrained by rigid environment 673
P. Rocco (Milano, I)

Path planning for robots by stochastic optimization methods 679
K. Marti (München, D)

Modelling and simulation of an agricultural tracked vehicle 685
G. Ferretti, R. Girelli (Milano, I)

Calculation of the Jacobian matrix for mechanisms with more than one DOF in each joint 693
K. Gotlih (Maribor, SI)

Automatic Control

Modular Modeling and Engineering Applications

Special Session organized by K. H. Fasol (Bochum, D)

Interaction of modeling and nonlinear control design 699
P. C. Müller (Wuppertal, D)

Energy flow modeling of mechatronic systems via object diagrams 705
M. Otter (Wessling, D), H. Elmqvist (Lund, S)

Object-oriented modelling of control engineering data 711
U. von Döllen, M. Schlothane (Bochum, D)

Hierarchic modular statemodels for development and verification of control systems 717
M. Rempe (Bochum, D)

Nonlinear dynamic modeling of a gas engine using an RBF-network 723
R. Korb, H. P. Jörgl (Wien, A), B. Lutz (Jenbach, A)

Module-oriented modeling of aircraft structural behaviour 729
G. Beuck, L. Merz (Hamburg, D)

Structure of rainfall-runoff nonlinear model with physical interpretation 737
C. Verde, C. Cruickshank (Coyoacan, Mexico)

Modelling and simulation of purification processes in wastewater treatment plants using the modular toolbox KSIM 743
M. Köhne, M. Schuhen, D. Schönberger (Siegen, D)

Mathematical Modelling for Multidimensional Nonlinear Characteristic Fields

Special Session organized by H. Tolle (Darmstadt, D)

Nonlinear dynamics modelling via operating regime decomposition 751
Th. H. Göttsche, K. J. Hunt (Berlin, D), T. A. Johansen (Trondheim, N)

Dead-zone adaptation vs. overtraining phenomenon for basis function networks 757
M. Heiss (Wien, A)

A simulated annealing algorithm for approximating triangulated surfaces 763
M. Kropp (Unterentfelden, CH), P. R. Remmele, T. Roos (Zürich, CH)

Modelling nonlinear characteristics using hierarchical Delaunay-networks 769
T. Ullrich, H. Tolle (Darmstadt, D)

Comparison of node insertion algorithms for Delaunay-networks 775
M. Brown (Southampton, UK), T. Ullrich (Darmstadt, D)

Contributed papers:

Time-suboptimal control design of singularly perturbed systems by reduced order feedback design 781
S. A. Mikhailov, P. C. Müller (Wuppertal, D)

Phase margin design of phase lead controllers using autotuning 787
A. F. Boz, D. P. Atherton (Brighton, UK)

The relationship between control requirements, process complexity and modeling effort in the design process of river control systems 793
B. Cuno (Fulda, D), S. Theobald (Karlsruhe, D)

Physical Applications

Contributed papers:

The mathematical modelling of heat gain in buildings trough transparant insulation 799
D. Constales, R. Van Keer (Gent, B)
Parallel simulation of heat transfer processes 805
M. Holzinger, F. Breitenecker (Wien, A)
Continuum modelling of large networks in domains 809
J. Gwinner (Neubiberg, D)
Semi-physical modelling of a FDI benchmark process 815
D. Juricic, G. Dolanc, A. Rakar (Ljubljana, SI)

Environmental Systems

Environmental Modelling and Simulation

Special Session organized by A. Sydow (Berlin, D)

Sequential and parallel performance of EMEP's transboundary air pollution model 821
St. Unger (Berlin, D)
Simulation of suspended particulate matter transport in a navigation canal 827
Ch. Engelhardt, D. Prochnow, H. Bungartz (Berlin, D)
Determination of sources of particles in air by trajectory models 831
I. Gerharz, A. Sydow, S. Unger (Berlin, D)
Traffic simulation with the AI controlled CASSANDRA simulation system 835
A. Javor, M. Benko, Gy. Buzasy, A. Farkas, G. Szucs (Budapest, H)
Examples of the knowledge bases used in the computer expert systems for environmental protection 841
C. Orlowski, A. Tubielewicz (Gdansk, PL)
Analysis of traffic induced air pollution in the Budapest region 847
A. Sydow, Th. Lux, R.-P. Schäfer, M. Schmidt (Berlin, D)

Modelling for Control of Wastewater Treatment Processes

Special Session organized by J. P. Babary (Toulouse, F) and P. Lessard (Quebec, CDN)

Model based selection of an appropriate control strategy: application to an anaerobic digester 853
F. Chaume, J. F. Beteau (Grenoble, F)
The process of model building and simulation of ill-defined systems: Application to wastewater treatment 859
B. Kops, H. Vangheluwe, F. Claeys, F. Coen, P. Vanrolleghem, Z. Yuan, G. C. Vansteenkiste (Gent, B)
Model based design of a novell process for ammonia removal from concentrated flows 865
C. Hellinga, M. C. M. van Loosdrecht, J. J. Heijnen (Delft, NL)
Modelling of an intermittently aerated bioreactor 871
J. Derco, A. Kovacs, S. Shansab (Bratislava, SK)
A reduced order model for control of a single reactor activated sludge process 877
S. Julien, J. P. Babary (Toulouse, F), P. Lessard (Quebec, CDN)

Contributed papers:

A fuzzy self-tuning algorithm for power system management in next generation vehicles 883
X. J. R. Avula (Rolla, USA), M. Sridhar (Detroit, USA)
Models of migration of organic pollutants in soils 889
A. Urbaniak, M. Wika, A. Voelkel (Poznan, PL)

Biology and Biotechnical Engineering

Modelling of Chemical and Biochemical Reactors

Special Session organized by Ph. Bogaerts (Bruxelles, B)

Modelling of batch reactors for simulation, optimization and process control 895
M. V. Le Lann, M. Cabassud, G. Casamatta (Toulouse, F)

Observers synthesis methodology for chemical reactors 901
A. M. Gibon-Fargeot, F. Celle-Couenne, H. Hammouri (Villeurbanne, F)
Design of the initiation of batch polymerisation reactors 907
S. Nemeth, P. Arva (Veszprem, H)
Analytical solution of the non uniform heat exchange in a reactor cooling coil with time varying fluid flow 913
Ph. Bogaerts, J. Castillo, R. Hanus (Bruxelles, B)
Optimal experimental design for practical identification of unstructured growth models 919
K. J. Versyck, J. E. Claes, J. F. Van Impe (Heverlee, B)

Contributed papers:

Determination of renal reserve capacity by identification of kinetic systems 925
W. Estelberger, S. Zitta, K. Stoschitzky, R. Zweiker, T. Lang, F. Mayer, G. Reibnegger (Graz, A)
A new model of the microplasma pulsation in p-n junction 931
B. Datsko, A. Demchuk, V. Gafiychuk, G. Ilchuk, Yo. Hromyak, V. Pavlysh (Lviv, UKR)
Mathematical modelling and cochlear implants 937
P. Lutter (Zürich, CH)
Simulation analysis of the effects of the junctional folds on spontaneous generation of the miniature endplate current at neuromuscular junction 943
T. Naka (Saitama, J)
Bioheat transfer problem and applications 949
I. Lubashevsky (Moscow, Russia), V. Gafiychuk (Lviv, UKR)
A dynamical model for alcoholic fermentation in winemaking with temperature control 955
A. P. G. M. Moreira, J. L. Martins de Carvalho (Porto, P)

Economic and Social Systems

Contributed papers:

Optimal macroeconomic policies for Austria: a simulation analysis 961
R. Neck (Osnabrück, D), S. Karbuz (Istanbul, TR)
The evolution of money and genetic algorithms 971
S. Staudinger (Vienna, A)
A deterministic model for manpower systems 977
H. Gao (Prague, CZ), B. F. Gao, D. H. Pan (Shenyang, P. R. China)
How does the introduction of expectations affect an economic model? 981
B. Lopez, J. M. Pacheco (Las Palmas, E)
Frustra laborat qui omnibus placere studet 985
R. Starkermann (N'Rohrdorf, CH)

Modelling and Designing of Information Processes in Anthropocentral Systems

Special Session organized by B. E. Fedunov (Moscow, Russia)

Computer support of macroeconomic decisions 993
J. Babarowski, J. Gutenbaum, M. Inkielman (Warszawa, PL)
Motive-object model of task decision process 999
P. I. Sosnin (Ulianovsk, Russia)
Models of complex anthropocentral objects for the system design of on-board algorithm and display support 1005
B. E. Fedunov (Moscow, Russia)

Theoretic Aspects

Inverse Problems

Special Session organized by G. Anger (Berlin, D)

Ill-posedness aspects of some nonlinear inverse problems 1009
G. Fleischer, B. Hofmann (Chemnitz, D)
Efficient detection of an inclusion in a conductor 1015
G. Alessandrini, E. Rosset (Triste, I)

Model Validation Methods and Applications

Special Session organized by D. J. Murray-Smith (Glasgow, UK)

Model validation in iterative identification and controller design 1021
M. Zele, D. Juricic (Ljubljana, SI)
The role of model structure validation in pharmacokinetic studies 1027
R. Karba, A. Mrhar, A. Belic, I. Grabnar, S. Primozic, B. Zupancic (Ljubljana, SI)
Applications of the distortion method for model validation 1033
G. J. Gray, Lew Koi Voon, D. J. Murray-Smith (Glasgow, UK)

Contributed papers:

The tearing problem: definition, algorithm and application to generate efficient computational code from DAE systems 1039
E. Carpanzano, R. Girelli (Milano, I)
Transfer function models for multidimensional systems 1047
R. Rabenstein (Erlangen, D)
Modelling of state events 1053
J. Plank, F. Breitenecker (Wien, A)
A unified theory for automation systems 1057
A. Mircescu, E. Schnieder (Braunschweig, D)
Controlled quasicontinuous orbits 1063
V. Gontar, M. Gutman (Beer-Sheva, IL)
Identification of a nonlinear compressor model 1065
C. J. Rivera, J. V. R. Prasad (Atlanta, USA)
Modelling of distributed parameter system for the state observation purposes 1071
W. Byrski, P. Kubik (Krakow, PL)
A fault detection filter for bilinear systems with unknown inputs 1077
C. Mechmeche, M. Zasadzinski, H. Rafaralahy, J. Y. Keller, M. Darouach (Nancy, F)
Wavelets in optimization and approximations 1083
A. N. Fedorova, M. G. Zeitlin (St. Petersburg, Russia)
Mathematical models of discrete self-similarity 1089
F. M. Borodich (Glasgow, UK)
Computers as living systems 1095
V. Gontar, (Beer-Sheva, IL)

Posters

A test based on training samples for the observation classification problem 1105
I. G. Bairamov (Tandogan-Ankara, TR)
Simulation of a packed bed distillation column using orthogonal collocation and finite element 1106
Y. Cabbar, M. Alpbaz, H. Hapoglu, S. Karacan (Ankara, TR)
Predictive model of ambulatory systolic blood pressure, using a semi quantitative measure of physical activity 1107
S. Charbonnier, F. Marques, A. Cheruy (St. Martin d'Heres, F)
Dynamic model simulation of aerobic yeast production in a batch bioreactor 1108
S. Ertunc, N. Bursali, Y. Cabbar, M. Alpbaz (Ankara, TR)
Modelling a communal disposal system for biological waste 1109
F. Koch, P. Krejsa, E. Rybin (Seibersdorf, A), F. Breitenecker, M. Holzinger, M. Lingl, M. Zimmermann (Wien, A)
Modelling of the human arterial network for an expert system for preoperative predictions 1110
K. Kaser, J. Krocza, M. Suda (Seibersdorf, A), C. Almeder, F. Breitenecker, S. Wassertheurer (Wien,A)
Electro-hydrodynamics 1111
W. Khan (Mayaguez, Puerto Rico)
Modelling of a traffic junction with a general purpose graphical discrete simulator 1112
C. Kiss (Wien, A)
Modelling individual properties of objects in microscopic traffic simulation using GPSS/H 1113
M. Klug (Wien, A)

Simulation and identification of compartment models for the determination of parameters of renal function 1114
H. Url (Wien, A)
What's wrong with bond graphs 1115
H. Mann, J. Hozak (Praha, CZ)
Mathematical modelling via internet 1116
H. Mann, L. Waldmann (Praha, CZ)
Chances and opportunities of numerical simulation in refrigeration 1117
J. Philipp, M. Kauschke (Dresden, D)
Two-way composition for lossless layered media systems 1118
E. Szaraniec (Krakow, PL)
One model of a human-machine problem-solving system 1119
V. A. Vishnyakov, O. V. German (Minsk, Belarus')
Simulation of temperature fields brought about by grinding hardly worked materials 1120
S. F. Lushpenko (Kharkov, Ukraine)

Index of authors 1121

GOOD RANDOM NUMBER GENERATORS ARE (NOT SO) EASY TO FIND

P. Hellekalek
Dept. of Mathematics, Salzburg University
Hellbrunner Straße 34, A-5020 Salzburg
e-mail: peter.hellekalek@sbg.ac.at

Abstract. Every random number generator has its advantages and deficiencies. There are no "safe" generators. The practitioner's problem is how to decide which random number generator will suit his needs best.

In this paper, we will discuss criteria for good random number generators: theoretical support, empirical evidence and practical aspects. We will study several recent algorithms that perform better than most generators in actual use. We will compare the different methods and supply numerical results as well as selected pointers and links to important literature and other sources. Additional information on random number generation, including the code of most algorithms discussed in this paper is available from our web-server under the address

`http://random.mat.sbg.ac.at/`

1 Introduction

Random number generators ("RNGs") are the basic tools of stochastic modeling. As any other craftsman, the modeler has to know his tools. Bad random number generators may ruin a simulation. There are several pitfalls to be avoided.

For example, if we try to check the correlations between consecutive random numbers $x_0, x_1, \ldots$, then a (still!) widely used generator produces nonoverlapping pairs (x_{2n}, x_{2n+1}), $n = 0, 1 \ldots$ above suspicion at the first look, see Figure 1.1a. In sharp contrast, the triples $(x_{3n}, x_{3n+1}, x_{3n+2})$ are extremely correlated and happen to lie on only fifteen planes, see Figure 1.1b. In both figures, 2^{15} points have been generated. For details on this phenomenon, we refer to Sections 4 and 5.

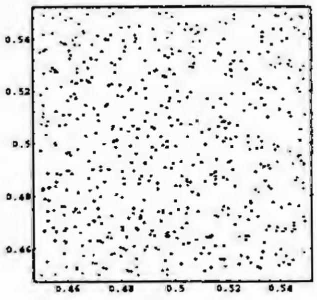


Figure 1.1a
LCG(2^{31}, 65539, 0, 1)
Dimension 2: Zoom into the Unit Interval

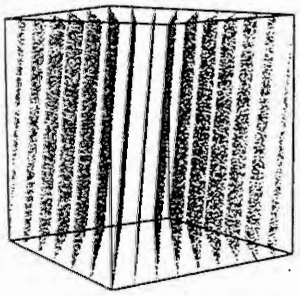


Figure 1.1b
LCG(2^{31}, 65539, 0, 1)
Dimension 3: The 15 Planes

In this paper, safety-measures against such unpleasant surprises will be given. We will discuss the current standards for good random number generators, the underlying mathematical and statistical concepts, and some new generators that meet these standards. Further, we will summarize the advantages and deficiencies of several algorithms to generate and test random numbers. Finally, we will present a "RNG Survival Kit" that contains the most important literature on this subject and links to web-sites that offer code and documents, and a "RNG Checklist" that allows the reader to assess his preferred generator on the basis of the concepts given in this concise survey.

A good generator is not so easy to find if one sets out to design it by oneself, without the necessary mathematical background.

On the other hand, with the references and links we supply, a good random number generator designed by experts is relatively easy to find.

The standard approach to generate nonuniform random numbers is to produce uniform random numbers first and then to transform them, see the monograph [7] and the software package CRAND [58, 59]. In this paper, we will restrict our attention to uniform random number generators.

2 What is a *Good* RNG?

The underlying problem is the following (see [2] for this quotation):

> *"Monte Carlo results are misleading when correlations hidden in the random numbers and in the simulated system interfere constructively."*

The answer to the question above will depend on the target application. In the present paper, we will focus our interest on uniform random number generators appropriate for stochastic simulation. In cryptology, the requirements for generators are different, see [26].

From a practitioner's view, random number generators are good if they yield the correct results in as many applications as possible. This desire cannot be fulfilled completely. It is known that every generator has to fail in certain simulations, in models that interfere with the particular regularities of a given generator and exhibit the hidden correlations between the random numbers. This unpleasant situation is a fact of (scientific) life and cannot be circumvented, see [2], [33], and [39]. Random number generators are nothing more than deterministic algorithms that produce numbers with certain distribution properties. These sequences of so-called "random" numbers are periodic. Their task is not to simulate "randomness", which is a notion that is difficult to define in terms of practical relevance, but to give the correct results in a simulation and, hence, to pass certain tests that the user considers relevant. The problem with randomness is partly due to the fact that random variables are mixed up with numbers. For example, the notion of independence is only defined for random variables, not for numbers. We refer the reader to [62] for details and some enlightening comments on this subject.

How can we provide good random number generators if we don't know the target application in advance? The designer of a generator does not know the sampling procedure nor the sample sizes nor the dimensions the practitioner will choose in his simulation. There are two facts every user of random number generators should know.

> FACT ONE: With random number generators, there are *no guarantees*, *only predictions*. This is not because the word "randomness" is involved but because the finitely many random numbers we produce and their transformed variates cannot fit every imaginable distribution well enough. Every generator has its regularities which, ocassionally, may become deficiencies. Hence, in a given application, even reliable generators may fail.
>
> FACT TWO: Although there are no guarantees, there are *mathematical safety-measures* against wrong simulation results due to inappropriate random number generators.

The first precaution is *empirical testing* of random number generators. We may run *application-specific tests* with known theoretical results and compare them to our empirical results, see [61] for examples from physics. As an alternative, we may search in the existing literature for tests that resemble our simulation problem. If we are lucky, a particular test design and parameters like the sample size or the dimension will be similar to our setup. Frequently, we will be unlucky. Good starting points for our search are the surveys [52, 54, 30, 33] and the "Links"-page at the Web-site `http://random.mat.sbg.ac.at/`.

If we cannot find a test that resembles our simulation problem, then we may submit the random numbers we want to use to a battery of empirical tests. It should be noted that empirical tests cannot prove anything formally, like "randomness" for a random number generator. The only conclusion we can derive from the results of an empirical test is that the samples that have been used pass or fail this particular test, nothing more and nothing less. Nothing can be concluded for other samples, other dimensions, or other initializations of the generator under study. Of course, if our generator passes many empirical tests, this will improve our confidence as well as our chances to get the right simulation results with this generator. We refer to Section 5 for further information.

The second safety-measure is *theoretical support* for a random number generator. It means that we know about the period length of the generator, know some of its structural properties, and its correlation behavior, see Section 4 for details. The period length will limit the size of the samples we can use safely with this generator. The structural properties will help us to decide if there might be unwanted side-effects in the simulation. The correlation properties of a random number generator are of central importance for many stochastic simulations.

Practical aspects of random number generators concern the speed of the algorithm, the ease of implementation, the possibility to use parallelization techniques, and the availability of portable implementations. One generator of a given type will not be enough for numerical practice. We need tables of tested parameters to be able to implement several generators of the same kind. If we happen to work with high-performance computers, then we will also require that extremely large samples and, necessarily, large periods are available.

A good random number generator fulfills this catalog of safety standards. As a matter of fact, all available generators lack answers in certain subcategories of this checklist. It is up to the practitioner to decide which aspects of a generator are important to him and to select an appropriate generator accordingly. We provide an "RNG Checklist" to assist with this choice in Section 9.

3 Examples of RNGs

In this section we will present several recent advances in the construction of random number generators that merit a strong recommendation, in particular to those practitioners that require large samples (i.e. long periods), speed, and/or little correlations. We will revisit these examples in Section 7, employing the concepts developed in Sections 4,5, and 6.

We refer the reader to the surveys [29, 33] for a comprehensive discussion of linear algorithms. For a concise presentation of most of the available algorithms with an emphasis on the theoretical background and on nonlinear generators we recommend [54].

Linear methods are the best-known and most widely used algorithms to produce random numbers. Their practical advantages are speed, ease of implementation, and the availability of portable code, parameters and test results. An eye-catching phenomenon that occurs with linear types are lattice structures of the d-tuples constructed from consecutive random numbers. This fact is not at all a defect, but an intrinsic property of this family of generators. It allows to define and compute figures of merit like the *spectral test* or *Beyer quotients* (see Section 4).

The classical example of a random number generator is the linear congruential generator ("LCG"). It is defined by the linear congruence $y_{n+1} \equiv ay_n + b \pmod m$, $n \geq 0$, where we have to choose the modulus m, a multiplier a, an additive term b, and an initial value y_0. This recursion of order one produces a sequence of numbers $(y_n)_{n\geq 0}$ in the set $\{0, 1, \ldots, m-1\}$. Random numbers x_n in $[0, 1[$ are obtained by the normalization $x_n := y_n/m$. We will denote this generator by $\mathrm{LCG}(m, a, b, y_0)$. Examples of well-known LCGs are the Ansi-C system generator $\mathrm{LCG}(2^{31}, 1103515245, 12345, 12345)$, the "Minimal Standard" generator LCG(2^{31} − 1,16807,0), infamous RANDU, which is $\mathrm{LCG}(2^{31}, 65539, 0)$, the SIMSCRIPT generator LCG($2^{31}-1$,630360016,0), NAG's $\mathrm{LCG}(2^{59}, 13^{13}, 0, 123456789(2^{32}+1))$, and Maple's $\mathrm{LCG}(10^{12} - 11, 427419669081, 0, 1)$.

If we study the distribution of d-tuples $\mathbf{x}_n = (x_n, x_{n+1}, \ldots, x_{n+d-1})$ and if we generate all possible points in $[0, 1[^d$, then we will observe lattice structures. This is a well-known phenomenon. In the examples below, all possible points in dimension $d = 2$ have been produced.

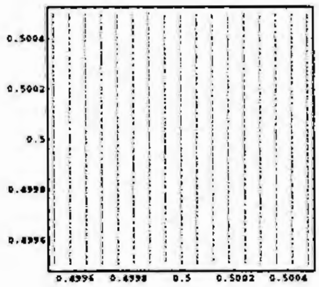


Figure 3.1a
Minimal Standard: LCG(2^{31} − 1,16807,0,1)

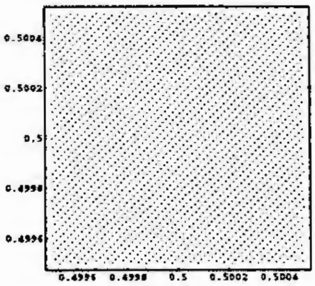


Figure 3.1b
SIMSCRIPT: LCG(2^{31} − 1,630360016,0,1)

Both LCGs have the period $2^{31} - 2$, which is best possible in this case (see [52, Section 7.3]). Figure 2 shows that maximum period does not guarantee good lattice structure. It does not imply good correlation properties either. This fact is true for all linear random number generators, not just LCGs.

There is a type of linear generator available that preserves the good properties of the LCG while improving upon some of its disadvantages. We increase the order of the linear recursion and obtain the *multiple recursive congruential generator* ("MRG").

Example 1 Let $m \geq 2$ be the modulus, $k \geq 1$ be the order of the recursion and choose $a_0, a_1, \ldots, a_{k-1}$ in $\mathbf{Z}_m := \{0, 1, \ldots, m-1\}$ with $\gcd(a_0, m)$=1. Then

$$y_{n+k} \equiv \sum_{j=0}^{k-1} a_j y_{n+j} \pmod{m}, \quad n \geq 0,$$

defines an MRG with initial values $y_0, \ldots, y_{k-1}$. We use the normalization $x_n := y_n/m$ to produce random numbers x_n in the unit interval $[0, 1[$. The maximum period of this generator is $m^k - 1$, see [54, 34]. The paper [34] contains tested parameters for MRGs with moduli m up to 2^{63} and code for one MRG in C. We will denote this MRG by "MRG1" and exhibit test results in Section 7.

One general problem with linear methods is the fact that correlations between random numbers separated by lags may be rather strong. Even in certain variants of the MRG, like the AWC and SWB generators of [45], this may lead to a very unfavorable performance in simulations, see [28, 33, 54] for further information. One solution to this problem is to combine generators. In the simplest version of this technique, we combine two generators by adding their output sequences $(x_n^{(1)})_{n\geq 0}$ and $(x_n^{(2)})_{n\geq 0}$ to obtain a new sequence $(x_n)_{n\geq 0}$,

$$x_n := x_n^{(1)} + x_n^{(2)} \pmod{1}, \quad n \geq 0.$$

If the two generators are chosen properly, then the period of the sequence $(x_n)_{n\geq 0}$ will be the product of the periods of the components. Combining generators without theoretical support may lead to disastrous generators. In the case of the LCG and MRG, the theory is well-known, see [31, 33].

Example 2 In [31], combined MRGs ("cMRG") were introduced and thoroughly analyzed. Further, a particular cMRG was assessed by the spectral test up to dimension $d = 20$ and its implementation in C was given. In Section 7, we will refer to this particular cMRG as "cMRG1".

Tausworthe generators can have unacceptably bad empirical performance. For this reason, in [32], combined Tausworthe generators ("cTG") were introduced to improve on the properties of single Tausworthe generators.

Example 3 In [32], an implementation in C of a cTG is given that has a period length of order 2^{88}. Further, the equidistribution properties of this type of generator are analyzed. In Section 7, we will present test results for this generator, which is denoted by "cTG1".

It is well-known that generalized feedback shift-register generators ("GFSR") are fast, although a little bit tricky to initialize. Recently, a very interesting variant of this linear method has been presented in [48, 49], the *twisted* GFSR ("tGFSR"). This generator produces a sequence $(\mathbf{x}_n)_{n\geq 0}$ of w-bit integers by the rule

$$\mathbf{x}_{n+p} = \mathbf{x}_{n+q} \oplus \mathbf{x}_n A, \quad n \geq 0,$$

where (w, p, q, A) are the parameters of the tGFSR and A is a $w \times w$ matrix with binary entries. This generator is fast and reliable if the parameters are chosen properly.

Example 4 The tGFSR "TT800" presented in [49] has a period length of 2^{800} and strong theoretical support. We will exhibit convincing empirical evidence in Section 7.

Inversive generators were constructed to overcome one property of linear generators that may turn into a deficiency (depending on the simulation problem), the lattice structure of d-tuples of consecutive random numbers. There are several variants of inversive generators, *inversive congruential* generators ("ICG"), *explicit-inversive* generators ("EICG"), *digital* inversive congruential generators ("dICG"), and combinations of ICGs and EICGs. Inversion certainly slows down the generation of random numbers. Compared to LCGs of the same size, inversive generators are three to ten times slower, depending on the processor's architecture (see [42]).

The importance of inversive random number generators stems from the fact that their intrinsic structure and correlation behavior are strongly different from linear generators. Hence, they are very useful in practice for verifying simulation results. We refer the reader to [17] for a concise survey of the ICG and EICG, in comparison to LCGs. A comprehensive discussion of all available nonlinear methods is contained in [54]. The implementation of inversive generators is discussed in [42]. This generic implementation in C is also available from the server `http://random.mat.sbg.ac.at/`. In the case of the ICG and EICG composite moduli lead to less convincing generators than prime moduli.

At the present state of the art, the dICG is slower than the ICG. From a disappointing speed factor of about 150 in the first implementation (see [8, page 72]) this disadvantage has now been reduced to a factor less than 8, see [57].

For a given prime number p, and for $c \in \mathbf{Z}_p$, let $\overline{c} := 0$ if $c = 0$ and $\overline{c} := c^{-1}$ if $c \neq 0$. In other words, $\overline{c}$ equals the number c^{p-2} modulo p.

Example 5a Inversive congruential generators ("ICG") were introduced in [9]. We have to choose the modulus p, a multiplier a, an additive term b, and an initial value y_0. Then the congruence

$$y_{n+1} \equiv a\overline{y}_n + b \pmod{p}, \quad n \geq 0, \tag{1}$$

defines an ICG. We denote this generator by ICG(p, a, b, y_0). It produces a sequence $(y_n)_{n\geq 0}$ in the set $Z = \{0, 1, \ldots, p-1\}$. Pseudorandom numbers x_n in $[0, 1[$ are obtained by the normalization $x_n := y_n/p$.

A prominent feature of the ICG with prime modulus is the absence of any lattice structure, in sharp contrast to linear generators. In the following scatter plot, all possible points (x_{2n}, x_{2n+1}), $n \geq 0$, in a region near the point (0.5, 0.5) are shown.

Figure 3.2
All Points of ICG(2^{31} – 1,1288490188,1,0)

Example 5b Explicit inversive congruential generators ("EICG") are due to [11]. The EICG is easier to handle in practice, for example when producing uncorrelated substreams. The cost is a slightly smaller maximum usable sample size, as empirical tests have shown (see [62, 41]).

We choose a prime number p, a multiplier $a \in \mathbf{Z}_p$, $a \neq 0$, an additive term $b \in \mathbf{Z}_p$, and an initial value n_0 in $\mathbf{Z}_p$. Then

$$y_n \equiv \overline{a(n + n_0) + b} \pmod{p}, \quad n \geq 0,$$

defines a sequence of pseudorandom numbers in $\{0, 1, \ldots, p-1\}$. As before, we put $x_n := y_n/p$, $n \geq 0$, to obtain pseudorandom numbers in $[0, 1[$. We shall denote this generator by EICG(p, a, b, n_0). In the definition of EICG(p, a, b, n_0), the additive term b is superfluous and can be omitted, see [18, 42].

It is easy to create ICG and EICG on demand. The choice of parameters for the EICG is simple. In case of the ICG, we may use a "mother-child" principle that yields many ICGs from one "mother" ICG (see [17]).

The "compound approach" presented in [10, 12] allows to combine ICG and EICG, provided they have full period. This method has important advantages: we may obtain very long periods easily, modular operations may be carried out with relatively small moduli, increasing the effectiveness of our computations, and the good correlation structure of the ICG and EICG is preserved. The price to pay is a significant loss of speed that makes combined inversive generators considerably slower than linear generators of comparable period length.

4 Theoretical Support

Theoretical support for random number generators is still widely ignored by practitioners. The three main questions here are period length, the intrinsic structure of the random numbers and -vectors produced by a generator, and correlation analysis.

It is clear that the period length of a generator will put a limit to the usable sample size. Random number generation is equivalent to drawing without replacement, see [30]. Hence, the sample size should be much smaller than the period length of the generator. In the case of linear methods, the square root of the period length seems to be a prudent upper bound for the usable sample size. This recommendation is based on empirical experience, there is no theoretical analysis available (see [43] for a short discussion). We refer to Section 7 for examples that show how different types of random number generators behave quite differently when the sample size is increased.

In the case of good random number generators, it is possible to provide conditions for the parameters of the generator to obtain maximum period length, see [54]. Further, it is important to have algorithms at hand to compute such parameters. The case of the ICG is a good illustration for this requirement, see [17, 54].

Intrinsic structures of random number generators like grid structures and related results like estimates of the number of points on hyperplanes are important to be known. For example, if one is aware of the grid structure of LCGs, then it will come as no surprise that this type of generator has difficulties with certain simulations. A good example is the *nearest pair test*, see [9] and [27, 36].

The most difficult and most important part of the theoretical assessment of random number generators is correlation analysis. More than twenty years of experience have shown that certain figures of merit for random number generators allow very reliable predictions of the performance of the samples produced with a generator in empirical tests. The latter are nothing less than prototypes of simulation problems. Hence, the importance of theoretical correlation analysis for numerical practice is beyond question. It should be stated clearly that none of these figures of merit can give us guarantees for the performance of the generator in our simulation. At present, there is no firm mathematical link between any figure of merit for random number generators and the empirical performance of samples. Nevertheless, and this fact is truly remarkable, the quality of prediction is excellent.

The basic concept to analyze correlations between random numbers is the following. Suppose we are given random numbers $x_0, x_1, \ldots$ in the unit interval $[0,1[$. To check for correlations between consecutive numbers, we construct either *overlapping* d-tuples $\mathbf{x}_n := (x_n, x_{n+1}, \ldots, x_{n+d-1})$ or *non-overlapping* d-tuples $\mathbf{x}_n := (x_{nd}, x_{nd+1}, \ldots, x_{nd+d-1})$ and assess the empirical distribution of finite sequences $\omega = (\mathbf{x}_n)_{n=0}^{N-1}$ in the d-dimensional unit cube $[0,1[^d$. The task is to measure how "well" ω is uniformly distributed. Strong correlations between consecutive random numbers will lead to significant deviations of the empirical distribution function of ω from uniform distribution, in some dimensions d. It is clear that the restricted type of d-tuples that is considered here cannot ensure against *long-range* correlations among the numbers x_n themselves. For this topic, we refer the reader to [5]. In the case of the EICG of [11], more general types of d-tuples have been considered (see also [53, 54]).

Interestingly, it has turned out that the behavior of *full-period* sequences ω with respect to theoretical figures of merit allows very reliable predictions of the performance of the random numbers x_n themselves in empirical tests. If the full-period point set ω has a good empirical distribution with respect to certain figures of merit in various dimensions d, then good empirical performance of the samples is highly probable. Practical evidence is that many target distributions will be simulated very well, see, for example, the empirical results in [15, 27, 41, 22]. This relation between properties of full-period sequences in higher dimensions and the behavior of –comparatively small– samples in low dimensions has not yet been put into rigorous mathematical form.

There are two schools of thought. The approach of School 1 (see [29]) is to optimize the parameters of a given type of generator such that the empirical distribution function of the point sets ω in $[0,1[^d$ approximates uniform distribution as closely as possible, leading to a so-called "super-uniform" distribution. This is done in as many dimensions d as is feasible in practice. The figure of merit that is used for this task is the *spectral test*, due to [3]. The spectral test has an important geometrical interpretation as the maximum distance between successive parallel hyperplanes covering all possible points $\mathbf{x}_n$ that the generator can produce. This interpretation leads to efficient algorithms to compute the value of the spectral test. We refer the reader to [25, 56, 16, 14, 29, 30, 33, 60] for details. The generator RANDU of Figure 1.1 is not bad with respect to the spectral test in dimension $d = 2$, but the value of the spectral test in dimension 3 is extremely small, thereby reflecting the catastrophic lattice structure in this dimension. This example explains why we have to consider the spectral test for a whole range of dimensions and compute its value for each of them.

The approach of School 2 (see [54]) is to construct generators where the empirical distribution function does not approximate uniform distribution "too well". The maximum distance between the empirical distribution function and uniform distribution is preferred to be of order $1/\sqrt{N}$, where N denotes the number of points $\mathbf{x}_n$ we consider. This is, roughly speaking and thinking of the law of the iterated logarithm "LIL" for discrepancy, the order of this quantity in the case of realizations of i.i.d. random variables on $[0,1[^d$ (see [24, 54]). We will call this kind of equidistribution "LIL-uniformity". The figure of merit that is used here is the two-sided Kolmogoroff-Smirnov test statistic, also known as *discrepancy* in number theory. In terms of discrepancy, super-uniformity has an order of magnitude $\mathcal{O}((\log N)^d/N)$. Niederreiter has developed a powerful number-theoretic method to estimate discrepancy by exponential sums, see [50, 52, 54]. This method has allowed to assess this figure of merit for most types of random number generators.

Both the spectral test and discrepancy have their advantages and shortcomings. The advantage of the spectral test is that it is readily computable even in higher dimensions (i.e. above d=20) under the condition that points $\mathbf{x}_n$ have lattice structure, see [38, 25, 56, 60]. This will only be the case for full-period sequences ω and for certain types of generators, mostly linear ones. Discrepancy is not limited to point sets with lattice structure. It is well-defined for every sample ω. Unfortunately, it is not possible to compute its value in practice, due to a complexity of order $\mathcal{O}(N^d)$, where N denotes the number of points and d the dimension. Recently, a probabilistic algorithm has been presented by [63]. There are only upper and lower bounds available, due to Niederreiter's advanced method. For both figures of merit, their distribution in dimensions $d \geq 2$ is not known. Therefore, we cannot design an empirical test for random number generators from this quantities. In dimension one, the commulative distribution function ("c.d.f.") of discrepancy is known. We refer the reader to [25, 52, 54] for details.

There is a new addition to this list of figures of merit, the *weighted spectral test*. It is due to [18]. This figure of merit is derived from the original concept of the spectral test. It is related to discrepancy, can be estimated as the latter, it does not require lattice structure for the point sets ω, and it takes $\mathcal{O}(dN^2)$ steps to compute it in any dimension d, see [20, 19]. The weighted spectral test may be interpreted as a mean square integration error, see [21]. Results on its distribution are already available, see [40, 21].

Beyer quotients are another figure of merit to assess lattices. Unfortunately, this quantity is known to be defined properly only in dimensions up to 6. Beyond this dimension, the Minkowski-reduced lattice bases involved need not be unique any more and their Beyer quotients might be different (see [39, Section 4.3.1]). For this reason, any results on bad Beyer quotients in dimensions higher than 6 are without mathematical justification at the present state of the art. A wrong basis might have been used.

5 Empirical Evidence

Theoretical support for random number generators is not enough. Empirical evidence is indispensable. Every empirical test is a simulation. If selected with care, then it will cover a whole class of simulation problems. As we have indicated before, nothing can be deduced from the results of an empirical test if the practitioner uses completely different parameters in his own simulation problem.

It is relatively easy to design an empirical ("statistical") test for random numbers. Every function of a finite number of $U(0,1)$-distributed random variables whose distribution is known and which can be computed efficiently will serve for this purpose. What really matters here is to design tests that constitute *prototypes* of simulation problems and measure different properties of random numbers. Hence, every test in our battery should represent a whole class of empirical tests. No serious effort has yet been undertaken to classify the many empirical tests available according to this principle.

There are well-established batteries of empirical tests, see [25, 44, 27]. Marsaglia's DIEHARD battery is available on CD-ROM and from the Web-server `http://stat.fsu.edu/~geo/diehard.html`.

Several test statistics have been found to be rather discriminating between random number generators. In the class of bit-oriented tests that count the number of appearances of certain blocks of bits, Marsaglia's M-tuple test is outstanding, see [44, 62, 41, 23]. Other reliable tests are the run test (see [25]) and a geometric quantity, the nearest-pair test (see [27, 56]). Recently, a discrete version of an entropy test has been presented in [35]. It is not yet clear if this interesting test is really a new prototype not covered by the M-tuple test as employed in [62]. This example shows that, while it is relatively easy to design a new empirical test for random number generators, it is a nontrivial task to show that the new quantity is a meaningful addition to the established batteries of tests and strongly different from known test statistics.

On the basis of our practical experience we recommend the following approach to test design. Suppose we use a random variable Y with known c.d.f. F_Y. With the help of a random number generator, we produce K realizations $y_1, \ldots, y_K$ of this random variable. In the second step, we compare the empirical distribution function $\hat{F}_Y$ of the samples to the target distribution F_Y by some goodness-of-fit test, like the Kolmogoroff-Smirnov (KS) statistic. This procedure is called a *two-level* test, see [27, 29]. Two-level test designs are a good compromise between speed and power of a test, see [27] for details.

If we want to test a random number generator without a particular application in mind, then it makes more sense to choose a smaller number of strongly different test statistics and to vary the parameters of the tests (like the sample size or the dimension) within large intervals. In our opinion, it is less relevant for practice to run an enormous battery of tests without any idea if all these tests really measure different properties of the generator. Further, if we do not vary the parameters enough and work, for example, with fixed sample sizes in our tests then our chances to meet the user's needs are small.

6 Practical Aspects

Several aspects of a random number generator are of practical importance. For implementation, we need *tables of parameters* for good random number generators. Without *portable implementations* a generator will not be useful to the simulation community. Power users need *large samples*. For certain generators, in particular linear types, the limit for the usable sample size is close to $\sqrt{P}$, P the period of the generator, in many empirical tests. On 32-bit machines, most software packages work with LCGs of period length below 2^{32}. Hence, the maximum usable sample size is about 2^{15}, which is much too small for demanding simulations.

Parallel simulation creates additional problems (see [1]). Even reliable generators are unsafe when submitted to parallelization techniques. Basically, there are the following methods to generate random numbers on parallel processors. We may assign (i) L different generators to L different processors, or (ii) L different substreams of one large-period generator to the L processors. Technique (ii) has two variations. Either we use (a) a "leap-frog" method where we assign the substream $(x_{nL+j})_{n\geq 0}$ to processor j, $0 \leq j < L$, or (b) we assign a whole segment $(x_n)_{n\geq n_j}$ to processor j, where $n_1, \ldots, n_L$ is an appropriate set of initial values that assures disjointness of the substreams. Technique (b) is called "splitting" of a random number generator.

Approach (i) cannot be recommended in general. There are no results on correlations between different random number generators, with one notable exception. For the EICG the correlation behavior of parallel generators has been analyzed. It was found to be remarkably good, see [51, 53].

Approach (ii) is also unsafe territory. Linear methods like the LCG or MRG may occasionally (and unexpectedly) produce terrible subsequences with the leap-frog technique. We will illustrate this point with an example from [13]. Even a good generator like $LCG(2^{48}, 55151000561141, 0)$ (see [14]) produces a leap-frog subsequence that performs even worse than RANDU in the spectral test. If we happen to assign the leap-frog subsequence $(x_{23n})_{n\geq 0}$ to one processor, then the values of the spectral test show that this was an unfortunate decision which we might regret, see Table 6.1.

$d = 2$	3	4	5	6	7	8
0.2562	**0.0600**	**0.0114**	**0.0462**	0.1275	0.2031	0.2077

Table 6.1
Spectral Test for Dimensions 2 to 8.

Splitting is not safe either. So-called *long-range correlations* are lurking in the shadows (see [4, 6]). Again, the EICG recommends itself for empirical testing, due to strong theoretical support with respect to splitting, see [11, 51, 53].

Available parallel random number generator libraries are based on linear algorithms. We refer the reader to [47, 37, 46, 55]. These libraries should be used with the above warnings in mind.

7 Examples Revisited

In the preceding sections we have discussed several aspects of a good random number generator. We will now revisit the generators we have presented in Section 3.

Examples 1,2,3, and 4 have been constructed such that super-uniformity is achieved in as many dimensions as possible. For MRG1 and cMRG1 (see Examples 1 and 2) the parameters have been chosen with the spectral test, see [34, 31]. In Example 3, theoretical analysis yielded conditions for optimal equidistribution properties. These conditions allowed to perform exhaustive searches for optimal parameters, see [32]. A similar approach based on deep theoretical analysis was used in [49] to find good tGFSR like TT800.

Inversive generators are much less sensible to the choice of parameters. They yield LIL-uniformity once maximum period is assured by the parameters.

We will now report on the performance of our examples in a stringent test, Marsaglia's M-tuple test (see [44]). The test design and the graphic presentation of the results have been developed in [62]. We refer to this thesis for details of our setup.

From every component of an overlapping d-tuple $(x_n, x_{n+1}, \ldots, x_{n+d-1})$ of random numbers $x_n \in [0,1[$, $d = 4,5$, we take the first consecutive r blocks of 4 bits in its binary representation, $r = 6,8$. Then, for a given sample size N, we compute 32 values of the -theoretically equidistributed- upper tail probability of the M-tuple test. In the following figures, the sample size ranges between 2^{18} and 2^{26}. The sample size is given in a logarithmic scale.

In Figure 7.1a, we show the result of a two-sided KS test applied to these 32 values for MRG1. Values of the KS test statistic greater than the critical value 1.59 that corresponds to the significance level of 1% are shown in dark grey and indicate that the generator has failed the test.

In Figure 7.1b, we plot the 32 equidistributed values of this test statistic. The resulting patterns should be irregular. If, for a given sample size N, the corresponding box is either totally white or black, the generator has failed. White indicates too good approximation (which is a result of super-uniformity), black signals too large deviation from the expected values. We observe that the MRG performs well for the first 6 blocks of digits of length 4 in dimensions 4 and 5, but it fails to simulate the theoretical distribution if we consider 8 blocks of 4 bits in dimension 5. The combined generator cMRG1 (see Example 2 in Section 3) yields similar results and is therefore omitted.

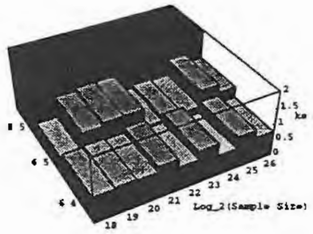


Figure 7.1a
MRG1: KS Values

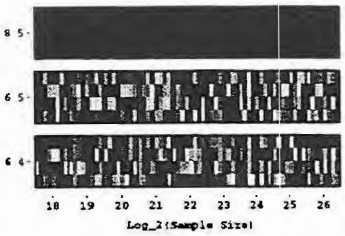


Figure 7.1b
MRG1: Upper Tail Probabilities

The cTG of Example 3 performs considerably better, as the following figures show. cTG1 has no problems with 32 bits.

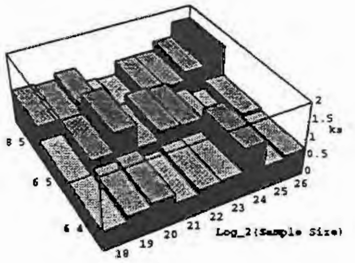


Figure 7.2a
cTG1: KS Values

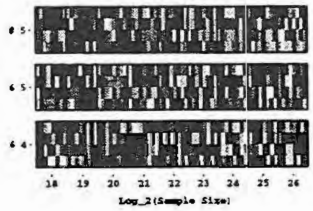


Figure 7.2b
cTG1: Upper Tail Probabilities

The tGFSR "TT800" of Example 4 gives a flawless performance. It is the only generator here without rejections.

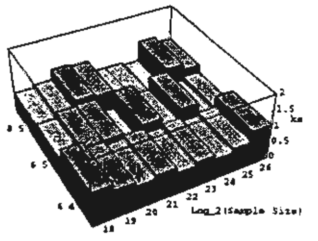


Figure 7.3a
TT800: KS Values

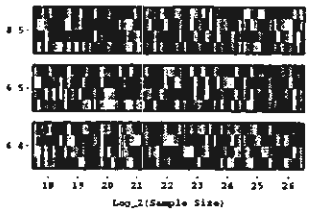


Figure 7.3b
TT800: Upper Tail Probabilities

In comparison to these long-period generators, we see that only a compound ICG of period length close to 2^{32} is able to keep up with the large generators above. cICG1 combines ICG(1031,55,1,0), ICG(1033,103,1,0), and ICG(2027,66,1,0). An ICG with period length $2^{31}-1$ like ICG1 = ICG($2^{31}-1$,1288490188,1,0) becomes "overloaded". A LCG of the same period length will perform poorly, as our example shows.

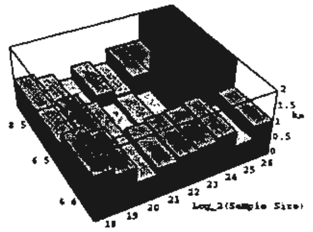


Figure 7.4a
ICG1: KS Values

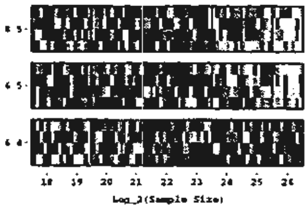


Figure 7.4b
ICG1: Upper Tail Probabilities

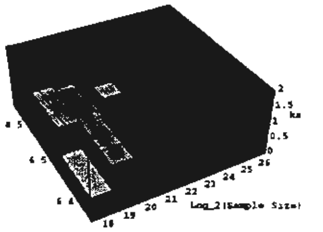


Figure 7.5a
ANSI-C: KS Values

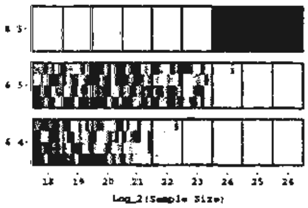


Figure 7.5b
ANSI-C: Upper Tail Probabilities

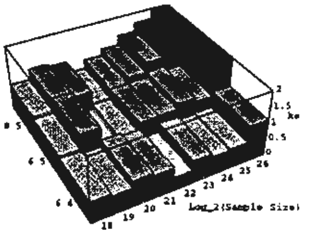


Figure 7.6a
cICG1: KS Values

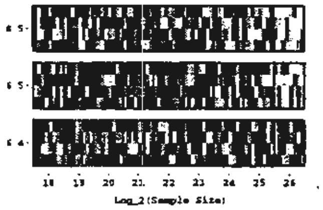


Figure 7.6b
cICG1: Upper Tail Probabilities

8 RNG Survival Kit

The following selection of papers and links allows an easy orientation in the field of random number generation.

As starting points, we recommend [29, 33], which cover a broad range of aspects. For readers that are interested in the mathematical background, [54] contains a wealth of comments and references. There are two monographs [52, 60] in this field for further reading. [25] is considered to be the "bible" of random number generation.

Numerous links to information and software can be obtained from the Web site

`http://random.mat.sbg.ac.at/`

9 RNG Checklist

Theoretical Support		
period length	conditions	
	algorithms for parameters	
structural properties	intrinsic structures	
	points on hyperplanes	
	equidistribution prop.	
correlation analysis	for particular parameters	
	for particular initializations	
	for parts of the period	
	for subsequences	
	for combinations of RNG's	

Empirical Evidence	
• variable sample size • two- or higher level tests	
bit-oriented tests	
tests for correlations	
geometric test quantities	
complexity	
transformation methods: sensitivity	

Practical Aspects	
tables of parameters available	
portable implementations available	
parallelization techniques apply	
large samples available	

10 Summary

Random number generators are like antibiotics. Every type of generator has its unwanted side-effects. There are no safe generators. Good random number generators are characterized by theoretical support, convincing empirical evidence, and positive practical aspects. They will produce correct results in many -though not all- simulations.

Open questions in this field concern reliable parallelization, the creation of good generators on demand, the sensitivity of transformation methods (to obtain nonuniform random numbers) to defects of the uniform random number generators, the classification of empirical tests, and the mathematical foundation of forecasting the empirical performance by theoretical figures of merit.

There are three rules for numerical practice that are worth to keep in mind.

1. Do not trust simulation results produced by only one (type of) generator, check the results with widely different generators before taking them seriously.

2. Do not combine, vectorize, or parallelize random number generators without theoretical and empirical support.

3. Get to know the properties of your random number generators. (We have supplied pointers and a checklist for this task)

Nowadays, the tool-box of stochastic simulation contains numerous reliable random number generators. It is up to the user to make the best out of them.

Acknowledgment. I would like to thank my research assistant Stefan Wegenkittl, who has carried out the necessary computations for the figures in Section 7.

This work has been supported by the Austrian Science Foundation, project P11143/MAT, and the Austrian Ministry of Science, Transportation, and Art, project CEI-PACT WP 5.1.2.1.2.

References

[1] S.L. Anderson. Random number generation on vector supercomputers and other advanced architectures. *SIAM Review*, **32**:221–251, 1990.

[2] A. Compagner. Operational conditions for random-number generation. *Phys. Review E*, **52**:5634–5645, 1995.

[3] R.R. Coveyou and R.D. MacPherson. Fourier analysis of uniform random number generators. *J. Assoc. Comput. Mach.*, **14**:100–119, 1967.

[4] A. De Matteis, J. Eichenauer-Herrmann, and H. Grothe. Computation of critical distances within multiplicative congruential pseudorandom number sequences. *J. Comp. Appl. Math.*, **39**:49–55, 1992.

[5] A. De Matteis and S. Pagnutti. Long-range correlations in linear and non-linear random number generators. *Parallel Computing*, **14**:207–210, 1990.

[6] A. De Matteis and S. Pagnutti. Critical distances in pseudorandom sequences generated with composite moduli. *Intern. J. Computer Math.*, **43**:189–196, 1992.

[7] L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, 1986.

[8] C. Döll. Die digitale Inversionsmethode zur Erzeugung von Pseudozufallszahlen. Master's thesis, Fachbereich Mathematik, Technische Hochschule Darmstadt, 1996.

[9] J. Eichenauer and J. Lehn. A non-linear congruential pseudo random number generator. *Statist. Papers*, **27**:315–326, 1986.

[10] J. Eichenauer-Herrmann. Explicit inversive congruential pseudorandom numbers: the compound approach. *Computing*, **51**:175–182, 1993.

[11] J. Eichenauer-Herrmann. Statistical independence of a new class of inversive congruential pseudorandom numbers. *Math. Comp.*, **60**:375–384, 1993.

[12] J. Eichenauer-Herrmann. Compound nonlinear congruential pseudorandom numbers. *Mh. Math.*, **117**:213–222, 1994.

[13] K. Entacher. A collection of selected pseudorandom number generators with linear structures. Report, The pLab Group, Dept. of Mathematics, University of Salzburg, 1996.

[14] G.S. Fishman. Multiplicative congruential random number generators with modulus 2^β: an exhaustive analysis for $\beta = 32$ and a partial analysis for $\beta = 48$. *Math. Comp.*, **54**:331–344, 1990.

[15] G.S. Fishman and L.R. Moore. A statistical evaluation of multiplicative congruential random number generators with modulus $2^{31} - 1$. *J. Amer. Statist. Assoc.*, **77**:129–136, 1982.

[16] G.S. Fishman and L.R. Moore III. An exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$. *SIAM J. Sci. Statist. Comput.*, **7**:24–45, 1986. Erratum, ibid. **7**:1058, 1986.

[17] P. Hellekalek. Inversive pseudorandom number generators: concepts, results, and links. In C. Alexopoulos, K. Kang, W.R. Lilegdon, and D. Goldsman, editors, *Proceedings of the 1995 Winter Simulation Conference*, pages 255–262, 1995.

[18] P. Hellekalek. On correlation analysis of pseudorandom numbers. Submitted to Proceedings of the Second International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Salzburg, July 9–12, 1996, 1996.

[19] P. Hellekalek and H. Leeb. Dyadic diaphony. *Acta Arith.*, 1996. To appear.

[20] P. Hellekalek and H. Niederreiter. The weighted spectral test: diaphony. 1996. Submitted to ACM Trans. Modeling and Computer Simulation.

[21] F. James, J. Hoogland, and R. Kleiss. Multidimensional sampling for simulation and integration: measures, discrepancies, and quasi-random numbers. Preprint submitted to Computer Physics Communications, 1996.

[22] B. Johnson. Radix-*b* extensions to some common empirical tests for pseudo-random number generators. To appear in ACM Trans. Modeling and Computer Simulation, 1996.

[23] K. Kankaala, T. Ala-Nissila, and I. Vattulainen. Bit-level correlations in some pseudorandom number generators. *Phys. Rev. E*, **48**:4211–4214, 1993.

[24] J. Kiefer. On large deviations of the empiric d.f. of vector chance variables and a law of the iterated logarithm. *Pacific J. Math.*, **11**:649–660, 1961.

[25] D.E. Knuth. *The Art of Computer Programming, Vol. 2.* Addison-Wesley, Reading, Mass., 1981.

[26] J. C. Lagarias. Pseudorandom numbers. *Statistical Science*, **8**:31–39, 1993.

[27] P. L'Ecuyer. Testing random number generators. In J.J. Swain et al., editor, *Proc. 1992 Winter Simulation Conference (Arlington, Va., 1992)*, pages 305–313. IEEE Press, Piscataway, N.J., 1992.

[28] P. L'Ecuyer. Bad lattice structures for vectors of non-successive values produced by some linear recurrences. 1994. To appear in ORSA J. on Computing.

[29] P. L'Ecuyer. Uniform random number generation. *Ann. Oper. Res.*, **53**:77–120, 1994.

[30] P. L'Ecuyer. Random number generators. In S. Gass and C. Harris, editors, *Encyclopedia of Operations Research and Management Science.* Kluwer Academic Publishers, 1995.

[31] P. L'Ecuyer. Combined multiple-recursive random number generators. To appear in Operations Res. **44**, 1996.

[32] P. L'Ecuyer. Maximally equidistributed combined Tausworthe generators. *Math. Comp.*, **65**:203–213, 1996.

[33] P. L'Ecuyer. Random number generation. In Jerry Banks, editor, *Handbook on Simulation.* Wiley, New York, 1997.

[34] P. L'Ecuyer, F. Blouin, and R. Couture. A search for good multiple recursive random number generators. *ACM Trans. Modeling and Computer Simulation*, **3**:87–98, 1993.

[35] P. L'Ecuyer, A. Compagner, and J.-F. Cordeau. Entropy tests for random number generators. Submitted to ACM Trans. Modeling and Computer Simulation, 1996.

[36] P. L'Ecuyer and J.-F. Cordeau. Close-point spatial tests for random number generators. draft version, 1996.

[37] P. L'Ecuyer and S. Coté. Implementing a random number package with splitting facilities. *ACM Trans. Math. Software*, **17**:98–111, 1991.

[38] P. L'Ecuyer and R. Couture. An implementation of the lattice and spectral tests for multiple recursive linear random number generators. *INFORMS J. on Comput.*, 1996. To appear.

[39] H. Leeb. Random numbers for computer simulation. Master's thesis, Institut für Mathematik, Universität Salzburg, Austria, 1995. Available from `http://random.mat.sbg.ac.at/`.

[40] H. Leeb. A weak law for diaphony. Rist++ 13, Research Institute for Software Technology, University of Salzburg, 1996.

[41] H. Leeb and S. Wegenkittl. Inversive and linear congruential pseudorandom number generators in empirical tests. To appear in ACM Trans. Modeling and Computer Simulation, 1996.

[42] O. Lendl. Explicit inversive pseudorandom numbers. Master's thesis, Institut für Mathematik, Universität Salzburg, Austria, 1996. Available from `http://random.mat.sbg.ac.at/`.

[43] N.M. MacLaren. A limit on the usable length of a pseudorandom sequence. *J. Statist. Comput. Simul.*, **42**:47–54, 1992.

[44] G. Marsaglia. A current view of random number generators. In L. Brillard, editor, *Computer Science and Statistics: The Interface*, pages 3–10, Amsterdam, 1985. Elsevier Science Publishers B.V. (North Holland).

[45] G. Marsaglia and A. Zaman. A new class of random number generators. *Ann. Appl. Prob.*, **1**:462–480, 1991.

[46] M. Mascagni, M.L. Robinson, D.V. Pryor, and S.A. Cuccaro. Parallel pseudorandom number generation using additive lagged-Fibonacci recursions. Technical report, Supercomputing Research Center, Institute for Defense Analyses, 1994.

[47] N. Masuda and F. Zimmermannn. PRNGlib: a parallel random number generator library. Technical report, Swiss Center for Scientific Computing, 1996. Available from `http://www.cscs.ch/Official/Publications.html`.

[48] M. Matsumoto and Y. Kurita. Twisted GFSR generators. *ACM Trans. Model. Comput. Simul.*, **2**:179–194, 1992.

[49] M. Matsumoto and Y. Kurita. Twisted GFSR generators II. *ACM Trans. Model. Comput. Simul.*, **4**:254–266, 1994.

[50] H. Niederreiter. Quasi-Monte Carlo methods and pseudo-random numbers. *Bull. Amer. Math. Soc.*, **84**:957–1041, 1978.

[51] H. Niederreiter. New methods for pseudorandom number and pseudorandom vector generation. In J.J. Swain et al., editor, *Proc. 1992 Winter Simulation Conference (Arlington, Va., 1992)*, pages 264–269. IEEE Press, Piscataway, N.J., 1992.

[52] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods.* SIAM, Philadelphia, 1992.

[53] H. Niederreiter. On a new class of pseudorandom numbers for simulation methods. *J. Comp. Appl. Math.*, **56**:159–167, 1994.

[54] H. Niederreiter. New developments in uniform pseudorandom number and vector generation. In H. Niederreiter and P.J.-S. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 106 of *Lecture Notes in Statistics*, pages 87–120. Springer-Verlag, New York, 1995.

[55] D.V. Pryor, S.A. Cuccaro, M. Mascagni, and M.L. Robinson. Implementation and usage of a portable and reproducible parallel pseudorandom number generator. Technical report, Supercomputing Research Center, Institute for Defense Analyses, 1994.

[56] B.D. Ripley. *Stochastic Simulation.* John Wiley, New York, 1987.

[57] K. Schaber. Digital inversive congruential generators. Master's thesis, Institut für Mathematik, Universität Salzburg, Austria, 1997. Available from `http://random.mat.sbg.ac.at/`.

[58] E. Stadlober and R. Kremer. Sampling from discrete and continuous distributions with c-Rand. In G. Pflug and U. Dieter, editors, *Simulation and Optimization*, volume 374 of *Lecture Notes in Economics and Math. Systems*, pages 154–162. Springer–Verlag, Berlin, 1992.

[59] E. Stadlober and F. Niederl. C-Rand: a package for generating nonuniform random variates. In *Compstat '94, Software Descriptions*, pages 63–64, 1994.

[60] S. Tezuka. *Uniform Random Numbers: Theory and Practice.* Kluwer Academic Publ., Norwell, Mass., 1995.

[61] I. Vattulainen, T. Ala-Nissila, and K. Kankaala. Physical models as tests of randomness. *Phys. Rev. E*, **52**:3205–3214, 1995.

[62] S. Wegenkittl. Empirical testing of pseudorandom number generators. Master's thesis, Institut für Mathematik, Universität Salzburg, Austria, 1995. Available from `http://random.mat.sbg.ac.at/`.

[63] P. Winker and K.-T. Fang. Application of threshold accepting to the evaluation of the discrepancy of a set of points. Research report, Universität Konstanz, 1995.

MODELLING AND ANALYSIS OF HYBRID SYSTEMS

Sebastian Engell

Lehrstuhl für Anlagensteuerungstechnik, Fachbereich Chemietechnik
Universität Dortmund, D-44221 Dortmund, Germany
email: engell@ast.chemietechnik.uni-dortmund.de
www: http://astwww.chemietechnik.uni-dortmund.de

Abstract. The paper discusses the practical importance of and techniques for modelling and analysis of hybrid systems. The emphasis of the discussion is on the problems encountered for systems of realistic size as they arise in technical applications and on potentially effective modelling paradigms and analysis methods.

1 Introduction

Traditionally, most of the work in modelling, simulation, and control of dynamical systems has been focused on continuous dynamic processes which can be described adequately by ordinary differential equations (ODEs) or differential-algebraic equations (DAEs). In computer science, communication theory, manufacturing, and other areas, in contrast, systems which perform transitions from one discrete state to another at certain nonpredetermined instances of time have been studied for a long time to investigate the correctness and performance of communication protocols, operating systems, transport systems, distributed computing systems, resource allocation strategies etc. In the last decade, the development of a systematic theory of dynamical systems where the dynamics consists of discrete state transitions, termed *discrete event dynamical systems* (DEDS) has started, building upon concepts and results both from computer science and mathematical systems theory [1,2].

Most technical systems, from washing machines to autonomous vehicles and chemical processing plants consist of continuous (sub-)processes which are started, parametrized, modified, reconfigured and stopped by a discrete control logic which in turn is triggered by clocks and/or sensor inputs from the the continuous processes. Moreover, the continuous processes themselves, at least on a certain level of abstraction of the model, may also exhibit transitions from one discrete state to another, e.g., backlash, stiction, collisions in mechanical systems or phase transitions in material handling and processing systems. The importance of and the growing interest in discrete phenomena in chemical process modelling and control was reflected by two invited lectures at the CPC V conference [3,4].

In process automation, the part of the control software which processes events and generates discrete outputs or switches between different procedures and strategies (e. g. monitoring of process variables and exception handling) usually is much larger than the software modules which perform (quasi-)continuous control.

Systems where continuous and discrete (i.e. discrete event) dynamics interact and their interaction determines the qualitative and quantitative behaviour are termed *hybrid dynamical systems*.

In this contribution, the modelling, simulation, and model-based analysis of hybrid systems is discussed. The focus is on the situation where continuous physical processes (plants, devices) are controlled by a switching logic, eventually in combination with continuous dynamic controllers. In the next section, an example of a rather simple hybrid technical system of this type is presented. From this example the need for model-based techniques for a systematic design and the analysis of hybrid systems is derived. In the sequel, available models of hybrid systems are reviewed and the problem of effective modelling methods for systems of realistic size is discussed. Then an overview of methods for the analysis of hybrid systems is given.

2 An illustrative example

2.1 The physical system

As an example, we consider the system sketched in fig. 1. Its purpose is to control the drawing speed in an industrial glass tube drawing process. The system consists of two identical pulling units. Each unit has two rolls, both driven by a motor. The two rolls can be moved in horizontal direction by a second motor which, by means of an elastic element, also controls the contact force on the tube in horizontal direction when the rolls make contact with the tube. The length of the tube under the pulling unit varies as from the continuously produced tube, pieces of a certain length are cut during the production process and therefore the vertical force which has to be generated by the pulling unit for constant drawing speed varies and may even change its sign. If the contact force is too low, the tube may slip. This damages the surface of the tube and, eventually, under the influence of gravity, the process may get out of control, i.e. the tube is extended and breaks, the lower part falls down, harming people and equipment. If the contact force is too high, the glass tube is crushed which has a similar potentially dangerous effect. The pullling has to be exactly vertical, but due to mechanical wear and imprecisions of the rolls, the diameters of the rolls differ and thus the drives must operate at different rotational speeds to ensure vertical pullling with minimal stress on the tube.

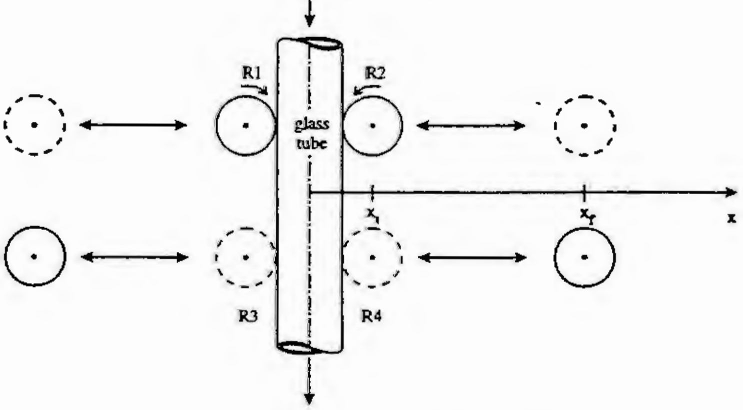


Fig. 1: Pulling unit with 4 rolls

2.2 Specification

We assume here that there are 5 binary impulse (event) inputs to the process by which its operation is controlled (which are generated by pushbuttons or from an operator console):

- "open 1" - "open 2" - "close 1" - "close 2" - "stop".

A reasonable (simplified) partial specification of the operation of this pulling unit is as follows:

a) Startup of a single unit

We assume that both drives are in their "open" position ($x = x_f$ for R2 and R4). When the signal "close 1" or the signal "close 2" is received, the chosen drive moves towards the glass tube with controlled speed until the force (which is measured) exceeds a certain threshold F_c. Then the controller of the motor which moves the rolls switches from speed control to force control. It has to be ensured that the force does not exceed a critical value F_u to avoid crushing the tube. Initially, both rolls are under speed control with an externally given set point (a main variable for quality control). When the contact force has been in the interval $F_c < f_1 < F_u$ for T_1 seconds, the drive of the slave roll of the unit which is closed must be switched into the pulling force distribution mode. In this mode, the speed of the master roll is controlled and the slave roll speed is set such that the force is split evenly between the two rolls.

b) Force control and startup of a second unit

If only one unit is active, its overall pulling force is monitored. If it exceeds a certain threshold F_{max} for more than T_2 seconds, the second unit is closed automatically in the same manner. During the closing process, the rolls of the second drive are under speed control (with the same reference value as the first drive) until the contact force reaches F_c. To minimize the damage to the tube, both rolls of the second drive are then switched to force control (the overall force now is split evenly among the four active rolls) immediately. The second drive can also be added manually by the signal "close 2" resp. "close 1".

c) Shutdown of a unit

When both drives are active and the signal "open 1" or "open 2" is received, it is checked whether the overall pulling force exceeds F_{max}. If so, the chosen drive is not opened. If the drive can be opened and both drives are active, speed control is transferred to the remaining drive and then the setpoint of the force of the drive which has to be opened is set to 0. After a timeout T_3, the motor which controls the horizontal movement is switched to speed control. After the contact force has fallen below a threshold F_{min}, the drive control of the roll motor is switched off. The rolls move with constant speed until a position sensor is triggered, then the movement is stopped.

d) Stop

When the "stop" signal is received, it is first checked whether the overall drawing force is above a critical value F_{crit} or not. If not, the signal is ignored. This ensures, that the drives are not opened when a long tube is below the rolls which may get out of control. If the drawing force is below F_{crit}, and two drives are active, drive 2 is opened first as described above. After a timeout T_4, the force is checked again, if opening of the remaining drive is possible, the same procedure as described for drive 2 is performed for drive 1 (both drives may open in parallel). If the moment is too high, the command is ignored and the alarm is triggered. If only one drive is active when the stop signal is received, the closing procedure is performed similarly if the moment is below F_{crit}.

2.3 Instrumentation

The complete unit is controlled by a programmable logic controller (PLC) which includes specialized modules for sophisticated quasi-continuous digital control algorithms for speed control and force control. The exchange of data between the modules can be realized by a variety of procedures and a priori introduces communication delays which depend on the actual state of the PLC and are not exactly known. The PLC itself of course operates in a large loop which introduces state-dependent delays as well. In addition, the sensors for force, moment and position provide signals with errors, both static and dynamic.

2.4 Design issues

The computer controlled pulling unit is a truly hybrid system. The electro-mechanical part (motors, belts with springs, glass tube) together with the continuous speed and force controllers is a continuous dynamical system with variable dynamics which are switched depending on physical conditions (how many rolls are in contact) and logical conditions which again depend on the measured variables in the system. The continuous dynamics can be discribed by different sets of differential equations for each configuration. Both the control logic and the continuous controllers strongly influence the behaviour of the system.

In the design of the control of the pulling unit, a number of choices have to be made and many parameters must be determined:

- number, type, and location of sensors
- structure of the continuous speed and force control loops
- type and parameters of the control algorithms in these loops
- values of the thresholds which are used in the logical program (which, because of the presence of measurement errors and communication delays may differ from those in the specification!)
- required processing speed of the PLC-modules
- communication procedures between the PLC modules and to and from the sensors and the motor control electronics.

Further, a logical program for switching between the different control modes must be developped.

Finally, the complete algorithm consisting of the quasi-continuous control algorithms, the logical program, and, last but not least, numerous error handling procedures (e.g., what to do if a message which transmits a sensor reading is not received for more than a certain time) must be implemented in the chosen PLC.

In a complex problem as this one, these design issues can hardly be handled without modelling and simulation. The structure and the algorithms for the isolated continuous control tasks can be designed and verified based on standard continuous models (ODEs) by simulation e.g. in MATLAB/SIMULINK. For this part of the problem, a variety of analysis and design methods and optimization algorithms are available. All the other issues are outside the scope of classical control theory. A good continuous control design alone does not guarantee success, but the interaction of e.g. the force control loop with the other elements, in particular with the discrete switching logic, under the presence of measurement errors and comunication delays, is of crucial importance. Despite the lack of theory, and often even without simulation of at least some critical situations, however, similar problems are solved by control engineers day by day in the design and implementation of control systems for real plants.

2.5 The goal: correct control by design or by verification

The goal of each engineer who designs such a system is that it works correctly in all situations and moreover is optimal for some reasonable cost function, e.g. minimal time between input signal "close x" and correct operation of the chosen drive or minimal damage to the tube surface. Thus the design should proceed in a formal manner, from "big" to "small" issues, initially based on coarse models which are then refined to include more and more aspects of the real system, with clearly specified assumptions on the elements of the problem which are simplified or neglected at each stage. E.g., one could start with the core of the switching algorithm, choose values for thresholds and timeouts using a simplified model, from this derive speficications for the behaviour of the continuous control loops, design or optimize the continuous controllers, and then include the robustness against sensor noise and communication delays. If all design steps were performed formally, based on models and supported by formal proofs of the correctnes of the design, a correct overall control would result.

A related, but slightly different approach is to assume that control algorithms (continuous and discrete) were designed more or less systematically and, in a final step, are to be *verified*, i.e. proven to be correct in all possible situations. For this, it is not sufficient to simulate some special cases, and it is unrealistic to simulate all possible sequences of inputs and disturbances from the process. Instead, all details of the control algorithms and of the behaviour of the physical plant must be modelled formally and the resulting closed-loop behaviour must be analyzed with formal methods. Then the software for the specific PLC must be generated automatically by software which also was verified formally.

To verify a discrete control program, the model of the real process and of the control software must include all relevant behaviours and effects, possibly in a simplified, worst-case manner. Also, the desired behaviour or forbidden behaviours of the system must be specified. Then, in by a formal, rigorous, automatic procedure, it must be checked, for all possible combinations of parameter values, communication delays etc. whether the controller ensures that the real process is operated according to the specification. The simplest form of a specification which can be verified is that the system will never be in a "forbidden state". Such forbidden states in our example are:

- the force control is active but the roll does not have contact to the tube
- the force exerted on the tube exceeds F_m
- the force exerted on the tube is below F_c for more than T_c seconds
- a unit is opened when the actual moment is above F_{max} resp. below F_{crit}
- a unit stops in a position where it has neither contact with the tube nor is in the terminal position.

A graphical representation of the verification approach is shown in fig. 2 [5,6]. It is desirable, that in the verification process, some parameters are left variable and bounds on the range of these parameters are determined. Then this freedom can be used to optimize the controller performance.

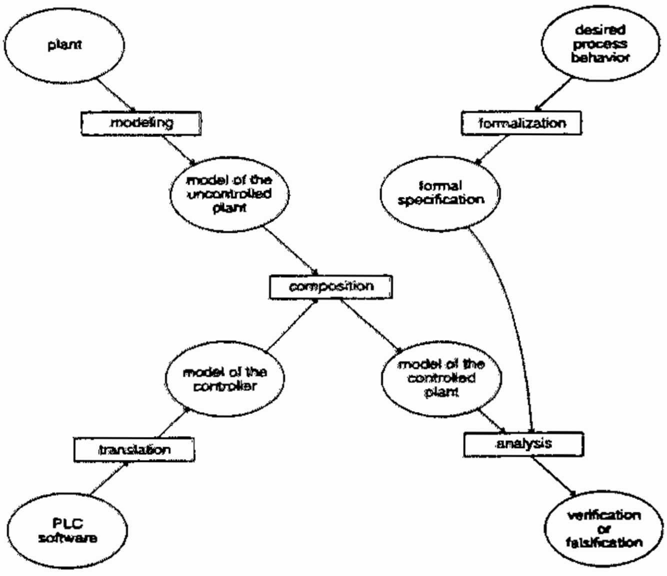


Figure 2: Formal verification of logical control systems

3 Models of hybrid systems

Mathematical models are the basis for any formal analysis of hybrid systems and for a simulation of their behaviours. Whereas for continuous systems, there is mainly one established, universal modelling paradigm, the ODE/DAE-system, for discrete systems and for hybrid systems, many modelling paradigms coexist and are useful for different purposes. We first give an overview of known models, then we discuss the problem of the combinatorical explosion and describe a modelling paradigm which may be effective for systems with both complex discrete and complex continuous dynamics.

3.1 Switched continuous systems

From a classical dynamical systems and control background, the most natural approach is to extend the standard models of continuous systems by discrete switchings of the system's dynamics.

An early attempt to provide a formal framework for the modelling and simulation of hybrid phenomena is the differential automaton introduced by Tavernini [7]. The differential automaton is an autonomous dynamical system which lives in a state space $S = \mathbb{R}^n \times Q$ where Q is a finite set of discrete states. The continuous trajectories are governed by ODEs

$$\dot{\mathbf{x}} = \mathrm{f}(\mathbf{x}, m) \tag{1}$$

and on the *switching manifolds* $g_{qp}(\mathbf{x}) = 0$, the discrete variable m switches from q to p. The trajectory of the **x**-component is continuous in this framework, and the discrete part m is piecewise constant. Under certain conditions on the switching manifolds, the system has a unique solution and infinitely fast switching cannot occur.

This model was generalized in various directions. In [8], local state spaces and jumps of the state in the case of switching were introduced as well as control inputs. Although one can argue that often discontinuous changes of the states of physical systems on the macroscopic scale are the result of simplifications in modeling, it is often helpful to include such abrupt changes (equivalent to impulsive inputs), e.g. to avoid problems with the numerical treatment of extremely different time-scales (cf. [9]). Comprehensive models which include all those mentioned were proposed by Branicki et al. [10] and Pantelides [11].

The most general switched continuous system is a DAE-system with dynamical state vector **x**, algebraic state vector **z**, continuous input vector **u**, continuous output vector **y**, mode input m, and event input vector $\boldsymbol{d}$.

The vectors **x** and **z** are real-valued, as well as **u** and **y**. m and $\boldsymbol{d}$ assume values from a finite set (alphabet, e.g. integers or symbols from a certain predefined set). m is piecewise constant and right-continuous. $\boldsymbol{d}$ is a vector event signal which is zero almost always and assumes values from its finite alphabet at a finite number of instances in a finite interval of time.

For constant mode input vector **m**, the SCB is a conventional DAE-system

$$\mathbf{f}_{\mathbf{m}}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}) = 0,\ \mathbf{x}(t_0) = \mathbf{x}_0,\ \mathbf{z}(t_0) = \mathbf{z}_0 \tag{2}$$

with continuous output

$$\mathbf{y} = \mathbf{h}_{\mathbf{m}}(\mathbf{x}, \mathbf{z}, \mathbf{u}). \tag{3}$$

The dimensions of the vectors **x** and **z** in general depend on the mode m. A mode change causes a change of the dynamics (including possibly changes of the dimensions of the variables **x** and **z**), but **x** and **z** are assumed to be continuous under mode changes, similar to the Tavernini model. Jumps of the state variables are enforced by the event input $\boldsymbol{d}$. If $\boldsymbol{d}$ is nonzero, the system is reinitialized according to

$$(\mathbf{x}(t), \mathbf{z}(t)) = \mathbf{g}(\mathbf{x}(t^-), \mathbf{z}(t^-), \mathbf{u}(t^-), m(t^-), \boldsymbol{d}(t)), \tag{4}$$

where $\mathbf{x}(t^-)$ etc. are the limits from the left of $\mathbf{x}(t)$, etc.

3.2 Discrete models

Alternative descriptions of hybrid systems can be derived from purely discrete, or logical, models. Logical models capture only the sequential, qualitative behaviour of the system's evolution, i.e. the sequence of states or transitions resp. the "language" which the system generates. There is no reference to time in the quantitative sense, but the transitions resp. events are ordered as they appear on a linear time axis.

Automata

The basic model of discrete systems is the finite state machine, or automaton. Formally, an automaton can be defined as $A=(Q,\Sigma,\delta,q_o)$, where Q is a finite set of discrete states, Σ is a set of events, δ: $Q x \Sigma \rightarrow 2^Q$ is the (partial) state transition function, and $q_o \in Q$ is the initial state. The automaton goes from one state to the next if a transition (graphically denoted by an directed arc) exists between these states and the event associated with the transition is registered. A trajectory of the automaton is a sequence of states $q_o,q_1,... \in Q$ satisfying $q_{k+1} \in \delta(q_k,\sigma_k)$ for a corresponding event sequence $\sigma_o,\sigma_1,... \in \Sigma$. A null event may be used to denote a transition which can occur without an input event. Automata are used to describe systems which accept words in a programming language, so an event corresponds to a symbol in an input sequence. If the automaton ends in an accepting state, the input was a legal expression. In this context, there is no need to specify what happens if none of the events for the outgoing transitions is registered when the system is in a given state - the input is not accepted in this case. The collection of all valid input sequences constitutes the *language* of A. Different automata can accept the same language, which is analogous to the existence of multiple state-space realizations for a given input-output behavior of a linear continuous dynamic system.

From the point of view of the analysis and controller synthesis for dynamical systems, it is more appropriate to consider an automaton as a *generator* which produces events and hence words in a language. Then, from each state, the automaton can perform a set of transitions, each of which may carry a label resp. produce an output event. For this sequence to be nontrivial, indeterminism, i.e. alternative arcs from one state to itself or to other states will usually be included. A further generalization is to specify a set of input events which trigger certain transitions in addition to the spontaneous transitions.

Statecharts [12,13]

Discrete systems resp. discrete models of continuous systems usually consist of subsystems which interact but also move from one state to the next independent of each other. The latter situation is called *concurrency* while dependencies among subsystems are termed *synchronization*. When formulating a system model, one would like to be able to connect (or *compose*) models of subsystems in a natural manner such that both the independent behaviour and the mutual dependency can easily be expressed. The weakness of automata models is that they are not very well suited for building such modular models. The standard method for combining automata models of several system components is the *synchronous composition* in which transitions occur simultaneously in the components whenever they have the same event label. Otherwise, events can occur independently without changing the states of other components. To analyse the overall system, the subsystems must be assembled in one large automaton the state space of which is the product of all local state spaces. The number of states then grows exponentially as more system components are added, and the model structure soon becomes incomprehensible.

Statecharts are intuitive graphical models which are based on the automaton model extended by

- *depth* (states can be refined by more detailed models for the behaviour in this state)
- *orthogonality* (or concurrency, i.e. the system can be in several states in parallel for which refined models can be formulated)
- *communication* between concurrent processes.

(In modern terminology, the state chart formalism clearly is "object oriented".)

A transition occurs if an associated external event occurs and may cause an action, i.e. generate an internal event which triggers another transition. After all these internal events (and the events caused by such events etc.) were processed, the system assumes its next state. For the definition and simulation of state charts, a powerful computer tool is available (*StateMate*) which is quite popular in industry.

Petri nets [14,15]

Petri nets were introduced to model concurrency, conflicts, and synchronization. A simple Petri net consists of places (circles) and transitions (bars or boxes) with arcs connecting places to transitions and transitions to places. The state of a Petri net model is given by the *marking* which identifies the number of tokens in each place. State transitions occur when Petri net transitions *fire*. In a simple Petri net, a transition *can* fire (it does not have to) when it is enabled, i.e. there is a token in each of its input places. When an enabled transition fires, a token is removed from each input place and a token is placed in each of its output places. Transitions are said to be in *conflict* if they are simultaneously enabled for a given marking, and the firing of one of the transitions disables the other transitions.

Petri nets can be combined to form large models from subsystems. This may be achieved by common Petri net transitions and places. In contrast to the synchronous composition of automata, the places and transitions of the subsystems retain their identity in the composite Petri net. Thus, the structure of the combined Petri net still reflects the logical structure of the system being modeled. The size of the Petri net grows linearly, rather than exponentially, with the number of components added to the system.

Associated with a Petri net are qualitative features of the system as *liveness*, i.e. the system does not reach a state from which no further transition is possible and *reachability*. To analyze a Petri net, one can define an automaton in which each state correspondes to a (reachable) marking in the Petri net and the state transitions correspond to the enabled Petri net transitions for each marking. This is called the reachability graph for the Petri net. When Petri net analysis is based on this construction, the economy of the Petri net representation does not carry over to savings in computation since the number of markings can grow exponentially with the size of the Petri net.

Extensions of the simple Petri net model described above to coloured Petri nets [16] provide a means to model complex systems in a relatively transparent fashion. The tokens assume attributes with valuations that can be included in the transition firing rules. However, for a formal analysis, coloured nets must be unfolded into simple nets which may become very large.

3.3 Timed models

The discrete models described in the previous section refer to time only by ordering the events (transitions) on a time axis. In many technical systems, timers are used to monitor the evolution of the continuous processes or the transmission of messages, and then the speed of the processes must be included in the analysis. This is also true for all resource allocation problems where the ressources are usually needed for a certain amount of time.

The most natural extension of Petri nets is to incorporate timers which define how long the token must have been in a state before or until a transition may fire. Alternatively, some authors associate time with the arcs which are only "open" during some interval [17]. The arrival of the token in a state initializes the clocks which are evaluated to determine the possibility of the downstream transitions to fire. For Petri nets with clocks (timed Petri nets), a reachability analysis is still feasible if special assumptions on the firing rules are made, i.e. that each enabled transition fires as soon as it is enabled. This is somewhat in conflict with the general philosophy of Petri nets to model transitions which may happen but are not enforced.

Timed automata [18] were introduced to verify the correctness of real-time systems, i.e. computer programs which have to react to inputs from the environment with timing specifications. A timed automaton is an automaton plus a set of clocks. The clocks all run at the same rate. A transition can reset one or more clocks to zero. The transitions may be conditioned not only on the occurence of events but additionaly on logical conditions the predicates of which are comparisons of the clocks with real numbers (thresholds). The language which the automaton accepts then is defined not only by sequences of symbols but also by timing constraints for the occurence of the symbols. From the generator point of view, the timed automaton produces words in a certain timed language - such a timed automaton is also called a safety timed automaton [19]. In the timed automaton model, the conditions for transitions can be represented in an alternative fashion by using *invariants*. An invariant is a logical expression, involving comparisons of clocks with real numbers, which must be satisfied for the system to enter the state and to remain in it. Putting this differently, when the invariant is no longer satisfied, the system must exit from this state. If input events and timing conditions for the transitions are also present, this may potentially cause conflicts - the model then becomes noncausal because the exiting time must be determined based on information on the future of the system or even of the inputs.

3.4 Hybrid automata [20]

From timed automata resp. timed Petri nets, it is straightforward to generalize to systems with general dynamics rather than clocks which are active in the discrete states. Hybrid automata include arbritrary continuous dynamics and discrete transitions which may depend on the development of the continuous dynamics. In a hybrid automaton, a switched continuous systems is "unfolded" into different discrete *locations*. Each location corresponds to one of the discrete modes, i.e. in each location, a specific system of differential equations is active. As for timed automata, the conditions for transitions of the discrete state from one location to another can be expressed by logical conditions and events attached to the arcs, or by invariants. Both may contain comparisons of functions of the continuous variables which appear in the dynamical equations of the states with each other or with constants. If a transition is enabled and the invariant becomes false simultaneously, the transition (resp. one of the enabled transitions if there is more than one) must be executed. Otherwise, it is possible that the system remains in the location. It is also possible to associate output events with the transitions. The initial conditions of the differential equations which are set when the discrete state reaches a location must also be defined. They may depend on all continuous and discrete state variables in the system.

Timed automata are hybrid automata where the dynamics are of the form

$$(\dot{x})_i = 1 \qquad (5)$$

and the initialization is restricted such that a continuous state variable either is reset to zero or remains unchanged. Other subclasses are linear hybrid systems where the dynamics are integrators (with different rates) and both the invariants and the conditions for the transitions involve linear expressions of the continuous variables, and integrator systems which are timed automata where the clocks can be stopped and restarted.

For systems with not too many discrete states, hybrid automata are relatively intuitive models of hybrid systems. In figs. 3 and 4, the horizontal movement of one pair of rolls and the rotational dynamics of one roll of the pulling unit described in section 2 are represented as hybrid automata. With each discrete state, a set of dynamical equations is associated which represents the continuous dynamics for this situation. For the sake of clarity of the representation, in figs. 3, 4 we adopted the convention that a transition does take place immediately when its associated condition is satisfied resp. the event occurs rather than using invariants.

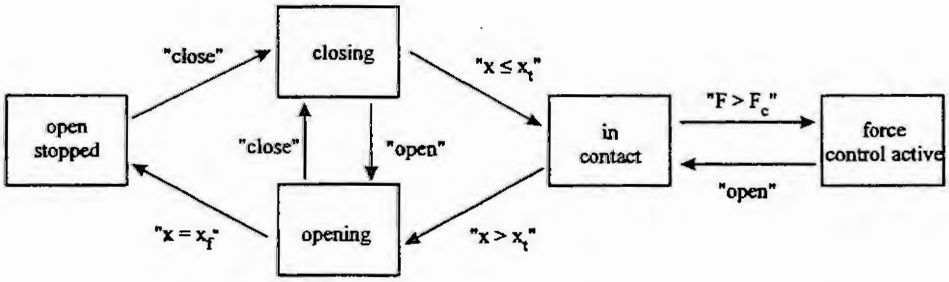


Fig. 3: States and transitions of one pair of rolls (horizontal movement)

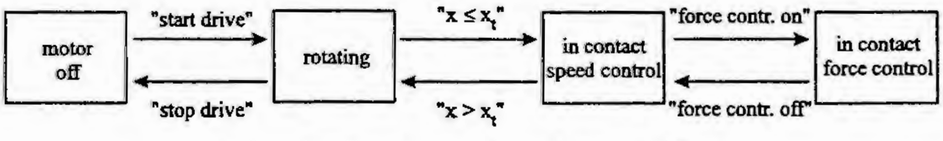


Fig. 4: States and transitions of one roll drive

Two inportant observations can be made for the example. First, it should be noted that these rather transparent submodels do not represent the full range of possible behaviours of the subsystems but only their *specified* behaviour. To illustrate the difference, a full model of the potential states of the roll drive which includes unwanted situations is shown in fig. 5. At least 7 locations and a lot more transitions than before are necessary to model both the specified and the unwanted behaviour of one roll drive.

Secondly, remember that there are two pairs of rolls and 4 roll drives. In a model of the uncontrolled system, all combinations of the discrete states of all subsystems must be included. Hence the overall system has - without any logical control - 25x256 = 6,400 locations. In each of these locations, the continuous part of the system is described by a different set of differential equations. It is obvious, that there is little hope that someone will write down 6,400 sets of differential equations and draw or program explicitly an automaton which has 6,400 locations. This problem is of course much worse if all potential behaviours are included in the submodels. Therefore both the hybrid automaton and the switched continuous system as described in section 3.1 are not appropriate to model real-world hybrid systems of even moderate complexity.

Hybrid Petri nets can be defined in a similar fashion and share the same principal problem.

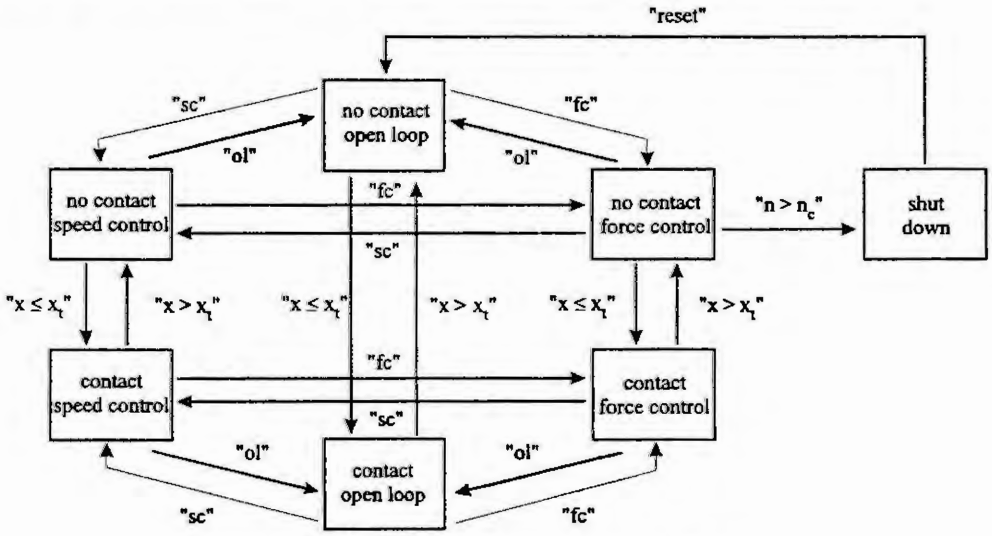


Fig. 5: Drive motor model with all possible behaviours

4 Modular models of hybrid systems

Modularity is one key ingredient of an efficient modelling paradigm for hybrid systems. In [21], a modular modeling paradigm for logic controlled continuous systems was introduced. It consists of ODE-subsystems of the Tavernini-type with an additional "event input" which can reinitialize the state vector. The logical part is described by *condition/event systems* [22]. This idea was developped further in [4,23]. Similar concepts (but not using condition/event systems) were proposed in [24,25].

We first introduce condition/event systems (*c/e-systems*) as a modular description of discrete systems where the interaction is expressed by signals which connect the subsystems, and introduce a generalized switched continuous block. With these two basic elements, hybrid systems consisting of complex discrete and continuous dynamics can be modelled by the interconncetion of blocks ("objects" if this sounds more modern).

A *c/e-system* can be described by a finite state automaton with two types of input and output signals:

- piecewise constant right continuous condition signals which assume values from a finite alphabet
- event signals which are almost always zero and assume values from a (different) alphabet.

The use of two types of signals permits a clear and transparent distinction between enabling and enforcing of state transitions. Similar signal types are used in logic programming languages.

A c/e-system has a condition input $c(t)$ and an event input $d(t)$, a condition output $p(t)$ and an event output $q(t)$ (see fig. 6).

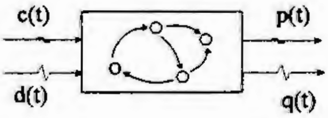


Figure 6: Condition/event system

The internal behaviour of a c/e-system is described by the *state transition function F:*

$$s(t) \in F(s(t^-), c(t^-), d(t)), \qquad (6)$$

where $s(t^-)$ denotes the limit from the left (which corresponds to the "previous" state), satisfying $\forall s \in S$, $c \in C$: $s \in F(s,c,0)$, (the system can always remain in the same state if no event occurs), and the *condition and event output functions G* and *H*

$$p(t) = G(s(t), c(t)) \qquad (7)$$

$$q(t) = H(s(t^-), s(t), d(t)), \qquad (8)$$

satisfying $\forall s \in S$: $0 = H(s,s,0)$ (event outputs only are generated if the internal state changes or an event is received). Note that the c/e-system can be nondeterministic.

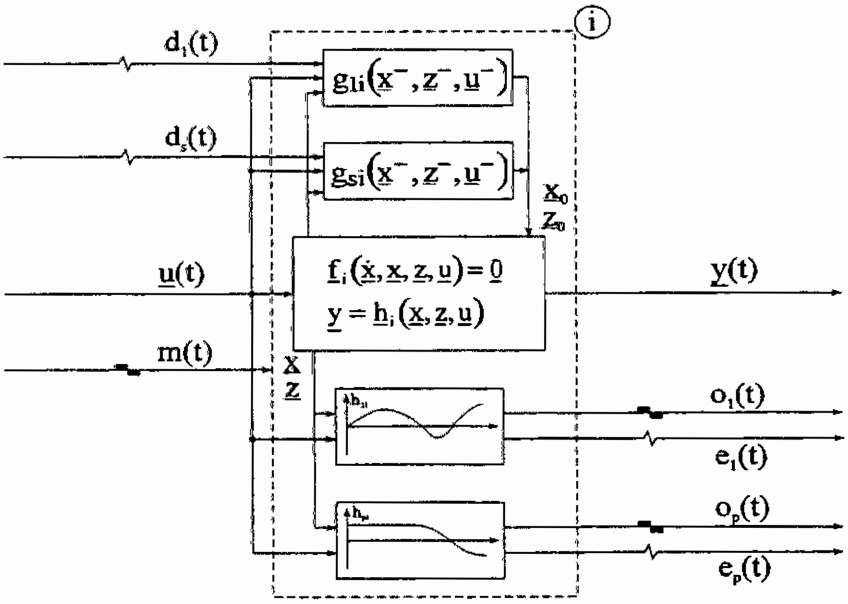
In contrast to other models, this is a modular input-output description in which all types of interactions of logical systems can be expressed conveniently, and all external quantities are defined as functions of time. Hence the coupling of c/e-systems and continuous systems is straightforward. In practice, the condition signals are used to transmit information about the state of another subsystem or the environment and to condition the transitions in the subsystem on the states of other subsystems resp. external conditions. This is not directly possible in automata models which communicate only via events which are generated as transitions occur. In order to recover the information about the state of another subsystem from the transitions, a (partial) model of that subsystem must be included. The event signals carry the information on transitions in other subsystems. In standard Petri nets only conditions for transitions depending on the state (the marking) elsewhere can be expressed and it is not straightforward to couple transitions unidirectionally, which is achieved here when an event signal enforces a transition. The introduction of both condition signals and event signals thus is very convenient for modelling. However, it is not absolutely necessary, as events can be generated by "differentiating" condition signals and, reciprocally, conditions can be created from events by suitable additional dynamics (i.e. a certain memory) in the subsystem.

To obtain a general modular modeling paradigm for hybrid systems, c/e-systems are coupled with switched DAE-systems which include threshold functions to generate condition and event outputs from continuous variables. A switched continuous block is a switched DAE-system as decribed in section 3.1 with the same inputs, the same continuous outputs and a quantized output vector *o* and an event output vector *e*.

The condition output is generated by quantization functions:

$$o_i = 1 \text{ if } k_i(x,z,u,m) \geq 0 \text{ else } o_i = 0, \qquad (9)$$

and the component e_i of the event output *e* is nonzero iff o_i changes. e_i indicates the direction of the change.



i
$d_1(t)$
$g_{1i}(\underline{x}^-, \underline{z}^-, \underline{u}^-)$
$d_s(t)$
$g_{si}(\underline{x}^-, \underline{z}^-, \underline{u}^-)$
$\underline{x}_0$
$\underline{z}_0$
$\underline{u}(t)$
$\underline{f}_i(\underline{\dot{x}}, \underline{x}, \underline{z}, \underline{u}) = \underline{0}$
$\underline{y} = \underline{h}_i(\underline{x}, \underline{z}, \underline{u})$
$\underline{y}(t)$
$m(t)$
$\underline{x}$
$\underline{z}$
h_{1i}
$o_1(t)$
$e_1(t)$
h_{pi}
$o_p(t)$
$e_p(t)$

Figure 7: Switched continuous block

The switched continuous block (SCB) may represent the physical part of the (sub)system, i.e. the (sub)process together with the discrete actuators and sensors. It is connected to the environment by physical streams, continuous or discrete (usually binary) signals and messages (events). A special case of the switched continuous block is a timer (clock). The clock has two modes, running (at a certain rate) and stopped, which are switched by the mode input. An event input resets the clock to a (possibly event-dependent) starting time x_0. A condition output indicates whether the clock is below or above a threshold, and an event output is generated at the time a threshold is reached. A clock is a condition/event system as far as the external behaviour is concerned and can thus be connected to other c/e-systems [26].

The DAE-equivalent of Tavernini's differential automaton where the dynamics change when a function of the continuous variables reaches a predefined threshold can be built from a switched continuous block and a simple c/e-system which generates the mode input from the event output (cf. figure 8). To introduce jumps, the event output and input can be coupled such that for some or all changes of the quantized variables, a reinitialization which may depend on the internal variables and the external inputs, takes place. Continuous blocks can be coupled via the continuous inputs and outputs, but also the discrete outputs of one subsystem can be connected to the discrete inputs of another subsystem.

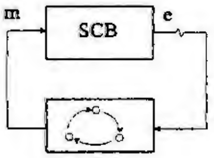


Figure 8: Tavernini's differential automation modelled by a SCB and a c/e-system

With the logical blocks, more complex logical behaviours can be modelled. Mode changes and state resets then can depend on the inner state of the logical system, and on arbitrary conditions and events in the overall system. The logical systems can be organized in layers where only the lower layer communicates with the continuous system, e.g. to represent automation hierarchies. Discrete dynamical changes can be represented locally by individual logical blocks associated with the continuous subsystems, whereas the global sequence control is represented by a central supervisory system which can also be modelled as an interconnection of timers and logical blocks. An example is shown in fig. 9.

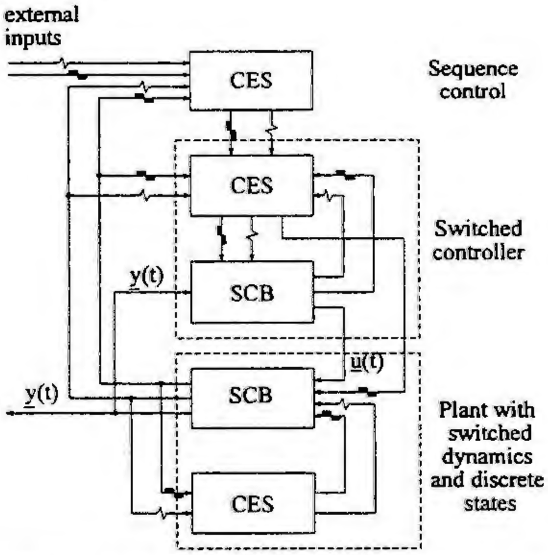


Fig. 9: Block-model of a process and a two layer control system

The problem of the explosion of the number of continuous system models which was mentioned in section 3.4 can be resolved by decomposing the continuous part of the system into subsystems which have a moderate number of discrete modes. These switched continuous blocks then interact via continuous inputs and outputs and may be coupled to logical systems which govern the mode switching. This can alternatively be viewed as one big switched continuous system which is coupled to several c/e-systems where the large number of possible dynamical behaviours is not defined explicitly but implicitly by subsystems which are switched independently. This approach is realized in the modelling environment *DYMOLA* [27].

5 Analysis of hybrid systems

5.1 Simulation

As the mathematical analysis of hybrid dynamical systems is very complex, general models which include the continuous as well as the discrete behaviour are most useful for simulation studies. It is only recently that simulation systems have become available which allow the implementation and simulation of the type of systems described above. Many simulation programs, such as *MATLAB/SIMULINK* which is frequently used for simulation of control systems, provide neither features for event-driven changes of the model structure nor for logical subsystems. The simulation of systems with switching with standard integration routines may give incorrect results or be very inefficient. A *MATLAB* routine which can handle switched ODEs was developped by Taylor [28].

DYMOLA [27] is a software package which allows modular modeling of dynamical systems including abrupt changes of the system structure and e.g. hysteresis and stick-slip effects based on bond graph models. Its strength is the symbolic manipulation of modularly defined equations to produce a minimal description of the overall system. The logical elements are transformed into algebraic equations when the simulation model is generated. It has been successfully applied to large mechanical systems with state dependent changes of the system structure. There are, however, no mechanisms provided to define and to handle complex sequences and logical decisions.

gPROMS [9,29] can handle large interconnected systems consisting of DAE-(sub)systems coupled with automata. The continuous system models are index 1 DAEs with mode changes of the Tavernini type (constant numer of differential equations, continuous state variables). On a higher level, "tasks" and "schedules", which may change the system structure including reinitialization of the state variables, can be defined.

BaSiP [30,31] is a modelling and simulation environment specifically designed to model recipe-driven batch processes. In *BaSiP,* the dynamics are described locally (in the "technical functions" of the plant) by ODEs including the generation of discrete signals. Logical systems can be defined locally (to model changes of the system structure on the subsystem level) as well as globally to perform the recipe-driven starting and stopping of operations, sequentially or in parallel. The dynamical state vector is generated *dynamically* by assembling the state vectors of the active components. In *BaSiP*, interactions of logical and continuous blocks as described above are simulated but defined only implicitly by the graphical configuration of plants and recipes.

5.2 Verification

Verification of logical programs nowadays is possible for relatively large problems due to the availability of computers with increasing speed and storage, and by the introduction of efficient data structures for representing Boolean functions [31] and the development of *symbolic model checking* [32,33] which is an effective method to search the tree of all possible state sequences of finite transition systems. The basis is a discrete state-transition model of the system dynamics. The desired properties of the system (both safety and liveness specifications) are then defined using temporal logic [34] and the program then verifies whether the properties are satisfied. Symbolic model checking has been applied in digital circuit and computer bus protocol verification. Applications of these tools to small process control examples can be found in [35,36]. *SIEMENS* has developped an inhouse tool for logical program verification based on the same approach [37].

This technique can be applied to hybrid systems if the continuous part is approximated by a purey discrete model (cf. [38-40]). In this case, however, much information is lost. Also, real logical control programs for technical systems often include timeouts etc. which cannot be adequately modelled by untimed descriptions.

At present, timed automata are the most general class of systems for which formal verification methods are available. HyTech [41], KRONOS [42] and UPPALL [43] are tools for verification of timed automata. These programs can also compute ranges of threshold values that will guarantee that the specified conditions remain valid. To date this and similar tools have however been applied only to simple examples. If a discretization of the time axis is introduced, symbolic model checking can also be extended to systems with clocks.

In order to apply these tools, the continuous part of the system must be approximated by a timed discrete system. Such an approximation in general introduces so-called spurious solutions, i.e. the discrete system includes behaviours which the continuous system cannot exhibit. Thus, if the verification shows that the controlled system can reach a forbidden state, it may turn out that this will not occur in reality. As the verfication tools also provide the sequence of events which led to the violation of the specification, a simulation of the original continuous-discrete system can be performed for the critical situations to check the result. If no violation of the specification is found by the analysis of the timed discrete closed-loop system and if the timed discrete plant model is an outer approximation of the continuous system, the control logic is proven to work correctly. There is a tradeoff between the closeness of the approximation and the granularity and the effort to derive the discrete model. The approximation of continuous systems by discrete ones is treated in a special session in this conference as well as in another plenary talk.

A verification technique based on timed c/e-systems (i.e. c/e systems with switched continuous blocks which are timers) is described in [44]. Recent work has shown that timed c/e-systems are equivalent to timed automata and can be transformed into timed automata with moderate computational effort [45]. This allows the use of the tools mentioned above for modular models of hybrid systems built from c/e-systems and timers. A tool for the verification of process control logic on the basis of timed c/e-system modelsis task is currently being developped in our group. Preliminary results with this approach were reported in [6]. The critical step here is the composition of the modules into one large system which is then the object of the verification process [46]. This is computationally very costly and moreover leads to a large, unstructured problem for verification. It seems therefore promising, to perform a *compositional* analysis where properties of small subsystems are verified and the correct functioning of the overall system is derived from these proven properties [47]. This approach is currently pursued in a joint research project on the verification of discrete process control by our group together with a computer science group.

Automatic theorem proving is another approach to the verification of logical systems in computer science [48-50]. Here, the system is described with logical *assertions*, and properties are verified as *theorems* by the proof system. The system description can be refined through the addition of more assertions. In [51] this approach was applied to a benchmark boiler control problem. A system which controls an elevator using the theorem-proving approach on-line is described in [52]. The problem with this approach so far seems to be that the complexity of the proofs explodes with the size of the problem (typically, tens or hundreds of lemmas must be formulated and are checked by the proof system for simple problems). A compositional approach here thus is necessary here as well.

The verification problem can also be put as an optimization problem where the "worst possible input" is computed which should drive the system to a forbidden state. If this can be shown to be infeasible, the correct operation is proven [53]. The attractive feature of this approach is that this can be done for the full system model and no simplification is needed. On the other hand, it leads to a large mixed integer-nonlinear programming problem which cannot be solved efficiently in general.

6 Concluding remark

Hybrid systems abund in reality and pose challenging problems in modelling, simulation and analysis - some were not treated here, e.g. the occurence of chaos in very simple hybrid systems [54,55] - which are far from being solved.

Acknowledgements

To write this paper would not have been possible without the discussions with many colleagues, in particular: Bruce Krogh (Pittsburgh), Willem-Paul de Roever, Yassine Lakhnech, and Luis Urbina (Kiel), Jim Taylor (Fredericton), Martin Otter (Oberpfaffenhofen), and Stefan Kowalewski, Ingo Hoffmann, Jörg Preussig, and Olaf Stursberg in Dortmund. The funding of the Deutsche Forschungsgemeinschaft of the focused research program (Schwerpunktprogramm) KONDISK (continuous-discrete dynamical systems) and the Graduiertenkolleg "Modelling and model-based development of complex technical systems" which supports some of the researchers mentioned above and led to productive and stimulating discussions is gratefully acknowledged.

References

[1] Cassandras, C.G.: *Discrete Event Systems: Modeling and Performance Analysis.* Irwin and Aksen, Homewood 1993

[2] Proceedings of the IEEE 77 (1989), 1 (Special Issue on Discrete Event Systems).

[3] Barton, P.I., and T. Park: Analysis and control of combined discrete/continuous systems: progress and challenges in the chemical processing industries. *Chemical Process Control V*, Tahoe City, 1996. Proceedings: CACHE Corporation (to appear).

[4] Engell, S., St. Kowalewski, and B.H. Krogh: Discrete events and hybrid systems in process control. *Chemical Process Control V*, Tahoe City, 1996. Proceeddings: CACHE Corporation (to appear).

[5] Kowalewski, St.: *Modulare diskrete Anlagenmodellierung zum systematischen Steuerungsentwurf.* Dissertation, Fachbereich Chemietechnik, Universität Dortmund, 1995. Aachen: Shaker-Verlag 1996.

[6] Kowalewski, S., R. Gesthuisen, and V. Roßmann: Model-based verification of batch process control software. *Proc. IEEE Conf. on Systems, Man, and Cybernetics 1994*, San Antonio, 331-336.

[7] Tavemini, L.: Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods & Applications* 11 (1987), 665-683.

[8] Back, A., J. Guckenheimer, and M. Myers: A dynamical simulation facility for hybrid systems. *Hybrid Systems* (*LNCS* 736), Springer, Berlin 1993, 255-267.

[9] Barton, P.I., and C.C. Pantelides: Modeling of combined discrete/continuous processes. *AIChE Journal* **40** (1994), 966-979.

[10] Branicky, M., V. Borkar, and S. Mitter: A unified framework for hybrid control. *Proc. IEEE Conf. on Decision and Control 1994*, 4228-4234.

[11] Pantelides, C. C.: Modeling, simulation and optimisation of hybrid processes. *Proceedings Workshop Analysis and Design of Event-Driven Operations in Process Systems*, Imperial College, London, 1995.

[12] Harel, D.: Statecharts: a visual formalism for complex systems. *Sci. Computer Programming* 2 (1987), 231-274.

[13] Brave, Y., and M. Heymann: Control of discrete event systems modeled as hierarchical state machines. *IEEE Trans. on Automatic Control* 38 (1993), 1803-1819.

[14] Murata, T.: Petri nets: properties, analysis, applications. *Proceedings of the IEEE* 77 (1989), 541-580.

[15] Brand, K.-P., and J. Kopainsky: Principles and engineering of process control with Petri nets. *IEEE Trans. on Automatic Control* 33 (1988), 138-149.

[16] Jensen , K.: *Coloured Petri Nets*. Springer, Berlin, 1995.

[17] Hanisch, H.-M.: Analysis of place/transition nets with timed arcs and its application to batch process control. In: *LNCS*, **691**, Springer, Berlin 1993, 282-299.

[18] Alur, R.; and D. Dill: The theory of timed automata. *Theoretical Computer Science*, **126** (1994), 183-235.

[19] Henzinger, T.A., X. Nicollin, J. Sifakis, and S. Yovine: Symbolic model checking for real-time systems. *Information and Computation,* 111 (1994), 193-244.

[20] Alur, R., C. Courcoubetis, T.A. Henzinger, and P.H. Ho: Hybrid Automata. In: *Hybrid Systems, LNCS* **736**, Springer, Berlin 1993, 209-239.

[21] Krogh, B.H.: Condition/event signal interfaces for block diagram modeling and analysis of hybrid systems. *Proc. 8th Int. Symp. on Intelligent Control Systems*, 180-185, 1993.

[22] Sreenivas, R.S., and B.H. Krogh: On condition/event systems with discrete state realizations. *Discrete Event Dynamic Systems: Theory and Applications,* **1** (1991), 209-236.

[23] Engell, S., and I. Hoffmann: Modular hierarchical models of hybrid systems. *Proc. CDC 1996*, Kobe, 142-143.

[24] Tittus, M.: *Control synthesis for batch processes*. Ph.D. Thesis, Chalmers University of Technology, Göteborg, Sweden, 1995.

[25] Antsaklis, P.J., J.A. Stiver, M. Lemmon: Hybrid system modeling and autonomous control systems, *Hybrid Systems* (*LNCS* 736), Springer, Berlin 1993, 366-392.

[26] S. Engell, St. Kowalewski, B.H. Krogh, and J. Preußig: Condition/event systems: a powerful paradigm for timed and untimed discrete models of technical systems. *Proc. EUROSIM 95*, 421-426

[27] Elmqvist, H., F.E. Cellier, and M. Otter: Object-oriented modeling of hybrid systems. *Proc. European Simulation Symposium*, Delft, 1993.

[28] Taylor, J.H.: Rigorous handling of state events in MATLAB. *Proc. IEEE Conf. on Control Applic.*, Albany, 1995.

[29] Barton, P.I.: *The Modeling and Simulation of Combined Discrete/Continuous Processes*, PhD Thesis, Univ. of London, 1992.

[30] Engell, S., and K. Wöllhaf: Dynamic simulation of batch plants. *Proc. ESCAPE-3*, Pergamon Press, 439-444, 1993.

[31] Engell, S., M. Fritz, C. Schulz, and K. Wöllhaf: Hybrid Simulation of Flexible Batch Plants. *Preprints IFAC Symposium DYCORD+ 1995*, 123-128.

[32] Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. *IEEE Trans. on Computers* 35 (1986), 677-691.

QUALITATIVE MODELLING OF DYNAMICAL SYSTEMS
Motivation, Methods, and Prospective Applications

Jan Lunze

Technische Universität Hamburg–Harburg
D-21071 Hamburg, Eissendorfer Str. 40
email lunze@tu-harburg.d400.de

Abstract. The paper describes the motivation of qualitative modelling of dynamical systems, surveys the principal lines of current research in this field and explains the main idea of an automata-theoretic approach, which has been successfully used to solve different problems of supervisory control.

1 Introduction

The modelling problem can be rather generally formulated as follows: For a given dynamical system S and a given set of questions about the behaviour B of S, find a representation M that helps to answer the given questions. Then, M is called the model of S.

This general formulation shows that the model used to solve a given problem has to be adapted to the questions to be answered. Therefore, there is no unique model but there are many different models M_i of a given system S. In a broad classification, quantitative and qualitative models have to be distinguished (Fig. 1).

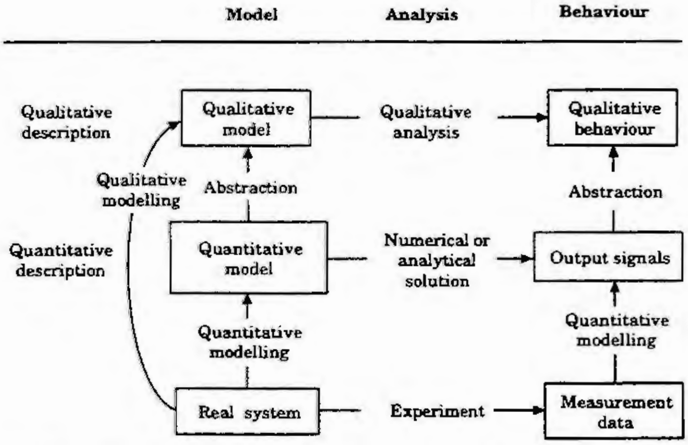


Fig. 1: Quantitative and qualitative modelling of dynamical systems

Quantitative modelling. In many engineering fields as well as in physics modelling a dynamical system means to find a set of differential or difference equations that precisely describe the system output $\boldsymbol{y}(k)$ for given input $\boldsymbol{u}(k)$ and initial state $\boldsymbol{x}_0$. Such models have the form

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{1}$$

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)), \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the system state, $\boldsymbol{u} \in \mathbb{R}^m$ the input and $\boldsymbol{y} \in \mathbb{R}^r$ the output. This model is referred to as the *quantitative model* of the system. The behaviour B is describes by pairs $(\boldsymbol{u}(t), \boldsymbol{y}(t))$ that satisfy eqns. (1), (2) for given $\boldsymbol{x}_0$.

The reason why such models dominate in engineering and physics are manifold:

- Quantitative models make it possible to precisely predict the future behaviour of the system.

- Quantitative models are compact representations where a single differential or difference equation may describe the performance of the system for a large set of input functions $\boldsymbol{u}(t)$ and initial states $\boldsymbol{x}_0$.
- Quantitative models are parametrised, i.e., they can be adjusted to different systems of a given class by simply changing the parameter values.

However, the application of such models for solving a given problem presupposes that the model together with the parameter values, the quantitative initial state $\boldsymbol{x}_0$ and input $\boldsymbol{u}(t)$ are known, and that it is really a part of the problem to *precisely* predict the behaviour of the system.

Qualitative modelling. There are many reasons why precise quantitative models are *not* a suitable representation of a given system:

- If the system is incompletely known, no precise quantitative model can be set up.
- If the inputs to the system or the initial state can be measured only roughly, the quantitative model cannot be used for prediction or simulation.
- If the system behaviour should not be precisely predicted but a qualitative assessment of the behaviour subject to a given *set* of input functions or initial states has to be found, the quantitative model is not the most suitable representation of the system.

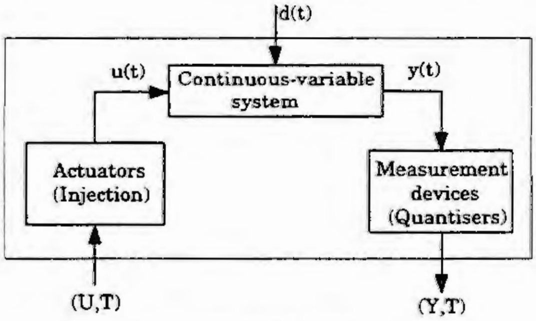


Fig. 2: Motivation for qualitative modelling

A typical situation where qualitative modelling has to be applied is shown in Fig. 2. The system under consideration can be controlled merely through a block called injection. The input to the system is given by a sequence of discrete events (U, T), where U is the name of the event and T the time instant, or by the qualitative value $[\boldsymbol{u}(t)]$. The injection block maps this event series to some input function $\boldsymbol{u}(t)$. The output $\boldsymbol{y}(t)$ is not precisely known, but merely a quantised information is available, which may be a sequence of events (Y, T) or a sequence of quantised outputs $[\boldsymbol{y}(k)]$. $\boldsymbol{d}$ is an unknown disturbance. The qualitative behaviour $[\mathcal{B}]$ is described by pairs of input and output sequences $((U, T), (Y, T))$ or $([\boldsymbol{u}(t)], [\boldsymbol{y}(t)])$.

Note that although the systems is again a continuous–variable system that can be described by some model (1), (2), a quantitative model is not adequate for describing the whole system depicted in Fig. 2 because neither the initial state and the input nor the output can be precisely measured. In this situation it is more convenient to describe the whole block shown in Fig. 2, which consists of the continuous–variable system, the injection and the quantiser.

In the following, models are called *qualitative*, if they refer to rough signal and/or parameter values. In such models the signals and parameters have typically symbolic rather than numeric values. Therefore, qualitative models refer to discrete sequences of input and output signals. Qualitative models are more abstract representations of continuous–variable systems as illustrated in Fig. 1.

Problems that can be solved by means of qualitative models. There are many engineering problems that refer to a qualitative assessment of the behaviour rather than to the quantitatively precise behaviour, as illustrated by the following problems taken from process control:

- **Process supervision**: The task is to decide whether the system performance is within prescribed tolerance bands or not.
- **Process diagnosis**: It has to be decided whether some faults occur within the system.
- **Supervisory control**: Discrete control actions have to be found by using a qualitative assessment of the current operating conditions in order to avoid safety-critical operation points and to satisfy given control aims.

Such problems can only be solved with reasonable effort if as many details of the system dynamics and the current system behaviour as possible are *neglected*. Qualitative modelling is a way to avoid the difficulties of setting up a precise quantitative model in situations where this quantitative model is not a reasonable representation of the system, and to *reduce* the complexity of the given problem.

2 Fundamentals of qualitative modelling

Qualitative values. The basic question of qualitative modelling is: Which parameter or signal values are qualitatively equivalent? That is, it has to be decided which details of the system behaviour are important for a given task and which can be neglected.

This question is, in general, answered by partitioning the input, state, and output spaces. As a typical example, Fig. 3 shows different regions in the state space of a second-order system, where the region $Q_x(2)$ may represent perferable operation points whereas $Q_x(1)$, $Q_x(3)$ and $Q_x(4)$ signify states where control actions have to be applied in order to make the system return into the 'safe' region $Q_x(1)$ and to avoid that the system reaches the safety-critical region $Q_x(5)$. Often, regular partitions of the state space are used as shown in Fig. 4, where the qualitative value of the state $[\boldsymbol{x}] = ([x_1] \;\; [x_2])'$ refers to quantisations of x_1 and x_2. For example, all states in the shaded rectangular region have the same qualitative value $(2 \;\; -1)'$.

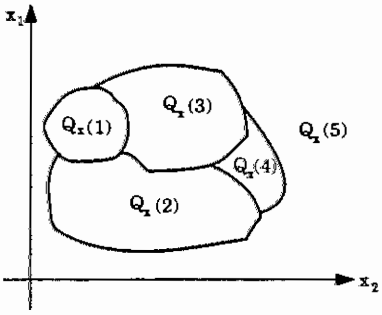


Fig. 3: Partition of the state space

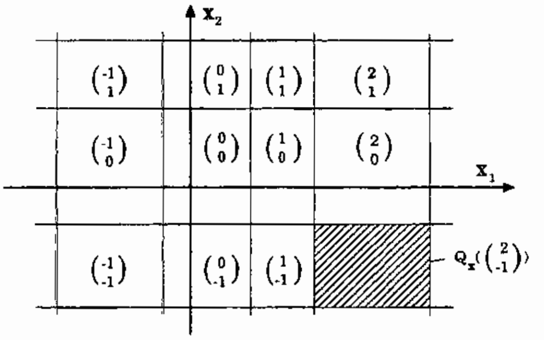


Fig. 4: Regular partition of the state space

Qualitative signals. After the signal spaces have been partitioned, a qualitative representation of signals $\boldsymbol{x}(t)$ can be obtained in two steps. First, the value $\boldsymbol{x}$ can be qualitatively described by $[\boldsymbol{x}] = z$, such that $\boldsymbol{x} \in Q_x(z)$. Second, the time axis is decomposed into different intervals where the interval bounds are given by time instants at which the signal changes its qualitative value. Both abstraction steps are illustrated in Fig. 5, where a continuous-variable signal $x(t)$ of the continuous time t is transformed into a signal with discrete values. The signal values are no longer numerical values but symbolic expressions. Instead of the continuous time t only discrete time points t_k are of interest. If these time points are numbered, the numbers can be used as qualitative value $[t]$ of t.

Qualitative modelling concerns primarily the quantisation of the signal values. Concerning the temporal abstraction two approaches can be adopted. The abstraction of time t can be made by sampling where $t_k = kT$ holds as in sampled-data systems. Alternatively, the interesting time points t_k are determined by the system itself, which is the basic assumption of the theory of discrete-event systems. After it has been defined what the qualitative signals of a given system are, is is clear that the qualitative behaviour of the system is given by input sequences $[\boldsymbol{u}([t])]$ and state or output sequences $[\boldsymbol{x}([t])]$ or $[\boldsymbol{y}([t])]$, respectively.

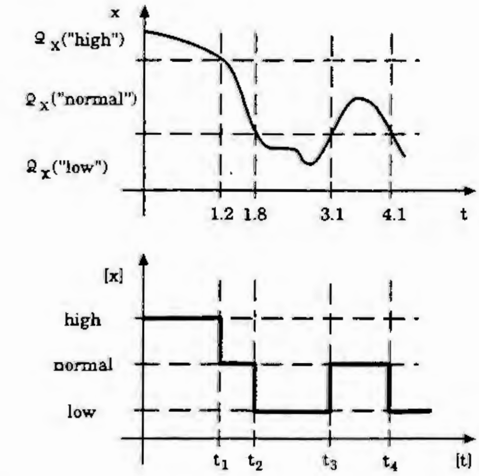


Fig. 5: Quantitative and qualitative representation of a function $x(t)$

Nondeterminism of the qualitative dynamics. One of the most important aspects of qualitative modelling is the nondeterminism of the qualitative behaviour $[\mathcal{B}]$ of the system. The reason for this can be simply explained from a system theoretic point of view. Since the *precise value* of the system state $\boldsymbol{x}(t_0)$ at a given time t_0 includes all the information that is necessary and sufficient for predicting the future behaviour $\boldsymbol{y}(t)$ $(t \geq t_0)$ of the system for *precisely* given input $\boldsymbol{u}(t)$, the trajectory $\boldsymbol{y}(t)$ *cannot* be unambiguously predicted, if only the qualitative state $[\boldsymbol{x}]$ and only the qualitative values $[\boldsymbol{u}]$ of the input are known. This nondeterminism occurs even in the qualitative behaviour described by $[\boldsymbol{y}]$. It has been investigated in detail in [13] that the qualitative behaviour is deterministic only under rather restrictive conditions. Hence, nondeterminism is a general phenomenon encountered in qualitative modelling due to the lack of information about the state and the input to the system. Consequently, qualitative models have to be nondeterministic.

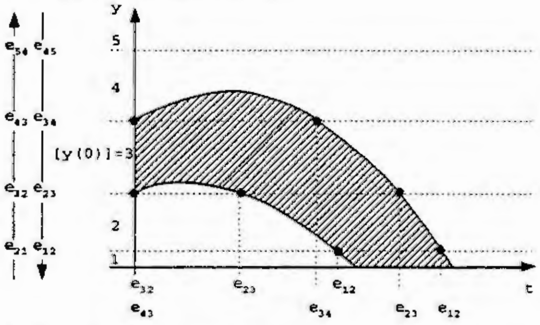


Fig. 6: Event sequences generated by a continuous–variable system

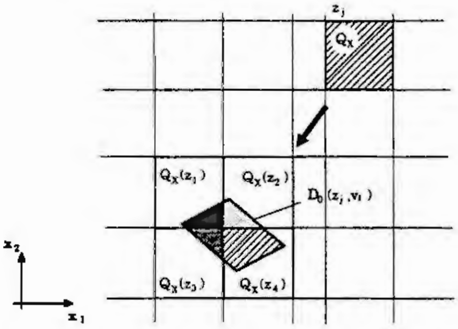


Fig. 7: Nondeterminism of the qualitative behaviour

Figs. 7 and 6 illustrate the phenomenon of nondeterminism. Since the initial state is only qualitatively known, the system may generate any trajectory of the set depicted in Fig. 6. These trajectories generate different sequences of qualitative values $[y]$, which is illustrated by the sequences of events e_{ij} describing changes of qualitative values. In Fig. 7 it is shown that if the system is dealt with as sampled–data system it moves from one state $\boldsymbol{x}(k) \in \mathcal{Q}_x$ into one successor state $\boldsymbol{x}(k+1) \in D_0$. Since D_0 overlaps with four different partitions of the state space the system may assume of the four qualitative states z_1, z_2, z_3 or z_4, which shows that the qualitative system behaviour is nondeterministic.

Basic questions of qualitative modelling. From a theoretical point of view, the basic questions that have to be solved when elaborating methods for qualitative modelling and analysis of dynamical systems are the following:

- Which concepts can be used for qualitative modelling of dynamical systems?

- How can qualitative models be set up?
- Which formal problems can be solved by means of qualitative models and what is their complexity?

From a practical viewpoint, the following questions have to be answered:

- Are qualitative models easy to be built and applied?
- What software tools can be elaborated?

The following section shows that there are preliminary answers to the theoretical questions but that a broad theoretical framework has to be elaborated yet. In the second part of the paper an automata-theoretic approach will be described which gives a partial answer to the questions raised. A stochastic automaton is used as qualitative model, which can be set up either as abstraction of a given quantitative model or by qualitative identification. It will be shown that such a qualitative model can be used to solve supervisory control problems.

3 Research directions in qualitative modelling

Approaches to qualitative modelling of dynamical systems have evolved from different fields. The following gives some entry points into the literature.

Artificial Intelligence approaches. Research work on qualitative reasoning and naive physics has brought about modelling schemes which imitate the kind of thinking that an engineer or physicist uses in order to predict the qualitative behaviour of a system [11], [19]. Typically the signal values are characterised merely as being positive, zero or negative. Logic formulas describe the relation among these signal values. Qualitative simulators have been implemented in order to solve these equations by means of qualitative algebras.

Although this reasearch direction is now more than 15 years old, industrial applications have not been really successful. The reason for this is obviously given by the fact that using simply the sign of the signal values as qualitative values ignores too much of the information that an engineer uses when reasoning about a given process.

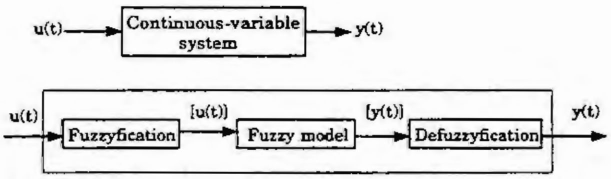


Fig. 8: Fuzzy modelling

The field of fuzzy modelling has a similar motivation. The model should be set up by using the *knowledge* about the system dynamics rather than by analysing the system in terms of first principles. Since the knowledge typically refers to qualitative states, inputs and outputs, the signal spaces are decomposed into fuzzy sets and the model set up as a set of fuzzy logic implications.

Contrary to qualitative modelling, the input and output signals are described quantitatively precisely. As illustrated in Fig. 8 the fuzzy model together with the fuzzyfication and defuzzyfication blocks represent a uniquely defined relation between the quantitative input $u(t)$ and output $y(t)$. Qualitative signal values $[u]$, $[y]$ appear only internally in the model. Hence, fuzzy modelling is a way for setting up quantitative rather than qualitative models although the internal knowledge base gives a discrete representation of the system.

Hybrid systems. Research on hybrid systems concerns phenomena that occur in the combination of continuous–variable and discrete–event systems. Several papers deal with the problem of finding a discrete control input to a continuous–variable systems by using quantised information about the current state or output, respectively. In [2] or [17] the quantitative model is used in order to analyse the hybrid control loop or to reconstruct the precise state from the quantised information.

Like qualitative modelling this research has to bridge the gap between continuous-variable and discrete-event system theory. However, the cited literature does not adopt the qualitative point of view but uses the precise quantitative model for analysis and design purposes.

Qualitative modelling and discrete event systems. The theory of discrete-event systems has brought about models that describe the event sequences generated by a given system. Events are typically given by abrupt changes of the signal values. The models have the form of automata, Petri nets, event graphs etc. If a system is described in this way, powerful analysis methods can be applied (cf, for example, [15], [16])..

Since the qualitative behaviour consists of a series of discrete events, the models of discrete-event systems theory can be applied. One important aim of qualitative modelling is to investigate in which way discrete event models can be set up for continuous-variable systems.

Different research lines can be distinguished. In [3] and [18] *deterministic* discrete-event models are used as qualitative representation. Based on these models, analysis and control principles known from continuous-variable system theory have been transfered to the discrete-event representation of the system, for example methods for controller design, state observation or optimal control.

Other authors take the *nondeterminsm* of the qualitative behaviour into account. In [12] a Petri net is used as qualitative model, which can be used for the design of sequential controllers as described in [14]. On the other hand, nondeterministic automata are used as in [1] and [13] as will be described in the next section in more detail.

4 An automata theoretic approach to qualitative modelling

4.1 The modelling problem

For a given discrete-time system (1), (2) with given quantisation of the input, state and output spaces a model has to be found that has the same input-output behaviour. That is, the system shown in Fig. 9 as a whole has to be described by some model. Note that the continuous-valued part is already assumed to be discrete in time. The quantisers Q_u, Q_x and Q_y map the signal spaces IR^m, IR^n and IR^r into the sets $\mathcal{N}_v$, $\mathcal{N}_z$ or $\mathcal{N}_z$ or integers, respectively.

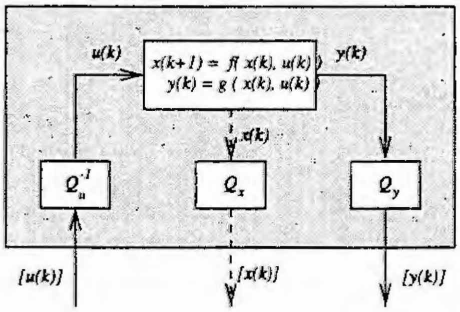


Fig. 9: The modelling problem

Since for a given qualitative initial state $[\boldsymbol{x}_0]$ and input sequence

$$[U] = ([\boldsymbol{u}(0)],\ [\boldsymbol{u}(1)], ...)$$

the qualitative behaviour is nondeterministic, the system may generate any qualitative state trajectory of the set $[\tilde{\boldsymbol{X}}([\boldsymbol{x}_0],[\boldsymbol{U}])]$ or qualitative output trajectory of the set $[\tilde{\boldsymbol{Y}}([\boldsymbol{x}_0],[\boldsymbol{U}])]$. The model should be set up in a way that it generates all these qualitative trajectories, that is the relations

$$\boldsymbol{Z}([\boldsymbol{x}_0],[\boldsymbol{U}]) \supseteq [\tilde{\boldsymbol{X}}([\boldsymbol{x}_0],[\boldsymbol{U}])] \quad (3)$$

$$\boldsymbol{W}([\boldsymbol{x}_0],[\boldsymbol{U}]) \supseteq [\tilde{\boldsymbol{Y}}([\boldsymbol{x}_0],[\boldsymbol{U}])] \quad (4)$$

have to hold where $\boldsymbol{Z}([\boldsymbol{x}_0],[\boldsymbol{U}])$ and $\boldsymbol{W}([\boldsymbol{x}_0],[\boldsymbol{U}])$ are the sets of state or output sequences of the model.

4.2 Stochastic description of the qualitative dynamics

Since the qualitative behaviour of the given system is nondeterministic, its dynamical properties can be characterised by the conditional probability distribution

$$L^*(z', w|z, v) = \text{Prob}\left(\begin{array}{c} [\boldsymbol{x}(k+1)] = z' \\ [\boldsymbol{y}(k)] = w \end{array} \middle| \begin{array}{c} [\boldsymbol{x}(k)] = z \\ [\boldsymbol{u}(k)] = v \end{array} \right) \tag{5}$$

that describes the probability that the system moves from the current state with qualitative value z under the influence of the input with qualitative value v into a state with qualitative value z' while generating an output with qualitative value w.

The qualitative model has the form of a stochastic automaton

$$S(\mathcal{N}_z, \mathcal{N}_v, \mathcal{N}_w, L(z', w|z, v)) \, , \tag{6}$$

whose behaviour relation L satisfies the condition

$$\text{ceil}\,(L(z', w|z, v)) \geq \text{ceil}\,(L^*(z', w|z, v)) \, . \tag{7}$$

where $\text{ceil}(a) = 1$ if $a > 0$ and $\text{ceil}(a) = 0$ if $a = 0$.

4.3 Qualitative identification

The question occurs whether it is possible to find a qualitative model (6) by means of experiments, in which only qualitative input and output sequences can be measured. The situation is similar to the identification of quantitative models as illustrated by Fig. 10.

Qualitative identification problem:

Given:	Qualitative input sequence $[\boldsymbol{U}] = ([\boldsymbol{u}(0)], \; [\boldsymbol{u}(1)], ...)$
	Qualitative state sequence $[\boldsymbol{X}] = ([\boldsymbol{x}(0)], \; [\boldsymbol{x}(1)], ...)$
	Signal partitions $\mathcal{Q}_x$ and $\mathcal{Q}_u$
Find:	Behaviour relation L satisfying eqn. (7).

L^* in eqn. (7) is the unknown conditional probability distribution defined in eqn. (5).

An identification algorithm is described in [8] for linear systems with $\boldsymbol{y} = \boldsymbol{x}$ and, hence, $[\boldsymbol{Y}] = [\boldsymbol{X}]$. The main idea is first to find a set of linear models

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k)$$

with interval matrices $\mathbb{A}$ and $\mathbb{B}$ and second to abstract from this set of systems a common qualitative model.

Qualitative identification algorithm

Given: The partition of the state space $\mathcal{Q}_x$ and the input space $\mathcal{Q}_u$.

1. Make experiments with the plant by applying a qualitative input sequence $[\boldsymbol{U}]$ and measure the qualitative trajectories $[\boldsymbol{X}]$.
2. Determine orthotope boundings $(\mathbb{A}, \mathbb{B})$ of the system matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.
3. Determine the hehaviour relation L such that the qualitative model satisfies eqn. (7) for all $\boldsymbol{A} \in \mathbb{A}$ and $\boldsymbol{B} \in \mathbb{B}$.

Result: Stochastic automaton S, which is a qualitative model of the system (1), (2).

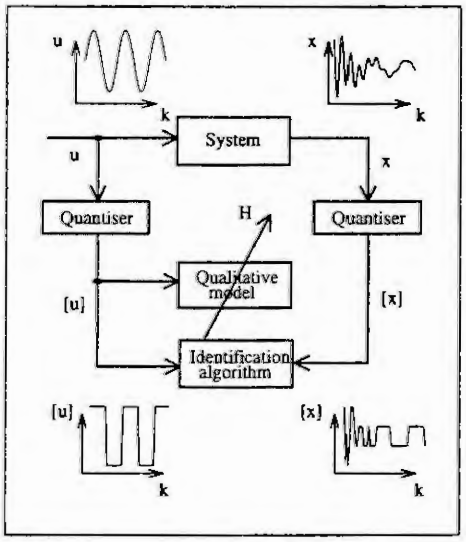


Fig. 10: Qualitative identification problem

4.4 Qualitative simulation

The qualitative model can be used to predict the future behaviour of a system. For a given qualitative initial state $[\boldsymbol{x}_0]$ and qualitative input sequence $[\boldsymbol{U}]$ the movement of the system can be described by

$$\begin{aligned}
\text{Prob}([\boldsymbol{x}(k+1)] = z') &= \sum_{z}\sum_{w} L(z', w|z, [\boldsymbol{u}(k)])\ \text{Prob}([\boldsymbol{x}(k)] = z) && (8)\\
\text{Prob}([\boldsymbol{y}(k)] = w) &= \sum_{z'}\sum_{z} L(z', w|z, [\boldsymbol{u}(k)])\ \text{Prob}([\boldsymbol{x}(k)] = z) && (9)\\
\text{Prob}([\boldsymbol{x}(0)] = z) &= \begin{cases} 1 & \text{for} \quad z = [\boldsymbol{x}_0] \\ 0 & \text{for} \quad z \neq [\boldsymbol{x}_0] \end{cases} && (10)
\end{aligned}$$

The first equation describes the state transition of the system and the second the qualitative output. It has been shown that the automaton satisfies the requirements (3) and (4) if and only if its behaviour relation satisfies eqn. (7).

5 Process supervision by means of qualitative models

The motivation of this section is given by the observation that on the level of abstraction of supervisory control continuous-variable systems are characterised by their qualitative behaviour. Supervisory control has, in general, no reference to precise quantitative values of the outputs but to regions Q_y of the output space in which the output vector currently is. The usefulness of qualitative models for solving these tasks will be illustrated by three example problems.

5.1 Observation of the qualitative state

The qualitative observation problem is similar to the classical observation problem where the current system state has to be determined from measured input and output sequences. However, it distinguished from the classical problem by two facts. First, the qualitative measurements $[\boldsymbol{y}(k)]$, $[\boldsymbol{x}(k)]$ are no longer quantitatively precise but qualitative describing some possibly large regions to which the output $\boldsymbol{y}(k)$ and the input $\boldsymbol{u}(k)$ belong. Second, the aim is to find the qualitative state $[\boldsymbol{x}(k)]$ rather than a quantitative estimate of the state $\boldsymbol{x}(k)$ (Fig. 11). Note, however, that like in the classical observation problem a continuous-variable discrete-time system (1) – (2) is the object under consideration.

Problem of qualitative state observation:

Given: Qualitative input sequence $[\boldsymbol{U}(0..T)]$
Qualitative output sequence $[\boldsymbol{Y}(0..T)]$
Find: Current qualitative state $[\boldsymbol{x}(T)]$

The observation problem can be solved by means of the following algorithm [7], where N denotes the number of qualitative states. The result is a sequence $\text{Prob}([\boldsymbol{x}(k)])$ $(k = 0, 1, ...)$ which gives an estimate of the discrete probability distribution of the qualitative state of the system.

Qualitative observation algorithm

```
Given:    Qualitative input sequence $[\boldsymbol{U}(0..T)]$,
          Qualitative output sequence $[\boldsymbol{Y}(0..T)]$,
          Qualitative model $\boldsymbol{L}_o(w|v)$.
Start:    $\text{Prob}([\boldsymbol{x}(0)] = z) = \frac{1}{N}$ for all $z \in \mathcal{N}_z$
Iterate:  $P_g(k) = \sum_z \text{Prob}([\boldsymbol{x}(k)] = z)$
          $\text{Prob}([\boldsymbol{x}(k+1)] = z') = \sum_z L(z', [\boldsymbol{y}(k)]|z, [\boldsymbol{u}(k)]) \frac{\text{Prob}([\boldsymbol{x}(k)]=z)}{P_g(k)}$
Result:   Sequence $\text{Prob}([\boldsymbol{x}(k)])$ for $k = 0, I, ...$
```

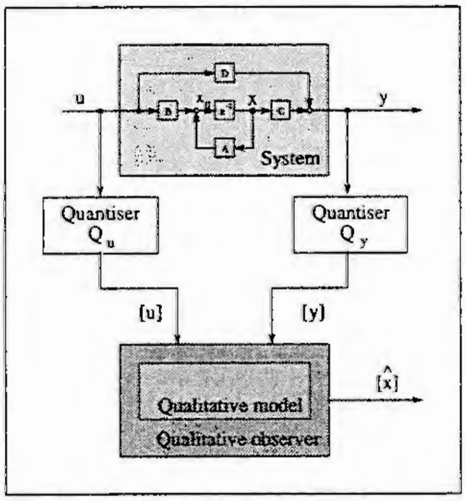


Fig. 11: Qualitative observation

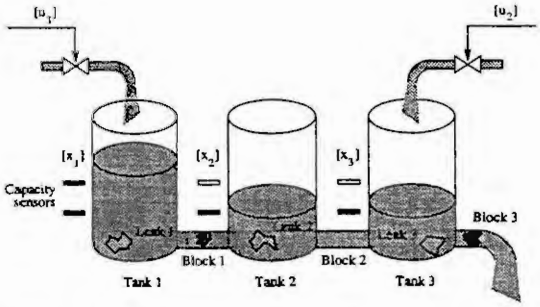


Fig. 12: Laboratory tank system

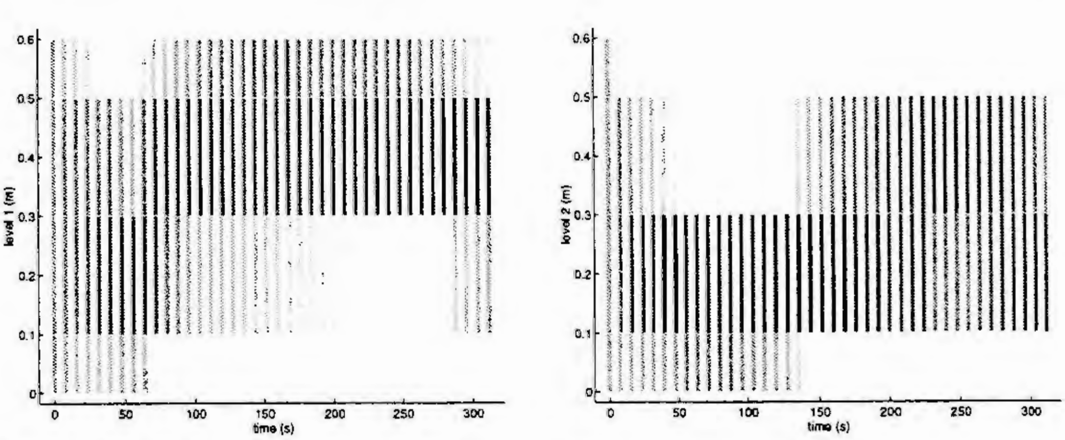


Fig. 13: Result of the qualitative observation algorithm

As an example consider the tank system shown in Fig. 12. For the observation algorithms merely the qualitative values of the inflows and of the outflow are known. The problem is to find a qualitative estimate of the levels of the three tanks. Due to the qualitative nature of the measured signals, no precise state can be reconstructed. However, the observation algorithm yields the characterisation of the tank levels by probability distributions as illustrated by Fig. 13 where the probabilities are symbolised by different grey values: the darker the bar, the more likely is the qualitative state.

5.2 Process diagnosis

Qualitative models can also be used for diagnosing faults. Assuming that the qualitative input and output sequences $[\boldsymbol{U}(0..T)]$, $[\boldsymbol{Y}(0..T)]$ are known, it can be checked whether the faultless system can generate these sequences starting in some initial state. Given the parameters a and the qualitative input and output sequences $[\boldsymbol{U}(0..T)]$ and $[\boldsymbol{Y}(0..T)]$, each qualitative state sequence $[\boldsymbol{X}(0..T)]$ can be associated with the probability $\text{Prob}\,([\boldsymbol{X}(0..T)] \mid a, [\boldsymbol{Y}(0..T)], [\boldsymbol{U}(0..T)])$ that the system follows this trajectory. Then the diagnostic problem can be formulated in terms of these possible qualitative state sequences (Fig. 14). As the qualitative state $[\boldsymbol{x}(k)]$ depends on the faults, for each measured qualitative input and output sequences a cummulative probability

$$p_a(T) = \sum_{[\boldsymbol{X}(0..T)]} \text{Prob}\,([\boldsymbol{X}(0..T)] \mid a, [\boldsymbol{Y}(0..T)], [\boldsymbol{U}(0..T)]) \tag{11}$$

for all qualitative state sequences with respect to the fault parameter $\boldsymbol{a}$ is defined. The diagnostic problem is to determine the most likely parameter

$$a^*(T) = \max_a \, p_a(T) \ , \tag{12}$$

for which the observed system behaviour is consistent with the qualitative model. The faults is considered to be also qualitative, i.e. the parameter vector belongs to a discrete set $\mathcal{A}$.

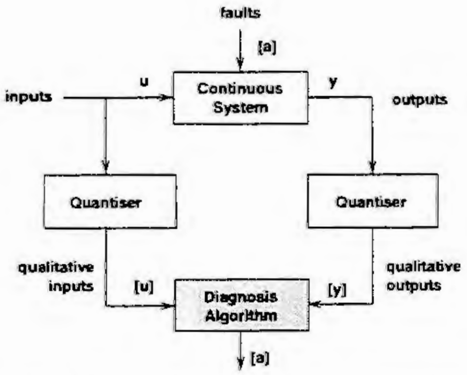


Fig. 14: Diagnosis with qualitative measurements

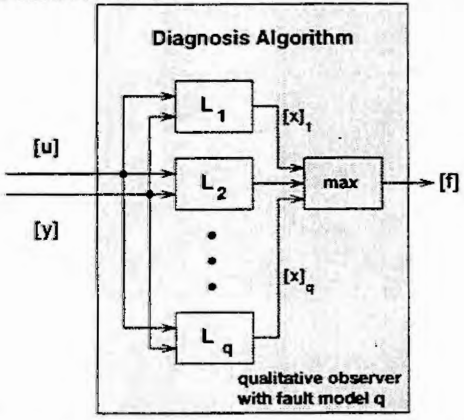


Fig. 15: Parallel qualitative observer structure

The diagnostic algorithm described in [9] applies the observation algorithm described in Section 5.1 to different models. This leads to a diagnostic system presented in Fig. 15, which consists of parallel qualitative observers for each of the faults $\boldsymbol{a} \in \mathcal{A}$.

Diagnostic algorithm

Given: Qualitative input sequence $[\boldsymbol{U}(0..T)]$,
Qualitative output sequence $[\boldsymbol{Y}(0..T)]$,
Set of qualitative models $S_1, \ldots S_q$ with behaviour relations $L_1, \ldots L_q$.

Start: Let $\text{Prob}([\boldsymbol{x}(0)] = z|a) = \frac{1}{N}$ for all $z \in \mathcal{N}_z$ and all $\boldsymbol{a} \in \mathcal{A}$

Iterate: $P_g(k,a) = \sum_z \text{Prob}([\boldsymbol{x}(k)] = z|a)$
$\text{Prob}([\boldsymbol{x}(k+1)] = z'|a) = \sum_z L_a(z', [\boldsymbol{y}(k)]|z, [\boldsymbol{u}(k)]) \, \frac{\text{Prob}([\boldsymbol{x}(k)]=z|a)}{P_g(k,a)}$

Evaluate: $p_a(k) = \sum_z \text{Prob}([\boldsymbol{x}(k)] = z|a)$
$a^*(k) = \max_a p_a(k)$

Result: Sequence of fault probabilities and the most likely fault $a^*(k)$.

$p_a(k)$ is an estimate of the probability that the fault a is present at time k.

The result of the parallel observers is given in Fig. 16. The diagnosis algorithm uses the qualitative measurements of all the state variables as indicated in Fig. 12 by the qualitative sensors. The computation of the maximum is omitted to show how the model probabilities $p_a(T)$ are changing with time. The probabilities generated by the diagnosis algorithm are drawn in greyscale: the darker the bar, the more likely the corresponding qualitative model is. The diagnostic algorithm detects the fault "blockl".

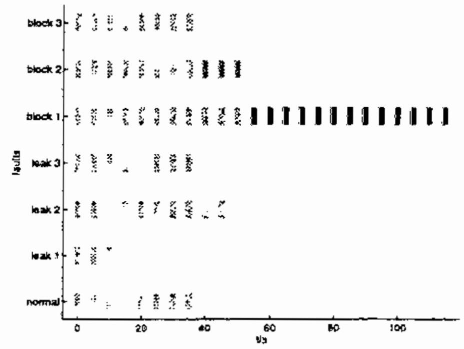


Fig. 16: Result of the dignosis

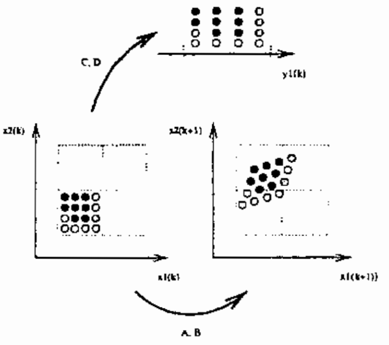


Fig. 17: Principle of the point mapping method

5.3 Qualitative control

Qualitative models are appropriate for feedback controller design if the accessible control input is given by a discrete set of input values. The qualitative controller maps the set of current qualitative states or outputs into this set of accessible inputs:

$$[u(k)] = k([y(k)]).$$

As a basis for the controller design it has been shown in [10] that the stability of the closed–loop system can be checked by means of the qualitative model of the plant and the qualitative controller. If the performance of the closed loop is described by some objective function to be minimised, optimisation methods can be used to find the control law. Since the input set consist of discrete values, the design problem is a problem of integer optimisation, which can be solved, for example, by means of branch–and–bound methods [6].

5.4 The QUAMO toolbox

The algorithms necessary for determining qualitative models, analysing the qualitative dynamics, solving observation and diagnostic problems and for designing qualitative feedback controllers have been implemented in the qualitative modelling (*QUAMO*) toolbox for MATLAB. One of the main numerical problems concern the determination of the behaviour relation L for a given system. Here, cell–mapping algorithms can be applied as described in [4]. Since the system can be nonlinear, the mapping of a grid of points covering a given partition $Q_x(i)$ of the qualitative state space by the system operator has to be investigated as illustrated by Fig. 17. The *QUAMO* toolbox is available from the author.

5.5 Conclusions

Qualitative modelling concerns the problem of describing tbe behaviour of a dynamical system without reference to precise quantiative signal values. The paper has described the motivation for using such kind of models and surveyed an automata–theoretic approach. Qualitative models are appropriate to the solution of more abstract problems like those occuring in process supervision.

Engineers are used to solve problems on different levels of abstraction. It is, therefore, an interesting challenge to investigate ways for combining quantitative and qualitative models. Then, more abstract questions like those concerning the existence of a solution to a given control problem can be answered by means of the qualitative model whereas the quantitative model can then be used to solve the control problem under real–time constraints. One way for such a combination has been described in [5] where a Petri net as qualitative model has been combined with differential equations describing the system performance in more detail if the system state x is within a given qualitative partition Q_x of the state space.

References

[1] Antsaklis, P.; Stiver, J.A.; Lemmon, M.: Hybrid system modeling and autonomous control systems, in: Grossman, R.L.; Nerode, A.; Ravn, A.P.; Rischel, H. (Eds.): *Hybrid Systems*, Springer–Verlag, Berlin 1993.

[2] Delchamps, D.F.: Stabilizing a linear systems with quantized state feedback', *IEEE Trans.* **AC–35** (1990) 916–924.

[3] Franke, D.: *Sequentielle Systeme*, Vieweg, Braunschweig 1994.

[4] Hsu, C.S.: *Cell-to-Cell Mapping*, Springer-Verlag New York, Berlin, Heidelberg, 1987.

[5] Kluwe, M., Krebs, V., Lunze, J., Richter, H.: Beratungssystem für die operative Prozeßführung, *Automatisierungstechnische Praxis* **36** (1994), 11-16.

[6] Lichtenberg, G.; Lunze, J.; Stöckel, D.: Entwurf qualitativer Regelungen mit Hilfe qualitativer Modelle, *Automatisierungstechnik* (1997).

[7] Lichtenberg, G.; Lunze, J.: Observation of qualitative states by means of a qualitative model, *Intern. Journal of Control* (1997).

[8] Lichtenberg, G.; Lunze, J.: Identification of discrete event models for continuous-variable systems, *Control '96*, Exeter, 1996, 711–715.

[9] Lichtenberg, G.; Steele, A.: An approach to fault diagnosis using parallel qualitative observers, *Workshop on Discrete Event Systems*, Edinburgh 1996, 290–295.

[10] Lunze, J.: Stabilization of nonlinear systems by qualitative feedback controllers, *Int. J. of Control* **62** (1995) 109–128.

[11] Lunze, J.: *Künstliche Intelligenz*, Band 2, Oldenbourg–Verlag, München 1995.

[12] Lunze, J.: A Petri–net approach to qualitative modelling of continuous dynamical systems, *Systems Analysis, Modelling, Simulation* **9** (1992), 89–111.

[13] Lunze, J.: Qualitative modelling of linear dynamical systems with quantized state measurements, *Automatica* **30** (1994), 417–431.

[14] Lunze, J.; Nixdorf, B.; Richter, H.: Eine Methode zur Prozeßführung kontinuierlicher Systeme auf der Basis eines qualitativen Prozeßmodells, *Automatisierungstechnische Praxis* **38** (1996), 46–54.

[15] Özveren, C. M.; Willsky, A.S.; Antsaklis P.J.: Stability and stabilizability of discrete event dynamic systems, *J. ACM* **38** (1991) 730–752.

[16] Ramadge, P.J.; Wonham W.M.: Supervisory control of a class of discrete event processes, *SIAM J. Control Optimization* **25** (1987).

[17] Raisch, J.: Analyse und Synthese einfacher hybrider Regelsysteme, *Automatisierungstechnik* **43** (1995), 224–235.

[18] Tittus, M.: *Control Synthesis for Batch Processes*, PhD–Thesis, Chalmers University of Technology, 1995.

[19] Weld, D.; deKleer J.: *Readings in Qualitative Reasoning*, Morgan Kaufman, 1990.

THE STRUCTURE AND CONSISTENCY OF THE MODEL-BASE FOR QUALITATIVE MODEL-BASED DIAGNOSIS

K.M. Hangos[1], Đ. Juričić[2], I. Gál[1]

[1]Systems and Control Laboratory, Computer and Automation Research Institute HAS
H-1518 Budapest POBox 63, HUNGARY
[2]Jožef Stefan Institute
Jamova 39, Ljubljana, SLOVENIA

Abstract. The structure and internal consistency of dynamic analytical process models is described in this paper. Moreover, the external consistency of the facets of analytical models as well as the external consistency with their qualitative counterparts is also investigated. Thereafter it is shown how such a knowledge can be used in the design and implementation of an object-oriented model-base for qualitative model-based diagnosis of chemical processes.

1. Introduction

Process modelling is a basic activity both in systems and control theory and systems engineering. Almost all areas of process control and diagnosis rely on dynamic process models. These areas include design, control, optimization and process scheduling as well as fault detection and fault diagnosis. Over the past 10 years there has been a number of projects which have attempted to develop modelling environments for different purposes, amongst which are OMOLA [1], MODEL.LA [9], ASCEND[7] and the work of Vazquez-Roman[11].

Many of these environments are concerned with the generation of model equations describing the system dynamics. These equation generation systems use some form of process description. This can be an interactive session which allows the user to draw a schematic of the process topology and then fill in details of the specific units which have been chosen. From this the differential-algebraic equations are automatically generated.

At the same time little has been done in analyzing the structure, syntax and semantics of these process models [2]. The aim of this paper is to show how such a knowledge can be used in designing and implementation of an object-oriented model-base for qualitative model-based diagnosis of chemical processes.

2. The qualitative diagnostic toolbox

The qualitative diagnostic toolbox encapsulates various qualitative model-based diagnostic methods:

- diagnosis based on coloured Petri nets (CPNs) [6],
- diagnosis based on signed directed graphs (SDGs) [8],
- diagnosis based on fault symptom trees (FSTs) [3],
- diagnosis based on multilevel flow modeling (MFM) [4].

The toolbox itself is an intelligent system consisting of the two functional parts:

- the diagnostic modules encapsulating the method dependent software modules and
- joint, method independent modules (collected in the framework of the toolbox) that serve as support to the specific diagnostic procedures.

The main elements of the qualitative model-based diagnostic toolbox are shown on Fig. 1. The oval blocks represent procedures or algorithms, while the rectangular blocks correspond to the data structures. A shadow of a block means that there are different blocks for the different diagnostic methods.

Major part of the knowledge about the process is component centered and is contained in the *Unit library*. This library consists of the set of objects associated to the process units. Each object contains different models of the unit that can be

- analytical,
- CPN models,

- SDG models,
- FSTs and
- MFMs.

In addition, the information about measurement points along with the faulty modes of the corresponding unit is buried within the object as well. The relation between the different models of the same unit (for example the relation between the SDG model and the analytical model) is described by a set of rules ensuring the consistency of the *Unit library*.

For easier manipulation, a graphical icon is associated to the object. Different objects can be connected together in order to represent different plant configurations in terms of *flowsheets*. Hence, the qualitative models to be diagnosed are built from the elements in the *Unit library*.

A flowsheet is regarded as a directed graph the vertices of which can be either units or other flowsheets whilst the edges correspond to the connections between units or other flowsheets. Consistency rules ensure correct connections between the units. Such an object-oriented representation of the process knowledge allows for easy flowsheet building from the *Flowsheet and Unit libraries*. It also ensures easy integration of new process units and their models.

The model-base of the toolbox is then *consists of the Flowsheet and Unit libraries together with rules ensuring internal and external consistency of the model elements.*

The *Model browser* serves for updating the elements of the *Unit and Flowsheet libraries*. It allows viewing and editing icons and qualitative models of the units. The user is also able to assemble, view and edit flowsheets from these components and connections.

Based on the assembled flowsheets the method specific *Model synthesis* module generates models used for qualitative fault diagnosis. Where appropriate, the diagnostic models are generated interactively by the user.

Diagnostic reasoning is performed within the *Diagnostic engines* which are also method specific. The input of the *Diagnostic engine* is provided by the *Data preprocessor* which converts either measured or simulated data into the qualitative form according to the method applied.

The information in the *Measured data files* can be viewed and edited by the *Data browser* which gives the description of the situation in which the data is measured. Also, time diagrams of the data can be displayed, e.g. to check the consistency with the results of the diagnosed faults when measured data describing, simulating a fault are used.

Diagnostic results from the *Diagnostic engine* have a unified form indicating the detection time of a fault, location of the fault and the related probability measure.

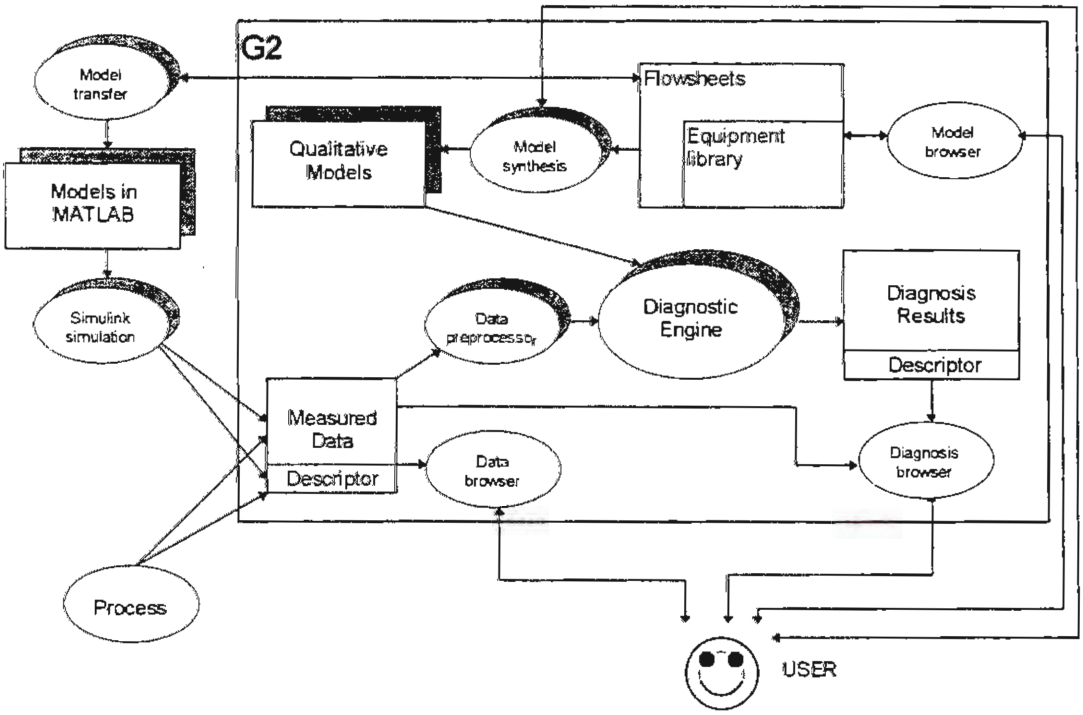


Fig. 1. The structure of the fault diagnostic system

The diagnostic toolbox also allows the transformation of the models into Matlab models by the *Model transfer* module. This module allows the user to simulate the processes in the MATLAB/Simulink environment so that the simulated data can be used as inputs for the *Diagnostic engine* through the *Data preprocessor*. Process faults simulated in this way can be used for testing the methods in the toolbox in an easy manner.

The toolbox is under construction in G2 [4].

3. The structure of process models

Process models from first engineering principles for process control and diagnosis are usually in the form of Differential-Algebraic Equation (DAE) systems. The differential part of the equations originate from lumped dynamic conservation balances and the algebraic part is of mixed origin. It can encounter

- transfer rate expressions,
- physico-chemical property relations,
- balance volume relations,
- equipment and control constraints.

The development of lumped dynamic process models follows a systematic procedure where in the first step the *finest set of mass balance volumes* are fixed. If only lumped models with initial values are considered and the physico-chemical properties in each phase are assumed to be functions of the thermodynamical state variables (temperature. pressure, composition) only then one can setup uniquely a process model, the so called *free model* over the set of the specified mass balance volumes. The structure of the free model is fixed by first engineering principles. its structural elements are identified and their internal relationships define the internal consistency of a free model.

All the other models are seen to be derived from the free model by *assumptions*, which define additional mathematical relationships or constraints between model variables, equations and parameters. Contradiction-free assumptions preserve the structural elements and the internal consistency of any derived process model.

A DAE model equation set, describing a partial model over a balance volume, can be regarded as an elementary process knowledge item, i.e. a *facet of the model-base*. Consistency between the equations and equation terms within the same facet is called *internal consistency* while inter-facet consistency is termed *external consistency*.

3.1. An example

Let us illustrate the notion of the elementary process unit class, flowsheet, the connection class, sockets and pins through an example. Figure 2. depicts the situation. We have two elementary process units (i.e. a stirred tank with inlets for hot and cold water and a valve that controls the outflow from the tank) and one connection (i.e. the connection between the tank and the valve).

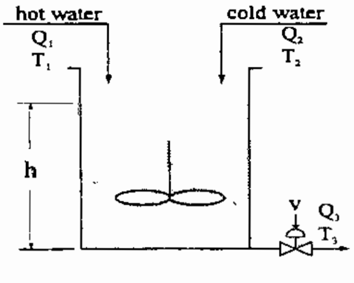


Figure 2. Stirred tank

Let us concentrate on the tank. It has two inlets (c.f. indices T1 and T2) and one outlet (c.f. index T3). Each in(out)let can be associated with three physical quantities (i.e. flow (Q), pressure (p) and temperature (T)). Relations between input and output flows and temperatures can be expressed with the following equations:

$$\begin{aligned}
\frac{dh}{dt} &= \frac{1}{A}(Q_{T1} + Q_{T2} - Q_{T3}) \\
\frac{dT_{T_3}}{dt} &= \frac{1}{Ah}(Q_{T1}(T_{T1} - T_{T3}) + Q_{T2}(T_{T2} - T_{T3}) \\
p_{T1} &= p_0 \\
p_{T2} &= p_0 \\
p_{T3} &= p_{T1} + \rho g h
\end{aligned} \tag{1}$$

3.2 Internal consistency

It is intuitively clear that the equations and the terms of the equations within a process model are not completely independent of each other but they could be related. It is usually assumed that the equations in a process model are not in formal contradiction.

Moreover, the convective and transfer terms in the mass balance equation for a given balance volume *induce convective and transfer terms* in the energy and component mass balance equations of the same balance volume, because mass and component mass flows always carry energy with them.

In the *example* the convective energy flow term $Q_{T1}T_{T1}$ in equation 1 is induced by the convective mass flow term Q_{T1} in equation 1. Moreover, the energy transfer term $Q_{T2}T_{T2}$ in equation 1 is induced by the mass transfer term Q_{T2} in equation I.

3.3 External consistency

One type of external consistency is the *connection type external consistency* which ensures that only appropriate quantities can be connected by a connection, e.g. mass type variable cannot be connected to an energy type variable. In the *example* the pressure of the valve p_{T3} can be equal only to another pressure type quantity, e.g. sum of p_{T1} plus ρgh in equation 1.

The other type of external consistency exists between two different type of models e.g. SDG and CPN. This consistency is centered around the analytical model of the system, because all qualitative models originate from it. For example, the SDG model of the system in the *example* would have the variables as its vertices and the equations would give the edges between the different vertices. E.g. p_{T3}, p_{T1} and h would be vertices and there would be edges from p_{T1} to p_{T3} and from h to p_{T3}.

4. The model-base of the toolbox

The software structure of the model-base of the toolbox is defined using a PC - based CASE tool System Architect [11], which, among others, supports Shlaer - Mellor object - oriented analysis.

The model elements are implemented as a set of classes and instances. The structure is based on the assumption that an arbitrary chemical process can be represented by a flowsheet. Every flowsheet can be defined as a directed graph, where units correspond to vertices and connections among them to edges.

Process_Unit class contains knowledge about process components and process flowsheets. Process units can be connected together by means of connection links related to *the Connection class* instances. Every process unit can be modelled by different qualitative models (MFM, SDG, FT, CPN). These qualitative models are attached to the Process_Unit class through *Qualitative_Model* class. This object class is decomposed into four subclasses that correspond to different qualitative modelling methodologies.

In order to facilitate the description and to avoid perpetual distinction between classes and instances it might be helpful to introduce the following terms:

- Elementary_PU (Elementary Process Unit, EPU),
- Complex_PU (Complex Process Unit, CPU),
- Physical Component (PC),
- Flowsheet.

Elementary_PU and *Complex_PU* are object classes; more precisely, they are subclasses of the Process_Unit class. The purpose of the *Elementary_PU* is to define the class of basic (undividable) parts of the process (e.g. classes of valves, pumps, reactors, etc.). *Complex_PU* refers to the classes of any aggregation either of Elementary_PU or other Complex_PUs. *Physical component* is an *instance* of the class EPU (e.g. concrete valve with concrete characteristic). Similarly, the flowsheet represents thc *instance* of the Complex_PU class.

Physical components (PC) are nodes in the flowsheet graph. All features of the physical components are defined by the Elementary_PU (EPU) class. Physical components are therefore modelled as offspring (i.e. subclass) of the EPU classes. Major features of PC (as the EPU class instance) are depicted in Figure 3. A socket is a point where a PC communicates with the environment. Each component has one input and one output socket. Each socket consists of zero to n pins. Each pin transfcrs one relevant physical value (e.g. flow, current, voltage, tank level, etc.). Each pin has its corresponding name. Physical components are connected among themselves through the so-called connections. There can be more than one connection associated with one socket.

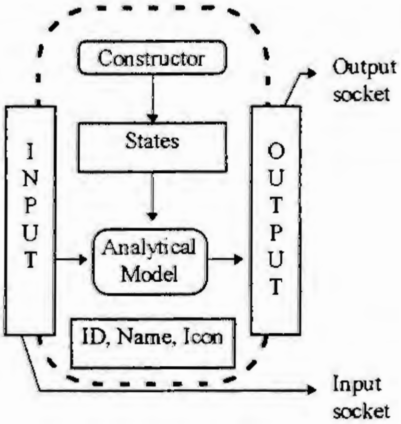


Figure 3.
Physical component (instance of Elementary_PU)

A *connection* is a link between two EPUs (e.g. EPU_1 and EPU_2). Each connection transfers one or more values. The purpose of a connection to map and mask a vector of relevant values from the EPU_1 output vector to the EPU_2 input vector. The mapping mechanism is performed through the attribute Patchboard. Most of input pins (i.e. output elements of the input process unit) map to the input pins of the output process unit. It is possible to omit some input pins, but all output pins must be used. It is possible to make several connections on the same PC socket, until only one output connection pin connects to one input socket pin.

5. Summary

The qualitative model based toolbox for the model-based diagnosis was described with the concept of the consistent model-base in the toolbox. The structure of and consistency of dynamic process models were investigated. The structure, the internal and external consistency of a dynamic analytical process model and the external consistency of the different models of a dynamic process were shown on a simple example. Then it was shown how this can be used in the qualitative diagnostic toolbox to form the object-oriented model-base to perform qualitative model-based diagnosis.

Acknowledgment . The research has been supported by the EC through COPERNICUS grant CT94-02337 and by Hungarian Research Fund through grant T017108 which is gratefully acknowledged.

6. References

1. Andersson, M., Omola - An Object Oriented Language for Model Representation, Licenciate Thesis, Department of Automatic Control, Lund University of Technology, Sweden 1990.

2. Hangos, K.M. and Cameron, I.T., A Formal Representation of Assumptions in Process Modelling, submitted to Chem. Eng. Science, 1996.

3. Lee, W.S., D.L. Grosh, F.A. Tillman and C.H. Lie Fault tree analysis, methods, and applications - a review. IEEE Transactions on Reliability, Vol. R -34, No.3., (1985), 194-203.

4. Lind, M., Modeling goals and functions of complex industrial plants. Applied Artificial Intelligence, Vol.8, (1994), 259 -283.

5. Moe, H.I., Dynamic Process Simulation. Studies on Modelling and Index Reduction, Ph.D. Thesis, University of Trondheim, 1995, Trondheim.

6. Murata, T. Petri Net: Properties, Analysis and Applications. Proceedings of the IEEE, 37, 4, (1989) 541-580

7. Piela, P. C., Epperly, T. G., Westerberg, K. M. and A. W. Westerberg., ASCEND : An Object Oriented Computer Environment for Modeling and Analysis: The Modeling Language. Comput. Chem. Engng., 15, (1991), 53.

8. Reinschke, K. J. Multivariable Control. A Graph-Theoretic Approach. In: Lecture Notes in Control and Information Sciences, 108 (1989) (Eds: Thoma M. and Wyner A.), Springer Verlag

9. Stephanopoulos, G, Henning, G and H. Leone., MODEL.LA: A Modeling Language for Process Engineering: The Formal Framework, Comput. Chem Engng., 14, 8, (1990), 813.

10. System Architect: User Guide and Reference Manual (1994). Popkin Software & Systems Incorporated.

11. Vazquez-Roman, R., Computer Aids for Process Modelling, PhD Thesis, Department of Chemical Engineering, Imperial College, 1992, London.

COMPLEXITY REDUCTION IN FUZZY MODELING

M. Setnes, R. Babuška and H.B. Verbruggen
Control Laboratory, Department of Electrical Engineering
Delft University of Technology, P.O.Box 5031, 2600GA Delft, The Netherlands
Email: m.setnes@et.tudelft.nl, Tel: +31 15 2783371, Fax: +31 15 2786679

Abstract. The interest in data driven approaches to the acquisition of fuzzy systems is increasing. Most of the approaches in the literature emphasize the global quantitative accuracy and not the transparency and interpretability of the resulting model. This paper discusses methods based on similarity analysis that, without performing additional knowledge or data acquisition, allow for the generation of fuzzy models of varying complexity. While models for simulation emphasize numerical accuracy, models for understanding the system and for operator interface are required to be transparent and interpretable. An application of the presented fuzzy modeling techniques to an air-conditioning system is described.

Introduction

Computational Intelligence techniques, such as fuzzy and neural systems, have proven to be useful in modeling of complex nonlinear systems. Both fuzzy and neural systems are recognized as universal approximators. Traditionally, a fuzzy model is built by using expert knowledge in the form of linguistic rules. Recently, there is an increasing interest in obtaining fuzzy models from measured numerical data. Different approaches have been proposed for this purpose, like fuzzy relations [13], neural network training techniques [9], and product-space clustering [2]. However, most of these approaches emphasize the global quantitative accuracy of the resulting model, and little attention is paid to linguistic and qualitative aspects, see e.g. [11] for an example.

In this paper, we discuss methods based on similarity analysis that can be applied to fuzzy models in order to obtain models of varying complexity and qualitative properties depending on the purpose of the modeling exercise. Three approaches are considered: 1) iterative compatibility analysis [1], 2) similarity relations, and 3) linguistic approximation. The approaches do not require additional knowledge or data acquisition. The user can fine-tune the numerical accuracy and transparency in order to obtain a suitable model.

The presented techniques are generally applicable to fuzzy rule-based models, and are illustrated on a fuzzy model of an air-conditioning system obtained from numerical data by means of product-space clustering. In the following sections, the used model structure and modeling method are first shortly described. Then the three approaches to model simplification are presented. Finally, the approaches are applied to the fuzzy model of the air-conditioning system, and the results are discussed with respect to accuracy, interpretability and computational load.

The Takagi-Sugeno fuzzy model

A rule-based model of the Takagi-Sugeno (TS) type [12] is considered. It consist of a set of fuzzy implications, or rules, which each describe a local input-output relation, typically in a linear form :

$$R_i : w_i(\textbf{If } x_1 \text{ is } A_{i1} \textbf{and ... and } x_n \text{ is } A_{in} \textbf{then } y_i = a_i x + b_i),\ i = 1, 2, \ldots, K. \tag{1}$$

Here R_i is the ith rule, $x = [x_1, \ldots, x_n]^T$ is the input (antecedent) variable, $A_{i1}, \ldots, A_{in}$ are fuzzy sets defined in the antecedent space, y_i is the rule output, and w_i is the rule weight. Typically, $w_i = 1, \forall i$, but it can be adjusted during the model reduction. K denotes the number of rules in the rule base, and the aggregated output of the model, $\hat{y}$, is calculated by taking the weighted average of the rule consequents:

$$\hat{y} = \frac{\sum_{i=1}^{K} w_i \beta_i y_i}{\sum_{i=1}^{K} w_i \beta_i}, \tag{2}$$

where β_i is the degree of activation of the ith rule:

$$\beta_i = \Pi_{j=1}^{n} \mu_{A_{ij}}(x_j), \quad i = 1, 2, \ldots, K, \tag{3}$$

and $\mu_{A_{ij}}(x_j) : \mathbb{R} \rightarrow [0, 1]$ is the membership function of the fuzzy set A_{ij} in the antecedent of R_i.

The structure identification is that of determining the input and the output variables of the fuzzy system. The regression matrix X and an output vector y are constructed from data measurements:

$$\mathrm{X}^T = [x_1, \ldots, x_N], \quad y^T = [y_1, \ldots, y_N]. \tag{4}$$

Here $N \gg n$ is the number of samples used for identification. The objective of identification is to construct the unknown nonlinear function $y = f(\mathrm{X})$ from the data, where f is the TS model (1).

The number of rules, K, the antecedent fuzzy sets, A_{ij}, and the consequent parameters, a_i, b_i are determined by means of fuzzy clustering in the product space of $\mathcal{X} \times \mathcal{Y}$. Hence, the data set Z to be clustered is composed from X and y:

$$\mathrm{Z}^T = [\mathrm{X}; y]\,. \tag{5}$$

Given Z and the number of clusters K, a fuzzy clustering algorithm [7, 3] is applied to compute the fuzzy partition matrix U. This provides a description of the system in terms of its local characteristic behavior in regions of the data identified by the clustering algorithm, and each cluster defines a rule [2]. Cluster validity measures can be applied to select a suitable fuzzy partition of Z [6].

The fuzzy sets in the antecedent of the rules are obtained from the partition matrix U, whose ikth element $\mu_{ik} \in [0, 1]$ is the membership degree of the data object z_k in cluster i. One-dimensional fuzzy sets A_{ij} are obtained from the multidimensional fuzzy sets defined point-wise in the ith row of the partition matrix $\mathrm{U} = [\mu_{ik}]$ by projections onto the space of the input variables x_j:

$$\mu_{A_{ij}}(x_{jk}) = \mathrm{proj}_j^{\mathbb{N}_{n+1}}(\mu_{ik}), \tag{6}$$

where proj is the point-wise projection operator [10]. The point-wise defined fuzzy sets A_{ij} are approximated by suitable parametric functions in order to compute $\mu_{A_{ij}}(x_j)$ for any value of x_j.

The consequent parameters for each rule are obtained as a weighted ordinary least-square estimate. Let $\boldsymbol{\theta}_i^T = [\boldsymbol{a}_i^T; b_i]$, let X_e denote the matrix $[\mathrm{X}; \mathbf{1}]$ and let W_i denote a diagonal matrix in $\mathbb{R}^{N \times N}$ having the weighted degree of activation, $w_i\beta_i(\boldsymbol{x}_k)$, as its kth diagonal element. If the columns of X_e are linearly independent and $w_i\beta_i(\boldsymbol{x}_k) > 0$ for $1 \leq k \leq N$, then the weighted least squares solution of $\boldsymbol{y} = \mathrm{X}_e\boldsymbol{\theta} + \epsilon$ becomes

$$\boldsymbol{\theta}_i = \left[\mathrm{X}_e^T \mathrm{W}_i \mathrm{X}_e\right]^{-1} \mathrm{X}_e^T \mathrm{W}_i \boldsymbol{y}\,. \tag{7}$$

Simplification and reduction

The transparency of fuzzy rule-based models obtained from data is often hampered by redundancy present in the form of many overlapping compatible fuzzy sets. In [1] we proposed to use a similarity measure to asses the compatibility (pair-wise similarity) of fuzzy sets in the rule base, in order to identify fuzzy sets that can be merged. Fuzzy sets estimated from data can also be similar to the universal set, adding no information to the model. Such sets can be removed from the antecedent of a rule. These operations reduce the number of fuzzy sets in the model. Reduction of the rule base follows when the premises of some rules becomes equal. Such rules are combined into one rule. In the following, we discuss three approaches to model simplification and reduction. To asses the compatibility of fuzzy sets we apply the fuzzy analog to the Jaccard index [4]:

$$c_{jlm} = \frac{\mid A_{lj} \cap A_{mj} \mid}{\mid A_{lj} \cup A_{mj} \mid}, \tag{8}$$

where $l, m = 1, 2, \ldots, K$, and $c_{jlm} \in [0, 1]$. The $\cap$ and $\cup$ operators are the intersection and the union, respectively, and $\mid \cdot \mid$ denotes the cardinality of a fuzzy set. The measure c_{jlm} quantifies the compatibility between the fuzzy sets A_{lj} and A_{mj} in the rules R_l and R_m, respectively.

Iterative compatibility analysis. This approach is based on iterative merging of compatible fuzzy sets [1]. It requires two thresholds from the user, $\lambda, \gamma \in (0, 1)$ for merging compatible fuzzy sets and removing fuzzy sets compatible with the universal set, respectively. In each iteration, the compatibility between all fuzzy sets in each antecedent dimension is analyzed. The pair of fuzzy sets having the highest compatibility $c > \lambda$ are merged. A new fuzzy set is created by merging and the rule base is updated by substituting this fuzzy set for the ones merged. The algorithm again evaluates the updated rule base, until there are no more fuzzy sets for which $c > \lambda$. Fuzzy sets compatible with the universal set are removed from the rules in which they occur. The algorithm is given in Table 1a.

Similarity relations. Also this approach requires two thresholds λ and γ. For each antecedent dimension, $j = 1, \ldots n$, a similarity relation between the fuzzy sets is obtained in two steps: First a $K \times K$ binary fuzzy *compatibility relation* $C_j = [c_{jlm}]$ is calculated, whose elements are obtained by (8). C_j is reflexive and symmetric. Second, a *similarity relation*, S_j, is calculated as the max-min transitive closure, C_{Tj}, of C_j [8]:

1. $C'_j = \max(C_j(C_j \circ C_j))$.
2. If $C'_j \neq C_j$, set $C_j = C'_j$ and go to 1.
3. Stop: $C_{Tj} = C'_j$, set $S_j = C_{Tj}$.

Here $\circ$ is the max-min composition. The lmth element of S_j, $[s_{jlm}]$, gives the similarity between A_{jl} and A_{jm}. For each antecedent dimension, the fuzzy sets having similarity $s_{jlm} > \lambda$ are merged. Fuzzy sets compatible with the universal set are removed. The algorithm is given in Table 1b.

Table 1: Two algorithms for fuzzy rule base simplification

Given a rule base $\mathbf{R} = \{R_i \mid i = 1, \ldots, K\}$, where R_i is given by (1). Select the thresholds $\lambda, \gamma \in (0, 1)$:

```
Repeat:
  Step 1: Select the two most compatible fuzzy sets:
    $A_{Lj} = \{(A_{lj}, A_{mj}) \mid c_{jlm} = \max_{\substack{i \neq p \\ j=1,\ldots,n \\ i,p=1,\ldots,K}} (c_{jip})\}.$
  Step 2: Merge selected fuzzy sets:
    If $c_{jlm} > \lambda$ merge $A_{lj}$ and $A_{mj}$,
      $A_{cj} = MERGE(\mathbf{A}_{Lj})$,
      set $A_{lj}, A_{mj} = A_{cj}$.
Until: $c_{jlm} < \lambda$.
Step 3: Remove fuzzy sets similar to universal set:
  For $j = 1, 2, \ldots, n$, calculate
    $c_{ij} = \frac{| A_{ij} \cap U_j |}{| A_{ij} \cup U_j |}, \quad i = 1, 2, \ldots, K,$
  where $\mu_{U_j} = 1, \forall x_j$.
  If $c_{ij} > \gamma$, remove $A_{ij}$ from the antecedent of $R_i$.
```

(a) Iterative compatibility analysis

```
Repeat for $j = 1, 2, \ldots, n$:
  Step 1: Calculate similarity relation:
    $\mathrm{C}_j = [c_{jlm}], \quad l, m = 1, 2, \ldots, K,$
    $\mathrm{S}_j = [s_{jlm}] = \mathrm{C}_{Tj}.$
  Step 2: Merge similar fuzzy sets:
    $\mathbf{A}_{Lj} = \{A_{lj} \mid s_{jlm} > \lambda, l \neq m\},$
    $A_{cj} = MERGE(\mathbf{A}_{Lj}),$
    $\forall A_{lj} \in \mathbf{A}_{Lj}$, set $A_{lj} = A_{cj}$.
  Step 3: Remove fuzzy sets similar to universal set:
    $c_{ij} = \frac{| A_{ij} \cap U_j |}{| A_{ij} \cup U_j |}, \quad i = 1, 2, \ldots, K,$
    where $\mu_{U_j} = 1, \forall x_j$.
    If $c_{ij} > \gamma$, remove $A_{ij}$ from the antecedent of $R_i$.
```

(b) Similarity relations

The first approach merges only one pair of fuzzy sets per iteration and the rule base is updated between iterations. The second approach merges all similar fuzzy sets per dimension simultaneously. The use of the transitive similarity relation gives different results than the iterative approach. Merging of fuzzy sets is accomplished by letting the support of the union of the sets in $\mathbf{A}_{Lj}$ be the support of the new fuzzy set A_{cj}. This guarantees completeness of the antecedent space. The kernel of A_{cj} is given by averaging the kernels of the sets in $\mathbf{A}_{Lj}$.

If the antecedents of $p \geq 2$ rules becomes equal, the p rules can be replaced by *one* common rule R_c. The consequent parameters of the reduced rule base can be re-estimated from training data (7), or one can calculate the parameters of R_c from the parameters of the p removed rules. The latter does not depend on the availability of data. This approach is now described: Let $Q \subset \{1, 2, \ldots, K\}$ be an subset of rule indices such that $A_{lj} = A_{mj}, j = 1, 2, \ldots, n, \forall l, m \in Q$. $\mathbf{R}_Q$ then denotes the set of rules with equal antecedents. The rule R_c replaces $\mathbf{R}_Q$, and its antecedent part equals that of $\mathbf{R}_Q$, i.e. $A_{cj} = A_{lj}, j = 1, 2, \ldots, n, l \in Q$. R_c accounts for $\mathbf{R}_Q$ by weighting R_c with the total weight of $\mathbf{R}_Q$, $w_c = \sum_{i \in Q} w_i$, and its consequent is an average of the consequents of $\mathbf{R}_Q$. Thus, the set of rules $\mathbf{R}_Q$ is represented by a single rule R_c with weight w_c and consequent parameters

$$\boldsymbol{\theta}_c = \frac{1}{w_c} \sum_{i \in Q} w_i \boldsymbol{\theta}_i. \tag{9}$$

Let $\overline{Q} = \{1, \ldots, K\} - Q$, the model output (2) now becomes

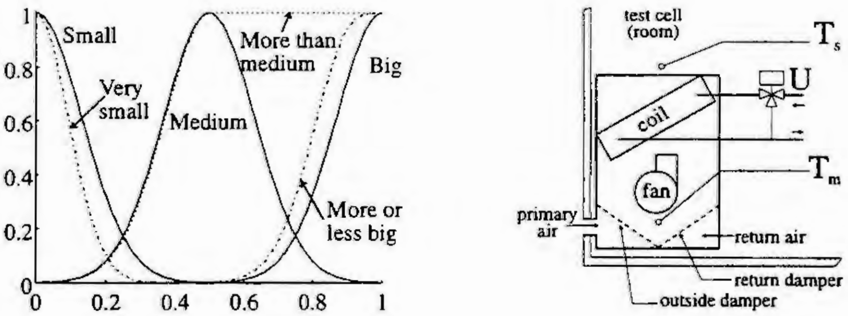
$$\hat{y} = \frac{\sum_{i \in \overline{Q}} w_i \beta_i y_i + w_c \beta_c y^r}{\sum_{i \in \overline{Q}} w_i \beta_i + w_c \beta_c}. \tag{10}$$

This substitution of $\mathbf{R}_Q$ by R_c does not alter the input-output mapping of the TS-model (1).

Linguistic approximation. A fuzzy model can be interpreted by means of linguistic approximation [5]. Using (8), the fuzzy sets in the model are compared to reference fuzzy sets and their modifications by linguistic hedges. The model is described in terms of the labels of the reference fuzzy sets. By substituting the reference fuzzy sets for the original fuzzy sets, the model can be directly interpreted linguistically as well as verified in simulation. Three reference fuzzy sets are used in our example, 'Small', 'Medium' and 'Big', shown in Fig. 1a, together with the linguistic hedges in Table 2.

Table 2: Linguistic hedges.

linguistic hedge	operation	linguistic hedge	operation
very A	μ_A^2	More than A	$\begin{cases} \mu_A(x), & \text{if } x < \min\{x \mid \mu_A(x) = 1\}, \\ 1, & \text{if } x \geq \min\{x \mid \mu_A(x) = 1\}. \end{cases}$
rather A	$\mathrm{int}(\mu_A^2)$		
more or less A	$\sqrt{\mu_A}$	Less than A	$\begin{cases} 1, & \text{if } x \leq \max\{x \mid \mu_A(x) = 1\}, \\ \mu_A(x), & \text{if } x > \max\{x \mid \mu_A(x) = 1\}. \end{cases}$



(a) Reference fuzzy sets *(b) Air-conditioning system*

Figure 1: Reference fuzzy sets (a) and the system considered in the application(b).

Application to a model of an air-conditioning system

We consider a model of an air-conditioning system consisting of a fan-coil unit. Hot or cold water is supplied to the coil through a valve. In the unit, outside (primary) air is mixed with return air from the room, see Fig. 1b. From systems measurements, a TS fuzzy model of has been obtained by clustering in ten clusters. The model predict the supply air temperature T_s based on its present and previous value, the mixed air temperature T_m, and the heating valve u, thus:

$$x(k) = [T_s(k), T_s(k-1), u(k-1), T_m(k)], \quad y(k) = T_s(k+1). \tag{11}$$

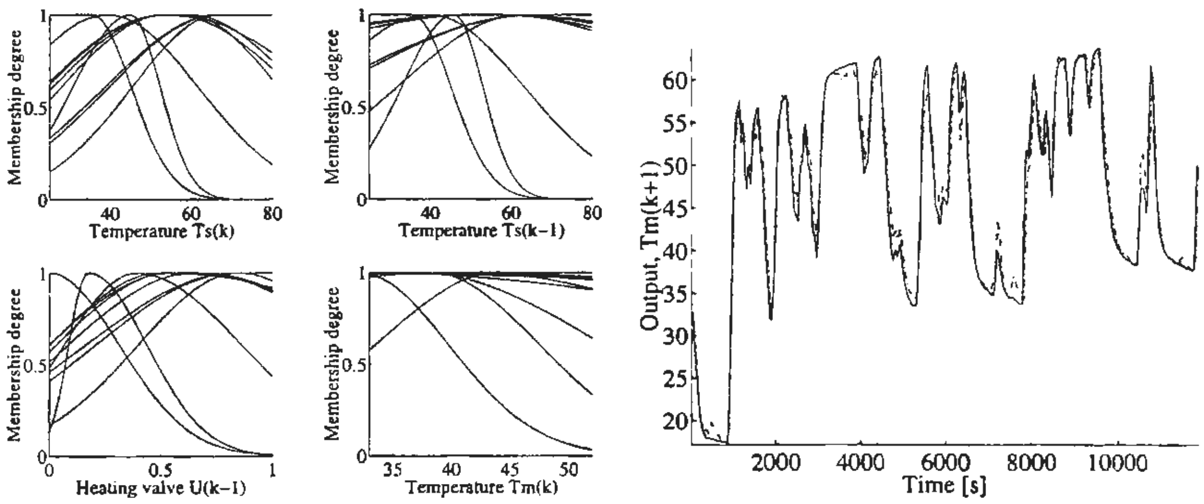
The model consist of ten rules, each with four antecedent fuzzy sets, of the form:

R_i : IF $T_s(k)$ is A_{i1} and $T_s(k-1)$ is A_{i2} and $u(k-1)$ is A_{i3} and $T_m(k-1)$ is A_{i4}**THEN** $T_s(k+1)$ is y_i

The total of 40 antecedent fuzzy sets used by the model are shown in Fig. 2. Applying the similarity relation approach of Table 1b with $\lambda = .8$ and $\gamma = .9$ gives an interesting result for the two antecedent variables $T_s(k)$ and $T_s(k-1)$. The partitioning of their domains are practically equal. This result is supported by knowledge about sampling time and systems dynamics. This suggests that one of the two variables could be removed from the model in further analysis. The simplified and reduced model consist of only 4 rules, given in Table 3, and 9 fuzzy sets, shown in Fig. 3. The new model is more transparent, and considerably faster than the original one. In a recursive simulation consisting of 397 predictions using unseen data, the original model uses 1.4 Mflops, while

Table 3: Simplified model.

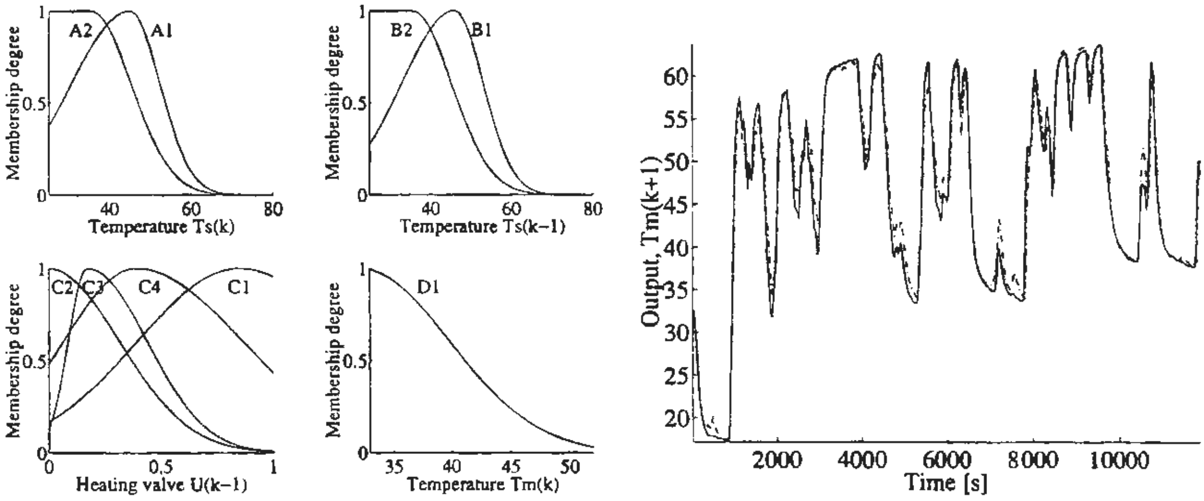
IF	$T_s(k)$ is,	$T_s(k-1)$ is,	$U(k-1)$ is,	$T_m(k)$ is,	THEN $T_s(k+1)$ is
	–	–	C_1	–	y_1
	A_1	B_1	C_2	–	y_2
	A_2	B_2	C_3	D_1	y_3
	–	–	C_4	–	y_4



(a) Fuzzy sets

(b) Recursive simulation

Figure 2: Original model: Fuzzy sets (a), recursive simulation (b)



(a) Fuzzy sets

(b) Recursive simulation

Figure 3: Original model: Fuzzy sets (a), recursive simulation (b)

the simplified model uses 0.4 Mflops. In this simulation, the root mean square (RMS) error of the original model is 1.89, while the RMS error of the new model is 1.91.

The application of linguistic approximation to the original model also gives a highly reduced rule base with 10 qualitative linguistic terms in 5 rules with the following antecedent parts:

If $T_s(k)$ is More than Low, $T_s(k-1)$ is More than Low, $u(k-1)$ is More than Low, $T_m(k)$ is Less than High
If $T_s(k)$ is More than Low, $T_s(k-1)$ is Less than High, $u(k-1)$ is More than Low, $T_m(k)$ is Less than High
If $T_s(k)$ is Less than High, $T_s(k-1)$ is Less than High, $u(k-1)$ is Less than High, $T_m(k)$ is More than Low
If $T_s(k)$ is Less than High, $T_s(k-1)$ is Less than High, $u(k-1)$ is Medium, $T_m(k)$ is Less than High
If $T_s(k)$ is Less than High, $T_s(k-1)$ is Less than High, $u(k-1)$ is More or less Med., $T_m(k)$ is Less than High

The linguistic description matches the simplified model in Fig. 3 quite well. For both variables $T_s(k)$ and $T_s(k-1)$, a partition into two fuzzy regions is found, and for variable $u(k-1)$, a partitioning in four fuzzy regions is found. For the input $T_m(k)$, both methods recognize the region 'Less than High', but the linguistic models also uses the region 'More than Low'. This region is removed from the simplified model due to its similarity with the universal set, and is thus implicit present with a membership 1 in the rules in Table 3 when D_1 is not used. The accuracy of the linguistic model is verified in a recursive simulation giving an RMS error of 2.17.

Conclusions

We have presented methods for complexity reduction in fuzzy models acquired from numerical data. The methods are based on similarity analysis of the fuzzy sets used in the antecedent space of the model. Distinction is made between iterative rule base reduction, reduction based on transitive similarity relations, and linguistic approximation.

The consequent parameters of the reduced fuzzy models can be re-estimated by least-squares techniques or recomputed from the parameters of the original model. The presented techniques have been applied to the fuzzy modeling of an real-world air-conditioning system. It is shown that the originally obtained model can be strongly reduced, allowing for qualitative interpretation, and faster computations, without deteriorating the prediction accuracy.

References

[1] R. Babuška, M. Setnes, U. Kaymak, and H.R. van Nauta Lemke. Rule base simplification with similarity measures. In *Proceedings FUZZ-IEEE'96*, pages 1642–1647, New Orleans, USA, 1996.

[2] R. Babuška and H.B. Verbruggen. Fuzzy set methods for local modeling and identification. In R. Murray-Smith and T. A. Johansen, editors, *Multiple Model Approaches to Nonlinear Modeling and Control*. Taylor & Francis, London, UK, 1996.

[3] J.C. Bezdek. *Pattern Recognition With Fuzzy Objective Function*. Plenum Press, New York, 1981.

[4] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.

[5] F. Esragh and E.H. Mamdani. A general approach to linguistic approximation. *Int. J. Man-Machine Studies*, 11:501–519, 1979.

[6] I. Gath and A. B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):773–781, July 1989.

[7] D.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE CDC*, pages 761–766, San Diego, USA, 1979.

[8] G.J. Klir and B. Youan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, New Jersey, 1995.

[9] B. Kosko. *Neural Networks and Fuzzy Systems: A dynamical systems approach to machine intelligence*. Prentice Hall, New Jersey, 1992.

[10] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of fuzzy systems*. John Wiley and Sons, Chichester, 1994.

[11] J.H. Nie and T.H. Lee. Rule-based modeling: Fast construction and optimal manipulation. *IEEE Transactions on Systems, Man and Cybernetics*, 26:728–738, 1996.

[12] M. Sugeno and G.T. Kang. Successive identificaiion of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems*, 42:315–334, 1991.

[13] R.M. Tong. The evaluation of fuzzy model derived from experimental data. *Fuzzy Sets and Systems*, 4:1–12, 1980.

ANALYTICAL SOLUTION APPROACHES OF NON UNIFORM TEMPERATURE PROFILES. APPLICATION TO HEAT EXCHANGER SIMULATIONS

J. Castillo[1] **and Ph. Bogaerts**[2]
Université Libre de Bruxelles
50 Av. F.-D. Roosevelt c.p.165-1050 Brussels (Belgium)
[1]Chemical engineering department (e-mail: jcastill@labauto.ulb.ac.be)
[2]Control engineering department (e-mail: pbogaert@labauto.ulb.ac.be)

Abstract. The analytical solution of the differential system modeling the enthalpy balance of a lumped heat exchanger has been developed for constant fluid flows and stepwise input temperatures. The interest of such a solution is illustrated with simulations aiming at comparing this solution with the true solution resulting from the resolution of the partial differential system describing the enthalpy balance of the heat exchanger.

1. Introduction

Together with other works [3], this study aims at proving the gains which can be obtained by solving analytically a partial differential problem, and this as far as possible before using approximate numerical methods. The example proposed consists of the mathematical model of a convective heat transfer between two fluids in flow. The non uniformity of the fluid temperatures leads to a distributed parameter problem which has to be solved analytically [1] and/or numerically [6].

Using the well-known approximated method of segmentation into finite volumes [2, 4, 5, 7] at uniform temperature, that system can be reduced to a system of 2N ordinary differential equations (N being the number of finite volumes).

The exchanger considered here is an industrial co-current flow apparatus. In situ measurements show that the temperature of the cold fluid can be assumed uniform. Nevertheless, due to the symmetry of the model, an analytical solution of the system of 2N ODE's resulting from the segmentation is proposed in the general case of two non uniform fluids. It will be shown that the simulated temperatures in the different cold volumes are almost identical, in agreement with the hypothesis of uniformity. This analytical solution has been developped for constant flows and for stepwise variations of the input temperature. The proposed method allows the estimation of the temperatures at given times without any numerical integration. Moreover it has the advantage of predicting the steady-state temperatures for both fluids thanks to a simple matrix product. The temperatures estimated by this method will be compared to the "true" values given by the analytical solution [1] of the distributed parameter model by doing the hypothesis of the cold flow uniformity.

The second section briefly presents the process and its mathematical models. The analytical solution of the system of 2N ODE's is given in the third section. The fourth section is devoted to the solution of the distributed parameter model by doing the hypothesis of the cold flow uniformity. The simulated temperatures obtained by both methods are presented in the fifth section. Conclusions are discussed in section six. The resolution of the differential system of 2N ODE's is presented in appendix1.The notations and numerical values for the simulations are given in appendix 2.

2. Description and mathematical models of the process

We consider here a co-current flow heat exchanger involving two liquid fluids. The flows and the input temperature of the cooling fluid are assumed time constant. The hot fluid comes out a previous exothermic reactor and must be cooled down to a given temperature. The mathematical models, which will be used in the simulations of the next sections, are given below. The significance of the symbols is given in appendix 2.

Energy balance for the i^{th} volumes

$$\begin{cases} \rho_f c_{p_f} V_f^i \dfrac{dT_f^i(t)}{dt} = US_{exch}^i\left(T_e^i(t) - T_f^i(t)\right) + \rho_f c_{p_f} q_f\left(T_f^{i-1}(t) - T_f^i(t)\right) \\ \rho_e c_{p_e} V_e^i \dfrac{dT_e^i(t)}{dt} = US_{exch}^i\left(T_f^i(t) - T_e^i(t)\right) + \rho_e c_{p_e} q_e\left(T_e^{i-1}(t) - T_e^i(t)\right) \end{cases} \qquad (1)$$

Substituting $V_f^i = \dfrac{V_f}{N}$ and $S_{exch}^i = \dfrac{S_{exch}}{N}$ in (1),

$$\begin{cases} \rho_f c_{p_f} V_f \dfrac{dT_f^{\,i}(t)}{dt} = US_{exch}\left(T_e^{\,i}(t) - T_f^{\,i}(t)\right) + \rho_f c_{p_f} q_f L \dfrac{T_f^{\,i-1}(t) - T_f^{\,i}(t)}{L/N} \\ \rho_e c_{p_e} V_e \dfrac{dT_e^{\,i}(t)}{dt} = US_{exch}\left(T_f^{\,i}(t) - T_e^{\,i}(t)\right) + \rho_e c_{p_e} q_e L \dfrac{T_e^{\,i-1}(t) - T_e^{\,i}(t)}{L/N} \end{cases} \quad \text{where } i = 1 \dots N \tag{2}$$

Note that $T_e^0 = T_e^{IN}$ and $T_f^0 = T_f^{IN}$.

For an infinite segmentation of the exchanger, the energy balances are modeled by the following partial differential system:

$$\begin{cases} \rho_f c_{p_f} V_f \dfrac{\partial T_f(x,t)}{\partial t} = US_{exch}\left(T_e(x,t) - T_f(x,t)\right) - \rho_f c_{p_f} q_f L \dfrac{\partial T_f(x,t)}{\partial x} \\ \rho_e c_{p_e} V_e \dfrac{\partial T_e(x,t)}{\partial t} = US_{exch}\left(T_f(x,t) - T_e(x,t)\right) - \rho_e c_{p_e} q_e L \dfrac{\partial T_e(x,t)}{\partial x} \end{cases} \tag{3}$$

Assuming that the temperature of the cold flow is uniform, it can be shown that the system (3) can be reduced to:

$$\begin{cases} \rho_f c_{p_f} V_f \dfrac{\partial T_f(x,t)}{\partial t} = US_{exch}\left(T_e(t) - T_f(x,t)\right) - \rho_f c_{p_f} q_f L \dfrac{\partial T_f(x,t)}{\partial x} \\ \rho_e c_{p_e} V_e \dfrac{dT_e(t)}{dt} = \dfrac{US_{exch}}{L} \displaystyle\int_0^L \left(T_f(x,t) - T_e(t)\right)dx + \rho_e c_{p_e} q_e \left(T_e^{IN} - T_e(t)\right) \end{cases} \tag{4}$$

3. Analytical solution of the finite volume model

The system (2) of 2N ODE's describing the energy balances of the 2N finite volumes can be written in matrix form as follows:

$$\frac{dT}{dt} = AT + B \tag{5}$$

$$= \begin{bmatrix} -a_f & 0 & \cdots & & 0 & \alpha_f & 0 & \cdots & & 0 \\ \beta_f & -a_f & 0 & & \vdots & 0 & \alpha_f & & & \vdots \\ 0 & \ddots & \ddots & & \vdots & \vdots & & \ddots & & \\ \vdots & & \ddots & \ddots & 0 & & & & \ddots & \\ 0 & \cdots & 0 & \beta_f & -a_f & 0 & \cdots & & & \alpha_f \\ \alpha_e & 0 & \cdots & & 0 & -a_e & 0 & \cdots & & 0 \\ 0 & \alpha_e & & & \vdots & \beta_e & -a_e & 0 & & \vdots \\ \vdots & & \ddots & & & 0 & \ddots & \ddots & & \vdots \\ & & & \ddots & & \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & & & \alpha_e & 0 & \cdots & 0 & \beta_e & -a_e \end{bmatrix} T + \begin{bmatrix} \beta_f T_f^{IN} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \beta_e T_e^{IN} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

Where T is the column vector of the temperatures $T_f^{\,i}$ and $T_e^{\,i}$: $T = {}^t[T_f^{\,1} T_f^{\,2} \dots T_f^{\,N} T_e^{\,1} T_e^{\,2} \dots T_e^{\,N}]$, $a_f = \alpha_f + \beta_f$ and $a_e = \alpha_e + \beta_e$ $\left(\alpha_f = \dfrac{US_{exch}}{\rho_f c_{p_f} V_f},\ \alpha_e = \dfrac{US_{exch}}{\rho_e c_{p_e} V_e},\ \beta_f = \dfrac{Nq_f}{V_f},\ \beta_e = \dfrac{Nq_e}{V_e}\right)$.

The resolution is presented in appendix 1. The considered time dependence of the input hot fluid temperature is the following:

$$T_f^{IN} = T_f^{IN1} \upsilon(-t) + T_f^{IN2} \upsilon(t) \tag{6}$$

4. Analytical solution of the partial differential system

The general solution of the partial differential equation (4) is given by [1]:

$$\begin{aligned} T_f(x,t)\upsilon(x)\upsilon(t) &= e^{-\frac{x}{\alpha}} T_f(0,t-\frac{\beta}{\alpha}x)\upsilon(x)\upsilon(t-\frac{\beta}{\alpha}x) \\ &+ e^{-\frac{t}{\beta}} T_f(x-\frac{\alpha}{\beta}t,0)\upsilon(t)\upsilon(x-\frac{\alpha}{\beta}t) \\ &+ \frac{e^{-\frac{t}{\beta}}}{\beta}\int_{t-\frac{\beta}{\alpha}x}^{t} e^{\frac{\tau}{\beta}} T_e(\tau)\upsilon(\tau)\,d\tau \end{aligned} \tag{7}$$

$$\text{where} \begin{cases} \alpha = \dfrac{q_f L}{\alpha_f V_f},\ \beta = \dfrac{1}{\alpha_f} \\ T_f(0,t) = T_f^{IN1}\upsilon(-t) + T_f^{IN2}\upsilon(t) \\ T_f(x,0) = \left(T_f^{IN1} - T_e(0)\right)e^{-\frac{x}{\alpha}} + T_e(0) \\ T_e(0) = \dfrac{(q_e/V_e)T_e^{IN} + ((\alpha\alpha_e)/L)\,(1-e^{-L/\alpha})T_f^{IN1}}{(q_e/V_e) + ((\alpha\alpha_e)/L)\,(1-e^{-L/\alpha})} \end{cases}$$

It is assumed that the initial temperatures are in steady state.

The simulation of $T_e(t)$ needs the computing of the integral $\int_0^L T_f(x,t)\,dx$. It can be shown that its value is:

$$\begin{aligned} \int_0^L T_f(x,t)\,dx &= \alpha T_f^{IN2}\left(1-e^{-\frac{1}{\alpha}MIN(L,\frac{\alpha}{\beta}t)}\right) \\ &+ e^{-\frac{t}{\beta}}\left(T_e(0)\left(L - MIN(L,\frac{\alpha}{\beta}t)\right) + \alpha\left(T_f^{IN1} - T_e(0)\right)\left(1-e^{-\frac{1}{\alpha}(L-MIN(L,\frac{\alpha}{\beta}t))}\right)\right) \\ &+ \frac{e^{-\frac{t}{\beta}}}{\beta}\Big(LF(t) + \frac{\alpha}{\beta}\Big(G(t-\frac{\beta}{\alpha}L)\,\upsilon(t-\frac{\beta}{\alpha}L) - G(t)\Big) \end{aligned} \tag{8}$$

$$\text{where} \begin{cases} F(t) = \displaystyle\int_0^t e^{\frac{\tau}{\beta}} T_e(\tau)\upsilon(\tau)\,d\tau \\ G(t) = \displaystyle\int_0^t F(\tau)\upsilon(\tau)\,d\tau \end{cases}$$

5. Examples of simulations

The different results in simulation are given in figures 1 to 6. It can be seen that the temperature of the cold fluid is correctly estimated by the segmentation method, even with only three finite volumes. However, the estimation of the hot fluid temperature is not so satisfactory, as it can be seen in figures 4 and 5. Figure 5 shows that there is a discontuity in the output temperature of the hot fluid. It is due to the plug character of the flow.

It is shown in figures 3 and 6 that the asymptotic temperatures very quicly converge to the "true" value when the number of finite volumes increases (with 15 finite volumes the error of estimation of the hot fluid temperature is smaller than half a degree).

It can be seen in figures 7 and 8 that the steady state temperatures in the different volumes are quasi identical for the cold fluid but not for the hot fluid, which means that the hypothesis of uniformity of the cold fluid's temperature is licit; this latter simulation is done in the the case of three finite volumes.

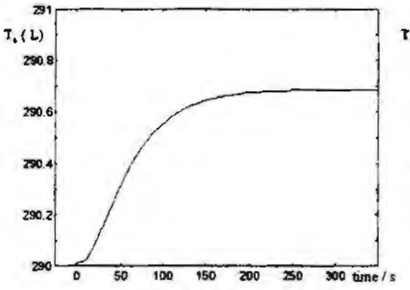


Fig. 1 Output temperature of the cold fluid obtained by using the segmentation method with three finite volumes

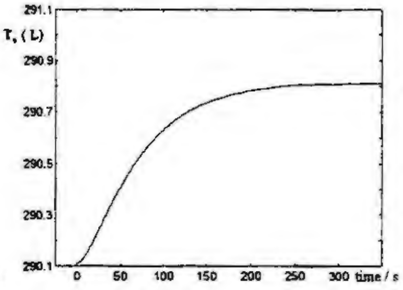


Fig. 2 Output temperature of the cold fluid obtained from the the analytical solution

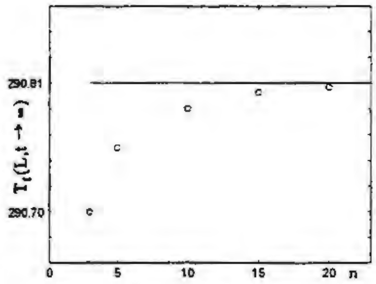


Fig. 3 Output steady-state temperature of the cold fluid as a function of the number of finite volumes in comparison with the analytical solution

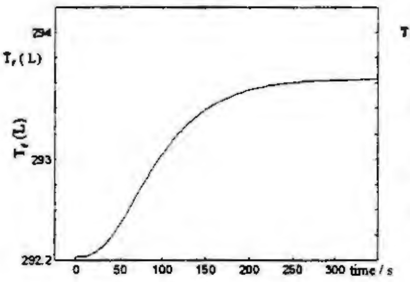


Fig. 4 Output temperature of the hot fluid obtained by using the segmentation method with three finite volumes

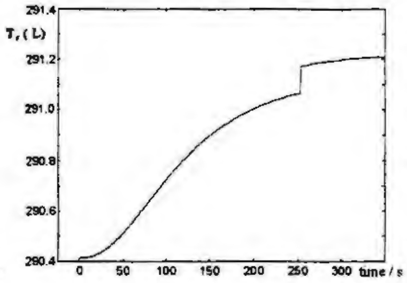


Fig. 5 Output temperature of the hot fluid obtained from the the analytical solution

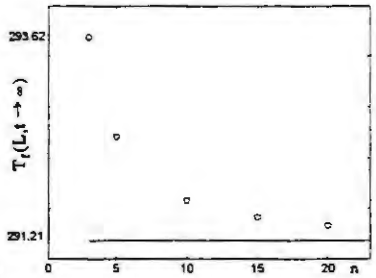


Fig. 6 Output steady-state temperature of the hot fluid as a function of the number of finite volumes in comparison with the analytical solution

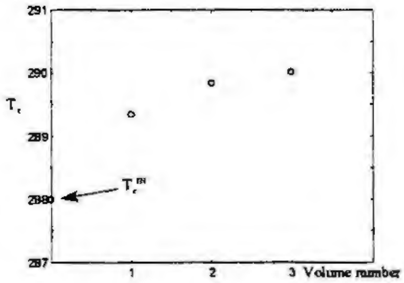


Fig. 7 Initial steady state temperature of the cold flow in each volume

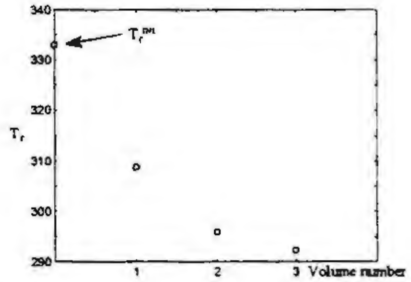


Fig. 8 Initial steady state temperature of the hot flow in each volume

6. Conclusions

The main result of this study consists of the analytical solution of the system of 2N ODE's describing the enthalpy balance of a lumped heat exchanger. The interest of this analytical solution (of an approximate model) consists of the possibility to determine both fluid temperatures without any numerical integration. The quality of the approximate solution is determined by comparison of its results with the "true" solution of the partial differential problem that we have previously published [1]. Moreover, the solution proposed allows to determine the asymptotic temperatures of both fluids thanks to a simple matrix product.

Appendix 1: resolution of the differential system

The general solution of the linear differential system (5) is given by the sum of two terms: the general solution of the homogeneous system ($\frac{dT}{dt} = AT$) and a particular solution of the non-homogeneous system.

The eigenvalues of the matrix A are solution of the equation $\det(A - \lambda I) = 0$. Let H=A-λI. The matrix H can be seen as composed of four blocks whose dimensions are N×N: $H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$.

It can be shown that $\det(H) = \det(H_{22})\det(H_{11} - H_{12}H_{22}^{-1}H_{21})$ [8]. Thus the determinant of H is given by the following expression: $\det(H) = \left((a_e+\lambda)(a_f+\lambda) - \alpha_e\alpha_f\right\}^N$ and there are two different eigenvalues of algebraic multiplicity N, both solution of the equation: $(a_e+\lambda)(a_f+\lambda) - \alpha_e\alpha_f = 0$. The eigenvalues are noted λ_1 and λ_2.

There is one set of N generalised eigenvectors associated to each eigenvalue: $EV_1{}^i$ and $EV_2{}^i$ (i=1...N). They are found by solving the following systems of algebraic equations:

$$\begin{cases} (A-\lambda_1 I)\, EV_1{}^1 = 0 \\ (A-\lambda_1 I)\, EV_1{}^i = EV_1{}^{(i-1)} \end{cases} \quad \text{and} \quad \begin{cases} (A-\lambda_2 I)\, EV_2{}^1 = 0 \\ (A-\lambda_2 I)\, EV_2{}^i = EV_2{}^{(i-1)} \end{cases} \quad (i=2...N)$$

The non-zero components of the eigenvectors are given by the following recurrent expressions:

$$\begin{cases} EV_1{}^i(2N-(i-k)) = \dfrac{(a_e+\lambda_1)\beta_f}{\alpha_e((a_f+\lambda_1)\beta_e+(a_e+\lambda_1)\beta_f)}\left(EV_1{}^{i-1}(N-(i-k)) - \beta_f EV_1{}^i(N-(i-(k-1)))\right) \\ \qquad + \dfrac{(a_f+\lambda_1)}{((a_f+\lambda_1)\beta_e+(a_e+\lambda_1)\beta_f)}\left(EV_1{}^{i-1}(2N-(i-(k+1))) + \dfrac{(a_e+\lambda_1)}{\alpha_f} EV_1{}^{i-1}(N-(i-(k+1)))\right) \\ EV_1{}^i(N-(i-k)) = -\dfrac{\alpha_f\beta_e}{(a_e-\lambda_1)\beta_f} EV_1{}^i(2N-(i-k)) + \dfrac{\alpha_f}{(a_e+\lambda_1)\beta_f} EV_1{}^{i-1}(2N-(i-(k+1))) \\ \qquad + \dfrac{1}{\beta_f} EV_1{}^{i-1}(N-(i-(k+1))) \end{cases} \tag{9}$$

(i = 1 ... N and k = 1 ... i-1)

$$\begin{cases} EV_1{}^i(2N) = 1 \\ EV_1{}^i(N) = \dfrac{\alpha_f}{a_f+\lambda_1} + \dfrac{\beta_f}{a_f+\lambda_1} EV_1{}^i(N-1) - \dfrac{I}{a_f+\lambda_1} EV_1{}^{i-1}(N) \end{cases}$$

The same expressions can be written for $EV_2{}^i$.

The general solution of the homogeneous system is:

$$\begin{aligned} T_H = & \left(c_1 EV_1{}^1 + c_2(EV_1{}^2 + t\,EV_1{}^1) + \ldots + c_N(EV_1{}^N + t\,EV_1{}^{N-1} + \ldots \frac{t^{N-1}}{N-1} EV_1{}^1) \right) e^{\lambda_1 t} \\ & + \left(d_1 EV_2{}^1 + d_2(EV_2{}^2 + t\,EV_2{}^1) + \ldots + d_N(EV_2{}^N + t\,EV_2{}^{N-1} + \ldots \frac{t^{N-1}}{N-1} EV_2{}^1) \right) e^{\lambda_2 t} \end{aligned} \tag{10}$$

Where c_i and d_i (i=1 ... N) are arbitrary constants to be deduced from the initial conditions.

A particular solution of the non homogeneous system is given, for a stepwise disturbance, by

$$T_P = -A^{-1}B \tag{11}$$

The general solution of the system of 2N ODE's is thus given by

$$T = T_P + T_H \tag{12}$$

Appendix 2: nomenclature and numerical values

c_{p_e}	specific heat of the cooling water $(4184\ J\ kg^{-1}\ K^{-1})$
c_{p_r}	specific heat of the reacting mixture $(2092\ J\ kg^{-1}\ K^{-1})$
L	length of the exchanger $(2.5\ m)$
N	number of finite volumes
q_e	flow of the cooling water $(0.0111\ m^3\ s^{-1})$
q_f	flow of the reacting mixture $(0.0011\ m^3\ s^{-1})$
S_{exch}	heat exchanger area $(40\ m^2)$
t	time (s)
T_e	temperature of the cooling water (K)
T_e^i	temperature of the cooling water in the i-th finite volume (K)
T_e^{IN}	temperature of the cooling water at the input of the exchanger $(288\ K)$
T_f	temperature of the reacting mixture (K)
T_f^i	temperature of the reacting mixture in the i-th finite volume (K)
T_f^{IN}	temperature of the reacting mixture at the input of the exchanger (K)
U	global coefficient of convective heat exchange $(285\ J\ s^{-1}\ m^{-2}\ K^{-1})$
V_e	volume of the cooling water $(0.2121\ m^3)$
V_f	volume of the reacting mixture $(0.2788\ m^3)$
x	position (m)
ρ_e	specific mass of the cooling water $(1000\ kg\ m^{-3})$
ρ_f	specific mass of the reacting mixture $(1000\ kg\ m^{-3})$

References

1. Bogaerts, Ph., Castillo, J. and Hanus, R., Analytical solution of the non uniform heat exchange in a reactor cooling coil with constant fluid flow. Accepted for publication in: Mathematics and Computers in Simulation.

2. Cabassud, M., Le Lann, M.-V., Ettedgui, B. and Casamatta, G., A general simulation model of batch chemical reactors for thermal control investigations. Chem. Eng. Technol., 17 (1994), 255-268.

3. Bogaerts, Ph., Castillo, J. and Hanus, R., Analytical solution of the non uniform heat exchange in a reactor cooling coil with time varying fluid flow. Accepted for publication in: Proc. of the IMACS 2nd Mathmod, Vienna, Elsevier, 1997.

4. Juba, M. R. and Hamer, J. W., Progress and challenges in batch process control. In: Proc. of the Third International Conference on Chemical Process Control, (Eds.: Morari, M. and McAvoy, T.J.) Cache Elsevier, 1986, 139-183.

5. Luyben, W. L., Process modeling, simulation and control for chemical engineers. McGraw-Hill, Chemical Engineering Series, 1990.

6. Maffezzoni, C. and Ferrarini, L., A characteristic ligne based method to build finite-dimensional models of heat exchangers. Mathematical Modelling of Systems, Vol. 1 no. 3 (1995), 141-166.

7. Szeifert, F., Chovan, T. and Nagy, L., Process dynamics and temperature control of fed-batch reactors. Computers chem. Engng., Vol. 19, Suppl., (1995), S447-S452

8. Strank, G., Linear Algebra and its applications. Academic Press.

COMBINATION OF TWO APPROACHES TO MODELLING OF PRESSURE - LEVEL PILOT PLANT

Srečko Milanič, Maja Atanasijevič - Kunc, Rihard Karba and Borut Zupančič
Faculty of Electrical Engineering, University of Ljubljana
Tržaška 25, 1000 Ljubljana, Slovenia
e-mail: srecko.milanic@fe.uni-lj.si

Abstract

In the paper presented, different models of a pilot pressure-level pilot plant are developed: a theoretical nonlinear, a linearised and an artificial neural network model. A Gaussian network together with the "hybrid learning rule" was used to obtain the neural model. Simulation results have demonstrated acceptable matching with measurement data for all the models. Validation was also done in the closed loop. At the end of the paper some thoughts are given about our future work which is to result in a hybrid model of the pilot plant.

Introduction

Modelling is nowadays commonly used in most fields of science and technology. Further more, modelling is inseparably intertwined with control design. In some approaches (e.g. model based predictive control) model design represents the majority of time and effort spent for control design. In some cases of application, a single model is sufficient, but it is often very useful to develop several models of different forms since different model forms posses different desired properties. In this paper, a theoretical model of a pressure - level pilot plant is given. An artificial neural network model was also identified using measurements form the real plant and theoretical model data. Each of these two approaches has its advantages and drawbacks depending on the application as the importance of the modelling procedure is multipurpose.

Short description of the pilot plant

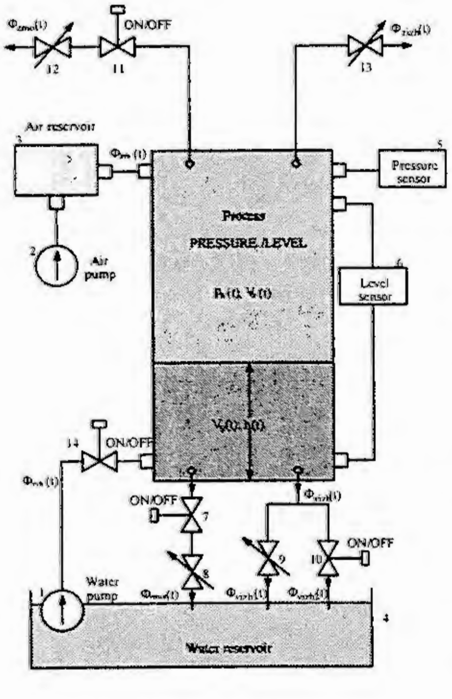


Figure 1: Schematic representation of the plant

Schematic representation of the pressure - level pilot plant is shown in fig. 1. It has been built for control studying purposes as this type of variables can frequently be met in process industry. The central part of the device is a closed tank where air-pressure and water - level can be controlled through two pumps. They represent the actuators of the system with input voltage range 0 - 10 V. Both outputs of the sensors are also inside the same voltage range. If either of process variables exceed the prescribed operating range the protection device switches off the adequate actuator. Normal working conditions of the system can be disturbed when changing set - points of corresponding valves, but there is no possibility of direct measurement of this signals. Different types of working conditions can be distinguished. In the case when control properties are posed on both process variables simultaneously the system is of multivariable nature due to the fact that in this case the air pressure and water level are mutually dependent. For the process nonlinear mathematical model was built [1] in such a manner that also start up and limited conditions were taken into account. It was further proved, that around the chosen working point linearised approximations can be accepted, but they are not at all reliable descriptions in the whole operating range.

Theoretical Modelling of the Pilot Plant

The device can be modelled as a cascade of three subsequent blocs: the activator block, the main process block and the sensor block. Each of these blocks has two inputs and two outputs. For the description of the actuator block, the following nonlinear equations were used:

$$\begin{aligned} T_{a11}T_{a12}\frac{d^2[\Phi_{zvh}(t)]}{dt^2}+[T_{a11}+T_{a12}]\frac{d[\Phi_{zvh}(t)]}{dt}+\Phi_{zvh}(t) &= K_{a1}u_1^2(t) \\ T_{a211}\frac{d[\Phi_{vvh}(t)]}{dt}+\Phi_{vvh}(t) &= K_{a2}u_2^5(t) \end{aligned} \quad (1)$$

u_1 and u_2 represent actuators input voltages while ϕ_{zvh} and ϕ_{vvh} are the resulting air and water flows. The inputs into the main process are therefore air and water flows. The outputs are air pressure p_z and water level h. Modelling of the main process starts with the following mass equilibrium equations:

$$\Phi_{zvh}(t)-\Phi_{zizh}(t) = \frac{d[m_z(t)]}{dt}; \quad \Phi_{vvh}(t)-\Phi_{vizh}(t) = \frac{d[m_v(t)]}{dt} \quad (2)$$

It has been assumed that the flow through valves is a square function of the pressure difference on the valve inlet, that water compression is negligible and that air in the closed tank can be treated as ideal gas. These assumptions result in the following equations:

$$\Phi_{zizh}(t) = K_{zv}\sqrt{p_z(t)-p_0}; \quad \Phi_{vizh}(t) = \sqrt{K_1h(t)+K_2(p_z(t)-p_0)} \quad (3)$$

where p_0 is normal air pressure and K_{zv}, K_1, K_2 are valve constants. The relation between air mass in the tank, its pressure p_z, its volume V_z and temperature T can be expressed using the generic gas equation:

$$m_z(t) = \frac{p_z(t)V_z(t)}{r_zT} = \frac{p_z(t)}{r_zT}S[H_0-h(t)] \quad (4)$$

where specific gas constant r_z can be evaluated from known conditions of the air. The nonlinear model of the main process is obtained in the form of two differential equations:

$$\begin{aligned} \frac{d[p_z(t)]}{dt} &= X_1\frac{K_B}{H_0-h(t)}\Phi_{zvh}(t)-X_2\frac{K_BK_{zv}}{H_0-h(t)}\sqrt{p_z(t)-p_0}+X_3\frac{1}{H_0-h(t)}p_z(t)\frac{d[h(t)]}{dt} \\ \frac{d[h(t)]}{dt} &= \frac{1}{C_2}\Phi_{vvh}(t)-\frac{1}{C_2}\sqrt{K_1h(t)+K_2(p_z(t)-p_0)} \end{aligned} \quad (5)$$

where $C_2=\rho_vS$, S representing the cross area of the tank, ρ_v is specific water weight and H_0 is the tank height. The fraction expression $\frac{r_zT}{S}$ is replaced by the constant K_B.

The constants X1=X2=X3, $X_i < 1$, i=1,2,3 have been introduced into equation (5) due to numerical reasons. With X_i=1, the model is very stiff and simulation results can go unstable. This system property can become especially problematic when the model is used in the closed loop with the controller. The constants X_i slow down the dynamics of equation (5) and therefore reduce the model stiffness.

Both sensors were assumed to be accurately described by linear characteristics:

$$y_1(t) = K_{s1}(p_z(t)-p_0)-\bar{y}_{1off}; \quad y_2(t) = K_{s2}h-\bar{y}_{2off} \quad (6)$$

Linearised approximations of model descriptions are often used for controller synthesis purposes. Using Taylor's series the nonlinear description of the main process can be transformed into the following transfer function matrix:

$$\underline{\underline{G}}(s) = \frac{1}{\frac{K_1}{K_2}(s\tau_{11}+1)(s\tau_{21}+1)+K_As}\begin{bmatrix} \frac{K_1}{K_2}(s\tau_{21}+1) & \frac{K_1}{K_2}R_{21}K_As \\ -R_1 & \frac{K_1}{K_2}R_{21}(s\tau_{11}+1) \end{bmatrix} \quad (7)$$

which can be further simplified into the following form:

$$\underline{\underline{G}}(s) = \begin{bmatrix} \frac{R_1}{(s\tau_{11}+1)} & \frac{R_{21}K_As}{(s\tau_{11}+1)(s\tau_{21}+1)} \\ \frac{-\frac{K_2R_1}{K_1}}{(s\tau_{11}+1)(s\tau_{21}+1)} & \frac{R_{21}}{(s\tau_{21}+1)} \end{bmatrix} \quad (8)$$

under the assumption that the expression (K_As) can be neglected in the common denominator of (7).

For the pumps the following transfer functions can be derived:

$$g_{a1}(s) = \frac{K_{a1l}}{(sT_{a11}+1)(sT_{a21}+1)}; \qquad g_{a2}(s) = \frac{K_{a2l}}{(sT_{a211}+1)}. \tag{9}$$

The parameters of the linearised approximation of course depend on nonlinear model and working conditions whereas the parameters of the nonlinear model were obtained on the basis of theoretical modelling approach and known system dimensions and properties. For the case presented, these constants were used: H_0=0.3m, S=24·10^{-4}m^2, p_0=76·10^3N/m^2, K_B=3.205·10^7/s^2, C_2=2.3958kg/m.

Since we didn't have the possibility to measure the input flows, the sensor transfer functions (K_{s1}=0.2V/mbar, K_{s2}=0.4V/cm) and time constants of actuators (T_{a11}=1s, T_{a12}=0.8s, T_{a211}=1.5s) were accepted as suggested in [8]. The input flows were estimated as $\overline{\Phi}_{zvh}$ = 22·10^{-6}kg/s and $\overline{\Phi}_{vvh}$=7·10^{-3}kg/s. For constants X_i the value 0.01 was found to be suitable.

Artificial Neural Network Models

The main reason why we decided to build a neural model was the stiffness of the theoretical model we already disposed of. Discrete artificial neural network models use a different numerical representation form the theoretical model. Therefore we decided to train the network using the theoretical nonlinear model (5) whose stiffness has been reduced. This prevented the neural model from inheriting the stiffness property. In the next step, the neural model parameters were further adapted to mimic as closely as possible the real plant behaviour in closed loop.

In order to obtain neural models of the pilot plant, we used a feed-forward sigmoidal multi-layer network and a Gaussian network together with the adherent training rules. *"The Random Activation Weight Network"* (RAWN) as recently presented by [2] is a one hidden layer sigmoidal network whose weights are computed by a non-iterative procedure. It is orders of magnitude faster than the standard back-propagation and usually yields excellent approximation results for MISO modelling [3]. However, for the case being presented, Gaussian nets, described by (10, 11):

$$y_j = \sum_{i=1}^{N} w_{ij} z_i \tag{10}$$

$$z_i = \exp\left[-\frac{1}{2}\left(\frac{(x_1-c_{i1})^2}{\sigma_{i1}^2}+\frac{(x_2-c_{i2})^2}{\sigma_{i2}^2}+\ldots+\frac{(x_n-c_{in})^2}{\sigma_{in}^2}\right)\right] \tag{11}$$

were found to yield superior performance. The sigmoidal network trained with the RAWN algorithm exhibited stability problems. The reasons are being investigated. Series parallel model as described in [5] was used for training sigmoidal and Gaussian networks. It guarantees network stability during training phase. Prediction of the process output was performed recursively (12):

$$\begin{aligned}\hat{y}_j(k+1) = f_{ANN}(&\hat{\underline{y}}(k), \hat{\underline{y}}(k-1),\ldots\\ &\ldots,\hat{\underline{y}}(k-l), \underline{u}(k), \underline{u}(k-1),\ldots,\underline{u}(k-m)).\end{aligned} \tag{12}$$

f_{ANN} denotes ANN approximation of a non-linear mapping, $\hat{y}_j(k+1)$ denotes model prediction of the j-th plant output signal, $\underline{u}(k-m)$ is the network input vector in time instant *k-m* and $\hat{\underline{y}}(k-l)$ denotes the network estimated output vector in time instant *k-l*. f_{ANN} depends on the network's parameters which have to be optimised during the training phase in order to minimise discrepancy between $\hat{\underline{y}}(k+1)$ and $\underline{y}(k+1)$.

The Gaussian network model consisted of 26 hidden nodes. The parameters c_{in}, s_{in}, w_i were determined by the *"hybrid algorithm"* [4]. After some trials with the nonlinear model of the plant, an identification signal was selected which excites the plant in a wide operating range. The response of the nonlinear model was simulated and used to train the neural model. Therefore, information about the plant was transferred from the nonlinear model to the neural model in form of training data points.

The models have been validated using the input signals presented in fig. 2, 3. The corresponding responses of the real plant, nonlinear model and neural model can be seen in fig. 4-7. To provide information more precise than visual, the differences between the plant responses and model responses were evaluated using root mean square error (RMSE) and Thiel's Inequality Coefficient (TIC) [6]. The results are summarised in tables 1-4.

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N}(y(k)-\hat{y}(k))^2}{N}} \qquad TIC = \frac{\sqrt{\sum_{i=1}^{n}(y_i - z_i)^2}}{\sqrt{\sum_{i=1}^{n} y_i^2} + \sqrt{\sum_{i=1}^{n} z_i^2}} \tag{13}$$

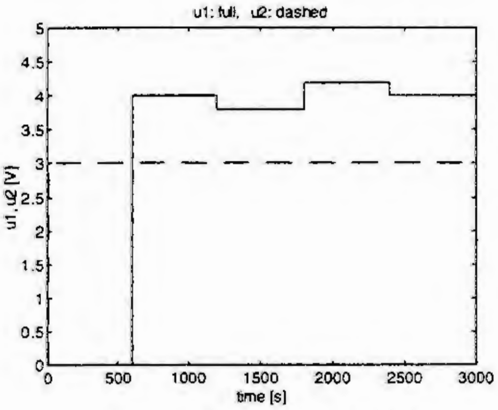


Figure 2: Input voltage signals 1

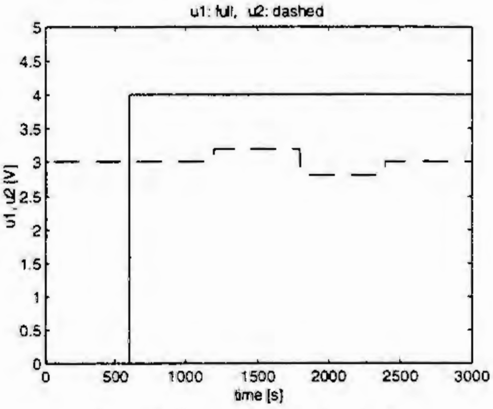


Figure 3: Input voltage signals 2

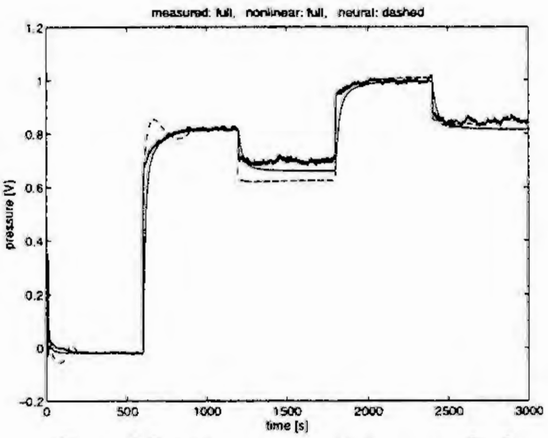


Figure 4: Pressure responses to inputs on fig. 2

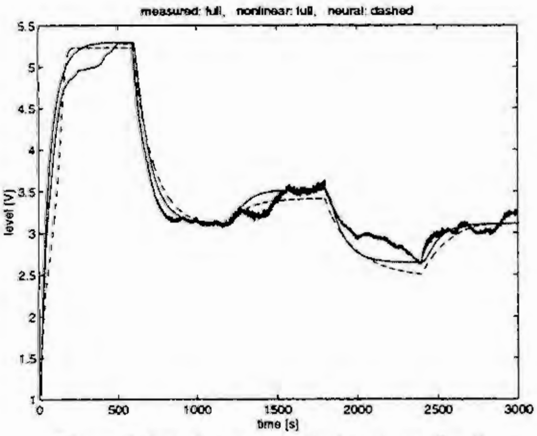


Figure 5: Level responses to inputs on fig. 2

	RMSE	TIC
nonlinear	0.0426	0.0287
neural	0.0425	0.0285

Table 1: Discrepancies of model responses on fig. 4

	RMSE	TIC
nonlinear	0.1663	0.0231
neural	0.2169	0.0305

Table 2: Discrepancies of model responses on fig. 5

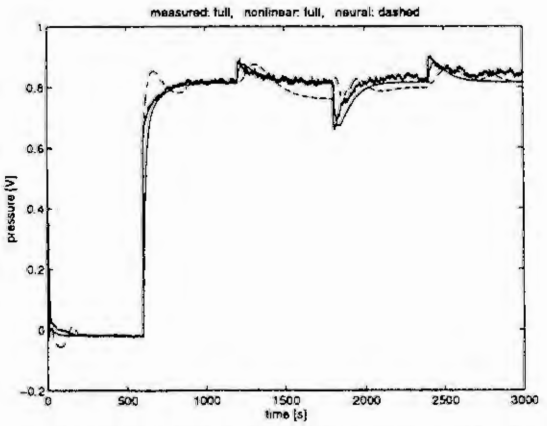


Figure 6: Pressure responses to inputs on fig. 3

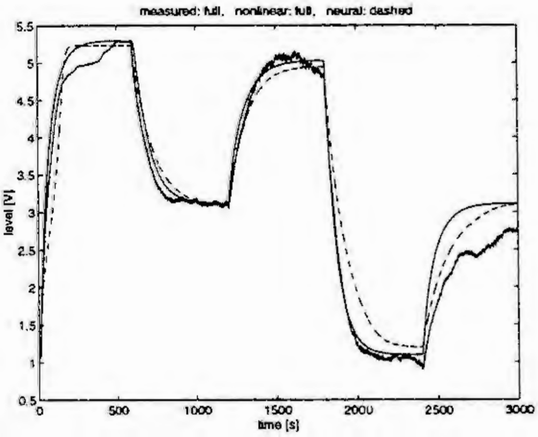


Figure 7: Level responses to inputs on fig. 3

	RMSE	TIC
nonlinear	0.0370	0.0254
neural	0.0378	0.0259

Table 3: Discrepancies of model responses on fig. 6

	RMSE	TIC
nonlinear	0.2963	0.0402
neural	0.3203	0.0439

Table 4: Discrepancies of model responses on fig. 7

It can be seen that both nonlinear and neural model yield comparable results. However, examining RMSE and TIC values for the signals, the values adherent to nonlinear model are slightly smaller except for the signals on fig. 4. It is important to notice that higher/lower RMSE values correspond to higher/lower TIC values respectively.

Closed Loop Validation

One of the most indicative tests when validating a model is to simulate the behaviour of the closed loop system. The simulated system's behaviour which contains the model and a regulator is compared to the real plant closed loop behaviour. Of course, the model output signals do often not differ much from the real plant outputs because of the regulator's action. In such a case, it is important to observe the control signals. If the control signals with which the regulator forces the model to follow the reference and the control signals from the real system do not differ much, then the obtained model is a very faithful prototype of the plant dynamics.

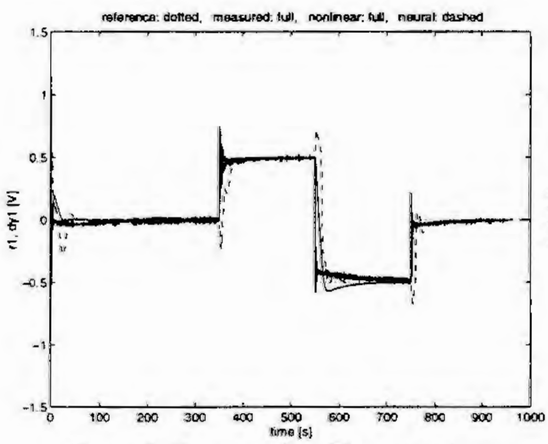


Figure 8: Pressure closed loop responses

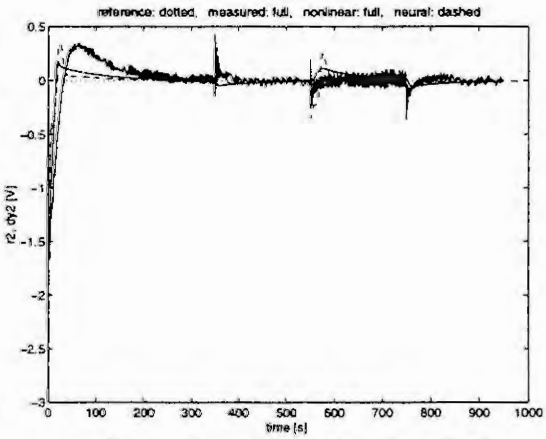


Figure 9: Level closed loop responses

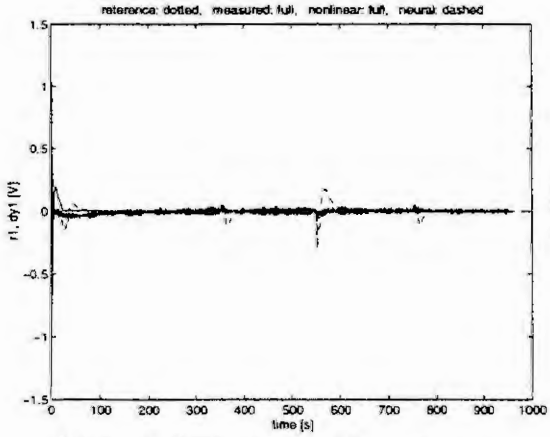


Figure 10: Pressure closed loop responses

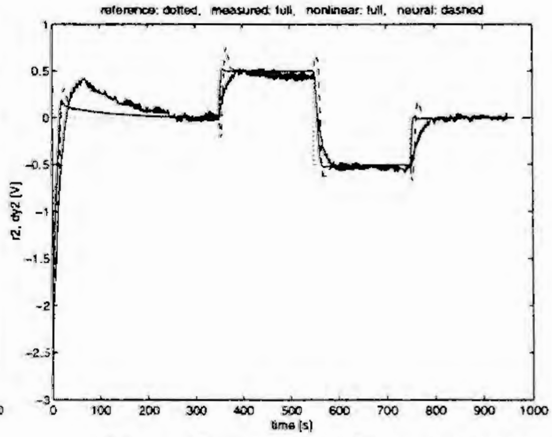


Figure 11: Level closed loop responses

Before closed loop validation, the weights *wij* (10) of the neural model were adapted on line with the real plant operating in closed loop. A gradient adaptation law was used to optimise the weights. Although neural model responses may seem quite similar to real plant responses, control signals when neural model was put into closed loop show obvious differences from control signals for the real plant.

Conclusions

Beside theoretical modelling, artificial neural networks were used to model a MIMO plant. The obtained models were not as accurate as it is generally expected for a SISO neural model. This remark proved true especially for the RAWN model, which not only was inaccurate but also exhibited stability problems. Empirically speaking, the RAWN algorithm has been found a very efficient tool for SISO modelling and its inability to model the dynamics of the plant was unforeseen. The Gaussian network model performed much better compared to the RAWN model, but its accuracy did not exceed the accuracy of the theoretical nonlinear model.

In the case presented, we first simulated the behaviour of already existing theoretical model and then used the obtained responses as training data for the neural model. Obviously, this approach is not very effective, since a great deal of information about the plant already compiled in form of the theoretical model is discarded while building a neural model. Only the information about number of plant inputs and outputs, order of the plant and simulated data is transferred to the neural model. All the information about inner structure of the plant contained in the theoretical model is dismissed. A very appealing possibility is to use the parallel hybrid modelling approach [7]. In this approach the theoretical model is used as-it-is and a neural Gaussian network is added in parallel to complement the existing theoretical model. This seems to be the most logical next step in modelling the pressure-level plant since the theoretical model is more accurate than neural models obtained so far. Other advantages of theoretical modelling should not be neglected, too, e.g. transparent model inner structure. On the other hand, the theoretical nonlinear model exhibits rather distinct discrepancies of level responses (fig. 5, 7). A hybrid parallel model could provide more accurate estimations of level responses.

References

1. Atanasijević-Kunc, M., Karba, R., and Zupančič, B.: "Multipurpose Modelling in the Evaluation of Laboratory Pilot Plant", *Proceedings of Eurosim '95*, 1995, pp. 855-860.

2. Braake H.A.B. te, van Straten, G.: "Random Activation Weight Neural Net for Fast Non-iterative Training". *Engng. Applic. Artif. Intell.*, Vol.8, No.1, 1995, pp. 71-80.

3. Milanič, S., Hecker, O., Karba, R.: "A Comparative Study of Neural Network Models for Model Based Predictive Control of a Thermal Plant". *CESA '96 IMACS Multiconference, Symposium on Modelling, Analysis and Simulation*, Lille, France, 1996, pp. 1244-1249.

4. Moody, J., Darken, C.J.: "Fast Learning in Network of Locally Tuned Processing Units". *Neural Computation*, Vol. 1, 1989, pp. 281-294.

5. Narendra, K.S., Parthasarathy, K.: "Identification and Control of Dynamical Systems Using Neural Networks". *IEEE Transactions on Neural Networks*, Vol.1, No.1, 1990.

6. Thiel, H.: "Economic Forecasting and Policy". North Holland, Amsterdam, 1970.

7. Thompson, M.L., Kramer, M.A.: "Modeling Chemical Processes Using Prior Knowledge and Neural Networks." *AIChE Journal*, Vol. 40, No. 8, August 1994, pp. 1328-1340.

8. Vetter, R.: "Design and Operation of UML, *Manual,*Leipzig, 1994.

IS IT REASONABLE TO THINK OF MULTIPLE-MODEL-BASED FLEXIBLE RECIPE CONTROL?

N. Hvala[1], S. Strmčnik[1], D. Šel[1] and S. Milanič[2]
[1]J. Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
e-mail: *nadja.hvala@ijs.si*
[2]Faculty of Electrical Engineering, University of Ljubljana,
Tržaška 25, 1000 Ljubljana, Slovenia

Abstract. The paper considers the implementation of flexible recipe control to the case study, *i.e.*, an industrial process of batch hydrolysis. The emphasis is given to the development of process model which is one of the most important parts of the flexible recipe control scheme. For the particular process three different models were developed, *i.e.*, a linguistic fuzzy model, a semi-empirical first-principle model, and a neural network model. Advantages and disadvantages of each model when used in a flexible recipe control scheme are presented. A combination of models is proposed as a most suitable solution. As possible approaches to model integration, multifaceted modelling approach and hybrid neural network modelling are considered.

1. Introduction

The paper deals with modelling of an industrial process of batch hydrolysis. The aim is to use the model for control purposes in order to improve the process performance related to quality and production requirements. The proposed control scheme is based on flexible recipe control concept which will be presented in more details in the paper. Flexible recipe control algorithms adjust the recipe parameters for each batch according to the current operating conditions. The concept strongly relies on process model. When applied to a particular case, the development of a process model is therefore one of the most important tasks.

For batch hydrolysis process considered in the paper, insufficient knowledge and information is available for developing the process model. Hence, the main question is what kind of a model should be developed in order to perform successfully the desired control tasks.

Considering the information about the process which come from different sources (knowledge, data, experience), and various tasks that should be performed based on the developed process model, a multifaceted approach to process modelling is a possible approach. Generally, multifaceted modelling represents modelling of one system from different aspects and on different levels of complexity. Multifaceted modelling advocates a multiplicity of partial models to support system objectives in contrast to one single comprehensive model which is extremely difficult to develop. Such models differ in level of abstraction and formalism. This leads to sets of overlapping and redundant representations. Concepts and tools are needed to organise such representations into a coherent whole.

Several authors address this question. [2] describes systematic derivation of related models based on the concept of systems morphisms. The concept is applied in event-based control. [6] considers two case studies where multifaceted modelling is used to support hierarchical design of chemical plants. Emphasis is given to internal consistency and communication among the models at the various levels of abstraction. [12] deals with computer assistance in multifaceted system modelling. The paper suggests principles for structuring the model and data bases in order to provide computer assistance in arbitrary composition and decomposition of systems.

In the paper, multifaceted modelling approach is discussed as a possible approach to modelling of batch hydrolysis. After introducing different models of batch hydrolysis, represented with differential equations, fuzzy sets and neural networks, special attention is given to the application of models in flexible recipe control scheme. In the discussion we try to find out whether the problem addressed in the paper suits into developed multifaceted modelling methodology. Hybrid neural network as another possibility of model integration is addressed at the end of the paper.

2. Flexible recipe control of batch hydrolysis process

2.1 Process description

Batch hydrolysis is one of the most important processes in the production of TiO_2 pigment from ilmenite ore. During this process TiO_2 particles are formed out of $TiOS0_4$ solution. The reaction occurs by adding seeds and water, and heating the solution to the boiling temperature. Reaction is described by

$$TiOSO_4 + H_2O \leftrightarrow TiO_2 + H_2SO_4. \quad (1)$$

The execution of each batch is performed according to the prescribed recipe which also determines the values of some important control variables, *i.e.*, the addition of seeds, the quantity and dynamics of added water, the duration of precipitation, *etc.* Most often these variables are set fixed in spite of variations of input solution which would require an adjustment of recipe parameters. As a result, the size and distribution of particles vary in a wide region and exceed the limits necessary to achieve the desired final quality of TiO_2 pigment.

Due to process complexity, the adjustment of recipe parameters cannot be performed by the operators. This is however a challenging task for the computer control system, which could compensate the variations of input solution by adjusting the recipe parameters. The adjustment can be performed by flexible recipe control based on process model [5,11].

2.2 Flexible recipe control scheme

Flexible recipes are aimed at improvement of the batch process performance. The idea is to compensate for the deviations in input variables and alterations in operating conditions, as well as to take into consideration the production costs which are all together described by the performance criterion. Compensation is carried out by altering the recipe parameters, so as to optimise the performance criterion. Two types of recipe improvement are possible, *i.e.*, recipe initialisation, where recipe parameters are altered before the beginning of the batch, and recipe correction, where recipe parameters are adjusted during the batch cycle. All recipe adjustments are based on process model by which the performance criterion of the current batch is computed and optimised for different settings of recipe parameters. The procedure for initialisation of recipe parameters is schematically shown in Fig. 1.

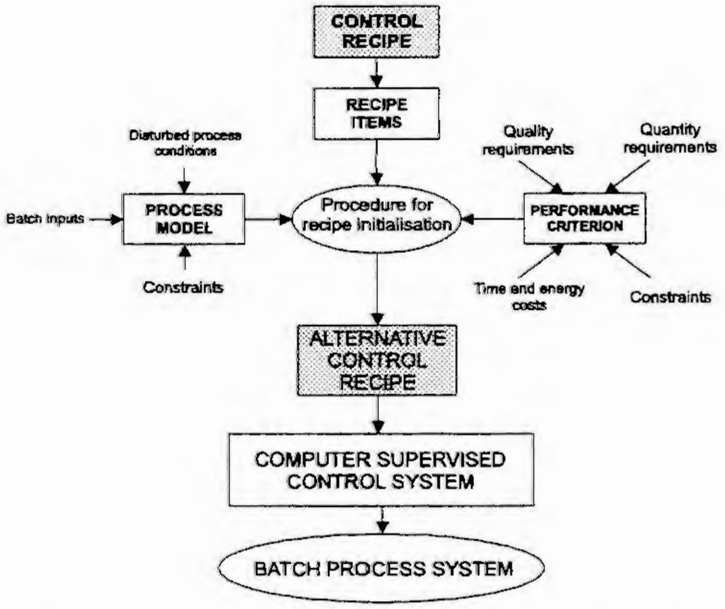


Fig. 1. Initialisation of recipe parameters performed by optimisation based on process model.

3. Models of batch hydrolysis

For the particular process of batch hydrolysis three models were developed, *i.e.*, a linguistic fuzzy model, a semi-empirical first-principle model, and a neural network model. Different modelling techniques were chosen in order to utilise all the process information available, *i.e.*, theoretical knowledge from chemical kinetic laws, experimental data from laboratory-plant experiments, experimental data from real-plant experiments, data from regular process operation, knowledge and experience from operation and control of the process.

3.1 Fuzzy model

Linguistic fuzzy model was developed at the beginning of the project when only qualitative knowledge extracted from laboratory-plant experiments was available. Based on experimental data, chemical engineers have identified three variables, *i.e.*, initial concentration of TiO_2, initial proportion of active acid to TiO_2 (AA/TiO_2) and the volume of added seeds, that influence the reaction rate and the particle size most significantly. From this a static input/output model with 3 inputs and one output was developed [7]. The influence of added seeds on reaction rate as well as the relation between reaction rate and particle size could be expressed in an explicit mathematical form. On the other hand, the relation between concentration of TiO_2, AA/TiO_2 and reaction rate was deduced from a small set of experiments and presented roughly in a graphical form. According to the type of knowledge available we have chosen fuzzy sets to represent this relation. The knowledge about the process was transformed directly into fuzzy rules, while membership functions were determined by trial and error in order to obtain good fit with process data. Fig. 2 represents reaction rate computed by fuzzy model for different AA/TiO_2 and TiO_2 concentrations. The figure is equivalent to the graphical representation of process behaviour given by chemical engineers. The rectangular represents the allowable region of process operation. Together with already mentioned deterministic part of the model, the fuzzy model determines the particle size as a function of model inputs (Fig. 3).

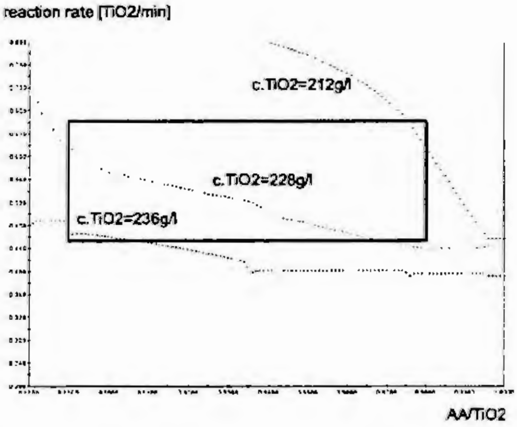


Fig. 2. Graphical representation of model relations given by chemical engineers and then represented by fuzzy model.

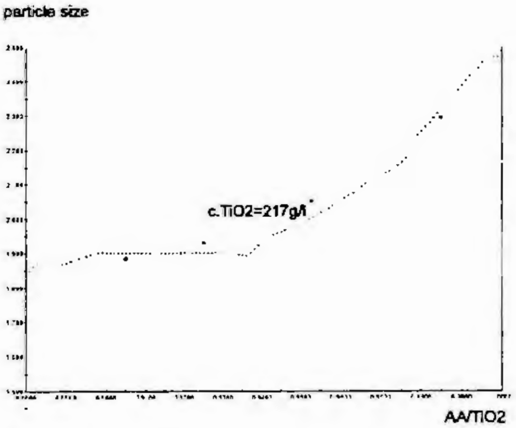


Fig. 3. Particle size computed by fuzzy model in comparison with measurements in laboratory plant experiments.

3.2 Semi-empirical first-principle model

Chemical engineers involved in the process however have more profound knowledge about the process concerning not only the input/output relations but also the reactions during the batch. They are able to describe approximately the course of most important process variables which can be modelled with chemical kinetic laws. From this the structure of a semi-empirical model was set in a form of differential equations.

The model output, *i.e.*, the precipitation *yield* is a function of time and is related to reaction (precipitation) rate *r(t)* according to the following equation

$$yield = \int r(t)dt. \tag{2}$$

According to chemical kinetics, reaction rate for chemical reaction $A + B \rightarrow C$ generally depends on reactant concentration A and B and is written as

$$r(t) = k(T)c_A^a c_B^b, \tag{3}$$

where a and b are empirically determined constants. *k(T)* is specific reaction rate that is modelled by Arrhenius equation

$$k = Ae^{-E/RT}, \tag{4}$$

where E represents activation energy, T temperature, A and R are constants.

Hydrolysis is reversible process. Therefore the concentration of a particular component C depends on reaction rate of production r_p and consumption r_c of this component and is generally given as

$$\frac{dc_C}{dt} = r_p - r_c. \tag{5}$$

Following these laws a set of differential equations was determined for each component in the hydrolysis process. Thereby it is necessary to note that final TiO_2 gel is produced via an intermediate product TiO_2*. In the nucleation process TiO_2* is partially transformed into seeds which together with the added seeds contribute to final TiO_2 gel formation from TiO_2*.

Using abbreviations $x=c_{TiOSO4}$, $v=c_{H2o}$, $z=c_{TiO2^*}$, $y=c_{H2SO4}$, $w=c_{TiO2gel}$, s is the volume of added seeds, r_6 is the added water and r_5 the activity of gel (constant), the set of equations is as follows:

$$\begin{aligned}
\frac{dx}{dt} &= k_2(z^c y^d) - k_1(x^a v^b) \\
\frac{dv}{dt} &= k_2(z^c y^d) - k_1(x^a v^b) + r_6 \\
\frac{dz}{dt} &= k_1(x^a v^b) - k_2(z^c y^d) + k_3(z^e y^f) \ - k_4(z^g s^{1/3}) + r_5 \\
\frac{dy}{dt} &= k_1(x^a v^b) - k_2(z^c y^d) \\
\frac{dw}{dt} &= k_4(y^g s^{1/3}) - r_5
\end{aligned} \tag{6}$$

Equations (6) represent semi-empirical process model that can be described as MISO model with 6 inputs and 1 output (TiO_2 gel concentration). Some of the inputs (*e.g.*, seeds, water) represent recipe parameters that are considered as control variables. The model parameters need to be estimated on process data. For that purpose a set of experiments was designed in order to access the course of reactions while varying the input concentrations and recipe parameters. In the model 5 parameters need to be estimated on data if the equations are simplified by setting all exponential parameters to 1.

Compared to fuzzy model, the developed semi-empirical model is a dynamic one and provides the information about the reaction components and consequently the reaction rate during the batch (Fig. 4), which is very important for correction algorithms, as described in the next chapter. Fig. 5 shows an example of simulated precipitation yield obtained by the model in comparison with measured data. Up to now, the model parameters were estimated on a small set of experiments. Currently a profound analysis based on a more representative set of experimental data is performed.

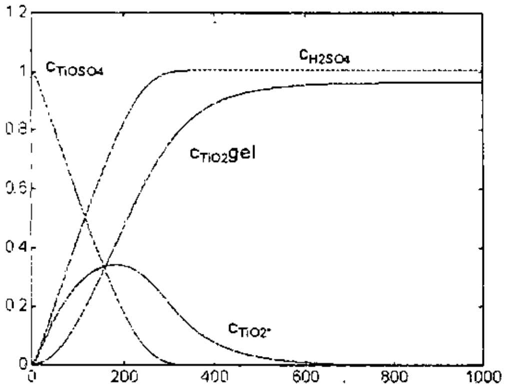


Figure 4. Process variables during one batch computed by semi-empirical model.

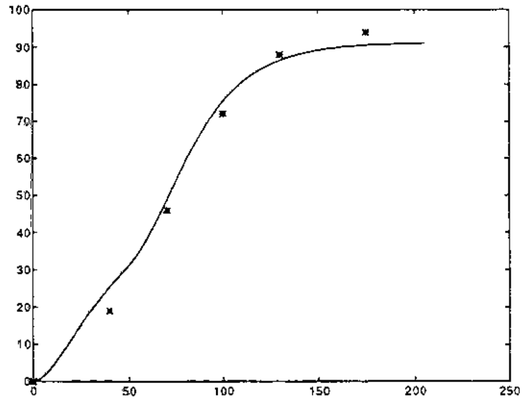


Fig. 5. Precipitation yield during one batch compared with measurements.

3.3 Neural network model

Generally, the development of a neural network model requires a lot of data. During normal operation of hydrolysis a lot of input-output data based on fixed recipe control are gathered. These data however do not include the variations of recipe parameters, which is necessary when designing a model for flexible recipe control. In order to identify the neural network model from real-plant data, the process has to be run for a longer period with small perturbations of the variables considered as model inputs.

As the neural network model is very efficient in an on-line control scheme due to its easy learning and adaptation capability, the current lack of proper data has been overcome by semi-empirical model, which had produced a learning data set for neural network model. The data included the values of input and output variables for different settings of recipe parameters and different operating conditions. The multilayer neural network model was designed. Two different training procedures gave almost the same approximation accuracy [3]. Simulation study has shown that the performance of the neural network model, when applied in a flexible recipe control scheme, is approximately the same as the performance of the semi-empirical process model [8].

4. The usage of models in flexible recipe control scheme

Although the main reason for development of different models was model comparison, it turned out that some types of models developed are more suitable to perform a particular task in the flexible recipe control scheme.

Fuzzy model is very appropriate for recipe initialisation due to its ability to represent the basic relations by if-then rules, which can be easily understood and verified by the operators. Together with easy implementation (look-up table) this can be of great importance when considering off-line or experimental usage, where it is necessary to check or even modify some of the model-based decisions.

As opposed to fuzzy model, which actually acts as a black-box model, the developed semi-empirical model is appropriate also for recipe correction algorithm. The model provides the information about the changes in the reaction rate (which cannot be directly measured in the process) as a consequence of different initial and operating conditions. The slope of the reaction curve is related to final quality and can be therefore used for correction of recipe parameters during the batch, when deviations are observed from the desired rate.

The neural network model can be used for the same purposes as the semi-empirical model. The advantage of this model is its ability to easy adapt to the changes in the real process. On-line model adaptation has to be performed in order to maintain model prediction capability which can otherwise fail due to inexact process model and long-term process changes. Compared to theoretical model with a pre-defined structure, better agreement with process data can be achieved with neural network model.

5. Possible approaches to integration of different models

From all mentioned above it can be concluded, that a model data base instead of a single model is preferred to perform different tasks in flexible recipe control of batch hydrolysis. This ranks the problem in the area of multifaceted modelling already mentioned in the introduction. Using such an approach some important questions need to be considered, *e.g.*, how to achieve systematic development of different models using different level of abstraction and different formalism; how to organise overlapping and redundant model representations into a coherent whole; how to develop different model representations so that they are consistent with each other and can be consistently modified?

In our particular case of hydrolysis model consistency seems to be of great importance, especially as all the models need to be on-line adapted during their usage in different flexible recipe control algorithms. In order to avoid modifying each process model, there is a need of an appropriate formalism which would allow consistent transition between the models. This question is however even more demanding due to fuzzy and neural networks representations. From the literature review we have found no references that would address this question.

More promising approach to integration of models is hybrid modelling. References of successful integration of theoretical and neural network models are reported [1,4,9,10]. Three different types of integration are possible:

- neural network model identified on augmented data set, the latter being produced from scarce experimental data and theoretical model,
- parallel approach where theoretical and neural network model are used in parallel; the neural network model is used to model the error of theoretical model,

- serial approach, where neural network model is used to estimate the theoretical model parameters or some variables that appear in the theoretical model.

For batch hydrolysis the first and/either the second approach can be used.

As the neural network model is difficult to identify from plant data, the first approach can be used to produce the augmented data set.

In on-line model-based control scheme the parallel approach can be used to combine good properties of both semi-empirical and neural network model:

- Semi-empirical process model derived from chemical kinetic laws is valid in a wide area and keeps predictive capability of the model in a wide operating region. It is however prone to modelling errors due to inexact and pre-defined model structure.
- On the contrary, the usage of neural network model is limited to the area where training data are available. Being flexible in structure, it gives very accurate non-linear mapping in this region, and is as such very suitable for modelling non-linear batch processes.

Combining these two models, the semi-empirical model serves for extrapolation, while the neural network model serves as an accurate non-linear mapping in regions, where plant data are available.

6. Conclusions

In the paper the development of models for control of batch hydrolysis process was presented. The models differ in modelling formalism and express different performance when used in initialisation, correction and model adaptation algorithms in flexible recipe control scheme. Disadvantages of each of the models suggest that integration of models is preferred to one single model. The multifaceted modelling methodology is a possible approach. The question of model consistency, being an essential requirement when dealing with a model data base, is however not well addressed in case of representations such as fuzzy sets and neural networks. Hybrid neural network modelling is considered as another possibility. In case of batch hydrolysis good predictive and adaptation capability are expected combining semi-empirical and neural network model. The procedure is now under development.

7. References

1. te Braake, H. A. B., van Can, H. J. L. and Verbruggen, H. B., Semi-Physical Modeling of Chemical Processes with Neural Networks. In: Prep. IFAC World Congress, San Francisco, 1996, 325-330.
2. Luh, C. J. and Zeigler, B. P., Abstracting Event-Based Control Models for High Autonomy Systems. IEEE Transactions on Systems, Man and Cybernetics, 23 (1993), 42-54.
3. Milanič, S., Šel, D., Hvala, N., Strmčnik, S. and Karba, R, Building Neural Network Models of Hydrolysis Process for Control Purposes. In: Prep. Modelling and Simulation ESM96, Budapest, Hungary, 1996, 700-704.
4. Psichogios, D. C. and Ungar, L. H., A Hybrid Neural Network-First Principles Approach to Process Modeling. AIChE Journal, 38 (1992), 1499-1511.
5. Rijnsdorp, J. E., Integrated Process Control and Automation. Elsevier, Amsterdam, 1991.
6. Stephanopoulos, G., Henning, G., and Leone, H., Model. LA. A Modelling Language for Process Engineering-II. Multifaceted Modeling of Processing Systems. Computers chem. Engng., 14 (1990), 847-869.
7. Šel, D., Strmčnik, S, Hvala, N. and Milanič, S., Model of Hydrolysis as Basis for Improvement of TiO_2 Pigment Production by Employing Flexible Recipes. In: Proc. Electrotechnical and Computer Science Conference ERK'95, Portorož, Slovenia, 1995, 337-340 (in Slovene).
8. Šel, D., Strmčnik, S., Hvala, N. and Milanič, S., Simulation Study of Flexible Recipe Implementation on a Batch Process. In: Prep. Modelling and Simulation ESM96, Budapest, Hungary, 1996, 336-340.
9. Thompson, M. L. and Kramer, M. A., Modeling Chemical Processes Using Prior Knowledge and Neural Networks. AIChE Journal, 40 (1994), 1328-1340.
10. Tsen, A. Y., Jang, S. S., Wong, D. S. H. and Joseph, B., Predictive Control of Quality in Batch Polymerization Using Hybrid ANN Models. AIChE Journal, 42 (1996), 455-465.
11. Verwater-Lukszo, Z., Otten, G. and Ingen Housz, T. J., Recipe Initialization for a batch process. In: Prep. IFAC Conference Integrated Systems Engineering, Baden-Baden, Germany, 1994, 93-97.
12. Zeigler, B., Elzas, M. S., Klir, G. J. and ören , T. I. (Eds.), Methodology in Systems Modelling and Simulation. North-Holland, Amsterdam, 1979.

FUZZY DIFFERENTIAL EQUATIONS

M. Oberguggenberger
Universität Innsbruck
Technikerstraße 13, A-6020 Innsbruck

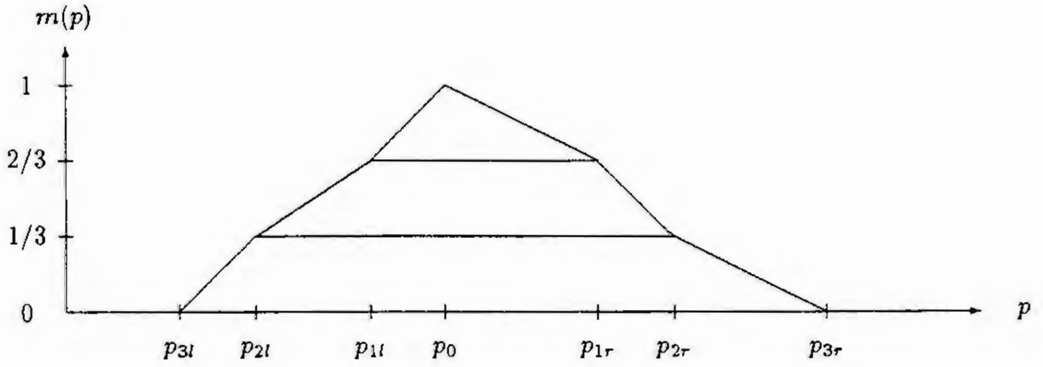
Abstract. In this paper solution concepts for systems of ordinary differential equations with fuzzy parameters are presented. Applying the Zadeh extension principle to the equations, the notions of a fuzzy solution and of a componentwise fuzzy solution are obtained. The fuzzy extension of the solution operator is shown to provide the unique solution in the former case, and the maximal solution in the latter case. In an interplay of interval analysis and possibility theory, these methods allow to process subjective information on the possible fluctuations of parameters in models involving ordinary differential equations.

1 Introduction

Fuzzy set theory is a powerful tool for modelling uncertainty and for processing vague or subjective information in mathematical models. While its main directions of development have been information theory, data analysis, artificial intelligence, decision theory, control, and image processing (see e.g. [1, 4, 9, 11, 12]), fuzzy set theory is increasingly used as a means for modelling and evaluating the influence of imprecisely known parameters in mathematical/technical/physical models. The purpose of this paper is to work out this approach when the models are constituted by systems of ordinary differential equations.

To set the stage, a short discussion of what we mean by fuzzy data appears appropriate. Imprecise knowledge of parameters and their fluctuations can be traced to various reasons, for example: lack of knowledge of boundary conditions; simplification in complex circumstances forcing a single parameter to cover a wider range of situations; lack of a precisely quantifiable definition of some verbally defined variable; stochastic variations of the outcome of repeated realizations of identical experiments (the latter situation, which is amenable to a probabilistic description, is but one special case of uncertainty).

The basic premise of the fuzzy approach is that vague data can be characterized by intervals of variation, supplied with a valuation. To describe the basic ideas, let us (preliminarily) use the language of subjective risk assessment. Assume that under normal conditions some parameter has a certain value , say p_0. Given risk 1 (high), it may fluctuate in an interval $[p_{1l}, p_{1r}]$, under risk 2 (medium) in a larger interval $[p_{2l}, p_{2r}]$, under risk 3 (low) in a yet larger interval $[p_{3l}, p_{3r}]$. We mark these intervals in a diagram, scaling the risks (arbitrarily) to $high = 2/3, medium = 1/3, low = 0$:



Joining the endpoints of the marked intervals, we obtain a valuation function $m(p)$ representing a qualitative description of the fluctuations of the parameter, given our risk assessment. The resulting contour can be interpreted in a number of ways.

1. *α-cuts:* The horizontals of height α correspond to risk level α. The projection of their intersection with the enveloping contour yields the interval of fluctuation at risk level α.

2. *Degree of possibility:* For any value p, the height $m(p)$ of the envelope is the degree of possibility that the given parameter assumes the value p.

3. *Multivalued logic:* The degree of possibility $m(p)$ can be interpreted as the truth value of the assertion "the parameter value is p".

It should be pointed out that $m(p)$ is generally not a probability, but rather a quantization of the assessment of the possible fluctuations of the parameter p. For more detailed information on designing the valuation function $m(p)$ we refer to [1, 4, 6, 11].

The crucial further ingredient in the fuzzy approach is that it admits all arithmetical operations with fuzzy parameters, as we shall see in Section 2. Thus the information and assessment can be processed in mathematical models and will be faithfully reflected in the results. The tool is the Zadeh extension principle [10] which is based on the possibility theoretic interpretation of the fuzzy description.

Thus fuzzy set theory, on the one hand, goes beyond interval analysis as it adds valuations together with the possibilistic methods. On the other hand, it differs from probability theory in its modelling assumptions, approach, and assertions. It can be applied in situations where a probabilistic interpretation is impossible, whether the fluctuations are non-stochastic in nature, no statistical data are available, or the axiomatic structure of a probabilistic model cannot be justified from the known information.

Based on the fuzzy description of parameters and mathematical objects, we shall be concerned here with systems of ordinary differential equations of the form

$$\begin{aligned} \dot{x}(t) &= F(t, x(t), b) \\ x(t_0) &= a\,. \end{aligned}$$

Here the initial data a and parameters b appearing in the function F will be fuzzy numbers. The solution $x(t)$ at any fixed point of time t will be a fuzzy number as well. The approach most natural from the view-point of possibility theoretical modelling is to apply the extension principle to the solution operator L, mapping the parameters a, b into the solution x. Thereby, the influence of the fluctuations of the input parameters on the result can be computed, together with their valuations. On the other hand, the extension principle can also be applied to the differential equation, viewing $t \to x(t)$ as a fuzzy function. Depending on how we interpret the initial data, we arrive at the concept of a fuzzy solution, respectively componentwise fuzzy solution. We are going to show that the solution arising from extending the solution operator L is the unique fuzzy solution, respectively the componentwise fuzzy solution with maximal degree of possibility. We shall also discuss computational aspects and intended applications.

What concerns other approaches in the literature, we mention that properties of the fuzzy solution operator have been studied in [2]. On the other hand, differentiation theory of multivalued maps and imbedding fuzzy sets into metric spaces is used in [3, 8]. This latter approach is not equivalent to ours; the corresponding solutions are not given by the fuzzy extension of the solution operator (see [3, Example 13.1.1]). For a discussion of using the derivatives of the bounding curves of the α-level sets, if they exist, we refer to [5].

2 Fuzzy sets

Given a basic set os discourse, a *fuzzy set A in X* is defined as a map

$$m_A : X \to [0,1]\,.$$

In this paper we shall always work with *normalized fuzzy sets*, requiring that

$$\{x \in X : m_A(x) = 1\} \neq \emptyset\,.$$

The set of fuzzy sets in X will be denoted by $\mathbb{F}(X)$. In accordance with the Introduction, given $x \in X$, $m_A(x)$ can be interpreted as

- the membership degree of the element x belonging to A;

- the truth value of the statement that x belongs to A;

- the degree of possibility that the variable A takes the value x.

The *α-level sets* A^α are the classical (crisp) subsets

$$A^\alpha = \{x \in X : m_A(x) \geq \alpha\}$$

of X. A *fuzzy real number* is an element $A \in \mathbb{F}(\mathbb{R})$ such that all level sets A^α are compact intervals for $0 < \alpha \leq 1$. Classical (crisp) real numbers a as well as compact intervals $[a, b]$ can be viewed as fuzzy real numbers with membership function the indicator function of a, respectively $[a, b]$.

Given a function $f : X \rightarrow Y$, the Zadeh extension principle allows f to be extended to a map $f : \mathbb{F}(X) \rightarrow \mathbb{F}(Y)$ in the following way: To every fuzzy set A in X the fuzzy set $f(A)$ in Y is assigned, with membership function

$$m_{f(A)}(y) = \sup\{\, m_A(x) : x \in f^{-1}(\{y\})\,\}$$

for $y \in Y$, with the provision that

$$m_{f(A)}(y) = 0 \text{ if } f^{-1}(\{y\}) = \emptyset.$$

The extension principle is functorial, that is, given $f : X \rightarrow Y$, $g : Y \rightarrow Z$ and $A \in \mathbb{F}(X)$, we have

$$(f \circ g)(A) = f(g(A))\,.$$

In case $X = X_1 \times \cdots \times X_n$ is a product set, we continue to write A for the elements of $\mathbb{F}(X) = \mathbb{F}(X_1 \times \cdots \times X_n)$, but we shall use vector notation $\vec{A}$ for the elements $(A_1, \cdots, A_n)$ of $\mathbb{F}(X_1) \times \cdots \times \mathbb{F}(X_n)$. The *tensor product map*

$$\otimes : \mathbb{F}(X_1) \times \cdots \times \mathbb{F}(X_n) \rightarrow \mathbb{F}(X_1 \times \cdots \times X_n)$$

defined by

$$m_{\otimes \vec{A}}(x_1, \cdots, x_n) = \min\{m_{A_j} : j = 1, \cdots, n\}$$

aggregates the vector $\vec{A} = (A_1, \cdots, A_n)$ of fuzzy sets to the single fuzzy set $A = \otimes\vec{A}$ in $\mathbb{F}(X)$. The possibility degree of a point $(x_1, \cdots, x_n)$ to belong to $\otimes\vec{A}$ is the minimum of the possibilities that each x_j belongs to A_j. This corresponds to the situation where no information on the interaction of the variables $x_1, \cdots, x_n$ is available. By the extension principle we can extend the projections $\Pi_k : X_1 \times \cdots \times X_n \rightarrow X_k$ to

$$\begin{aligned} \Pi_k &: \mathbb{F}(X_1) \times \cdots \times \mathbb{F}(X_n) \rightarrow \mathbb{F}(X_k) : \\ m_{\Pi_k(A)}(y) &= \sup\{m_A(x_1, \cdots, y, \cdots, x_n) : x_j \in X_j,\, j \neq k\}\,. \end{aligned}$$

Due to the fact that our fuzzy sets are normalized, we have that

$$\Pi_k(\otimes\vec{A}) = A_k$$

for all $\vec{A} = (A_1, \cdots, A_n) \in \mathbb{F}(X_1) \times \cdots \times \mathbb{F}(X_n)$.

Consider now a map $f : \mathbb{R}^n \rightarrow \mathbb{R}$. We can extend it to a map

$$f : \mathbb{F}(\mathbb{R}^n) \rightarrow \mathbb{F}(\mathbb{R})\,,$$

and so f acts on fuzzy subsets of $\mathbb{R}^n$. In many situations, information on the fluctuations of parameters $x_1, \cdots, x_n$ is given separately by fuzzy subsets $A_j \in \mathbb{F}(\mathbb{R}), j = 1, \cdots, n$. This case can be dealt with by composing f with the tensor product map, yielding

$$\begin{aligned} f^{\otimes} &: (\mathbb{F}(\mathbb{R}))^n \rightarrow \mathbb{F}(\mathbb{R}) : \\ m_{f^{\otimes}\vec{A}}(y) &= \sup\{\min(m_{A_1}(x_1), \cdots, m_{A_n}(x_n)) : y = f(x_1, \cdots, x_n)\}\,. \end{aligned}$$

The possibilistic interpretation is especially intuitive: To determine the degree of possibility π of y to belong to $f^{\otimes}\vec{A}$, consider all combinations $(x_1, \cdots, x_n)$ producing $y = f(x_1, \cdots, x_n)$. The possibility degree of each combination is the minimum of the individual degrees $m_{A_j}(x_j)$, while π is the largest value obtainable this way. It turns out that this corresponds to the composition of α-level sets. Indeed, when f is continuous and $A_1, \cdots, A_n$ are fuzzy real numbers, one can show [4] that $f^{\otimes}(A_1, \cdots, A_n)$ is a fuzzy number as well and

$$(f^{\otimes}(A_1, \cdots, A_n))^\alpha = f(A_1^\alpha, \cdots, A_n^\alpha)\ , \alpha > 0\,, \tag{1}$$

where the right hand side is the set theoretic image of the intervals $A_1^\alpha, \cdots, A_n^\alpha$. This observation brings us full circle to the risk theoretic interpretation of the Introduction. If at risk level α the parameters $x_1, \cdots, x_n$ fluctuate in the intervals $A_1^\alpha, \cdots, A_n^\alpha$, then the function value fluctuates in the interval $f(A_1^\alpha, \cdots, A_n^\alpha)$, and this is precisely what the extension principle says in this situation. In particular, formula (1) is the basis for numerical computations with fuzzy sets, reducing the calculations to interval arithmetic on each α-level set.

Remark 1: Caution is needed when substituting variables in the extension principle. For example, we can write the zero function $f : \mathbb{R} \to \mathbb{R}$ as $f(x) = x - x$. Of course, the extension of the zero function to $\mathbb{F}(R)$ is the zero function. However, if we extend the function $\mathbb{R}^2 \to \mathbb{R} : (x, y) \to x - y$ to $(\mathbb{F}(\mathbb{R}))^2$ and then substitute $x = y$, we may obtain a nonzero result: Take, for example, the intervals $A = [0,1], B = [0,1]$; then $A - B = [0,1] - [0,1] = [-1,1] \neq \{0\}$. In the latter case, A and B are considered as realizations of two independent parameters, so the result accumulates all fluctuations. Therefore, before applying the extension principle to a formula representing fluctuations of certain parameters, it must be re-interpreted so that each independent parameter appears only once (see [4]).

3 Fuzzy differential equations

Consider the $(n \times n)$-system of differential equations

$$\begin{aligned} \dot{x}(t) &= F(t, x(t), b) \\ x(0) &= a \end{aligned}$$

with parameters $a \in \mathbb{R}^n, b \in \mathbb{R}^m$. We assume that for each $b \in \mathbb{R}^m, F$ is smooth and globally Lipschitz with respect to the second variable, uniformly for t in compact sets. Then global unique solutions always exist, and we have the solution operator

$$L : \mathbb{R}^n \times \mathbb{R}^m \to C^1(\mathbb{R})^n\,,$$

mapping each parameter (a, b) to the solution x. To handle fuzzy parameters, we wish to apply the extension principle both to the equation and the solution operator. However, substituting $x = L(a, b)$ in the equation we observe that the paramter b will appear twice, so that we run precisely into the difficulty outlined in Remark 1. For this reason, we (i) rewrite the system so that the parameters appear in the initial data only, by introducing additional variables $x_{n+1}, \cdots, x_{m+n}$ with $\dot{x}_{m+j} = 0$, $x_{m+j}(0) = b_j$ and (ii) consider the system of equations as an entity. Thus we shall work with systems

$$\begin{aligned} \dot{x}(t) &= F(t, x(t)) \\ x(0) &= a \end{aligned}$$

only, introducing the equation operator

$$E \;:\; C^1(\mathbb{R})^n \to C(\mathbb{R})^n \;:\; x \to [t \to \dot{x}(t) - F(t, x(t))]\,,$$

the restriction operator

$$R \;:\; C^1(\mathbb{R})^n \to \mathbb{R}^n \;:\; x \to x(0)\,,$$

and the solution operator

$$L \;:\; \mathbb{R}^n \to C^1(\mathbb{R})^n \;:\; a \to x = L(a)\,.$$

By the extension principle, we have

$$E \;:\; \mathbb{F}(C^1(\mathbb{R})^n) \to \mathbb{F}(C(\mathbb{R})^n);\; R \;:\; \mathbb{F}(C^1(\mathbb{R})^n) \to \mathbb{F}(\mathbb{R}^n);\; L \;:\; \mathbb{F}(\mathbb{R}^n) \to \mathbb{F}(C^1(\mathbb{R})^n)\,.$$

Definition 1 *An element $X \in \mathbb{F}(C^1(\mathbb{R})^n)$ is called a fuzzy solution with initial value $A \in \mathbb{F}(\mathbb{R}^n)$, if*

$$EX = 0 \text{ in } \mathbb{F}(C(\mathbb{R})^n),\;\; RX = A \text{ in } \mathbb{F}(\mathbb{R}^n)\,. \tag{2}$$

Here 0 denotes the crisp zero function in $\mathbb{F}(C(\mathbb{R})^n)$, that is, $m_0(x) = 1$ if $x = 0, m_0(x) = 0$ otherwise.

Proposition 1 *Given $A \in \mathbb{F}(\mathbb{R}^n), X = L(A)$ is a fuzzy solution to problem (2).*

Proof: Since $R(L(a)) = a$ for $a \in \mathbb{R}^n$, we have by functoriality that $R(L(A)) = A$ for $A \in \mathbb{F}(\mathbb{R}^n)$ as well. To show that $X = L(A)$ solves the fuzzy differential equation, we compute

$$\begin{aligned} m_{EL(A)}(z) &= \sup\{m_{L(A)}(y) : z = E(y)\} \\ &= \sup\{\sup\{m_A(a) : y = L(a)\} : z = E(y)\}\,. \end{aligned}$$

If $z \neq 0$ and $z = E(y)$, then $\{a \in \mathbb{R}^n : y = L(a)\} = \emptyset$, so the inner supremun is zero, and $m_{EL(A)}(z) = 0$. If $z = 0$, we may take some $a \in \mathbb{R}^n$ with $m_A(a) = 1$ and let $y = L(a)$. Then $0 = E(y)$ and so the supremum equals 1. □

Let $S = \{x \in C^1(\mathbb{R})^n : E(x) = 0\}$. We can view $\mathbb{F}(S)$ as a subset of $\mathbb{F}(C^1(\mathbb{R})^n)$, setting the membership degree of any $x \in C^1(\mathbb{R})^n \setminus S$ to some $X \in \mathbb{F}(S)$ equal to zero.

Lemma 1 *If $X \in \mathbb{F}(C^1(\mathbb{R})^n)$ is a solution to (2), then X belongs to $\mathbb{F}(S)$.*

Proof: We have that $m_{EX}(z) = \sup\{m_X(y) : z = E(y)\}$. Suppose there exists $x \notin S$ such that $m_X(x) > 0$. Putting $z = E(x)$ we have $m_{EX}(z) \geq m_X(x) > 0$, contradicting the hypothesis that $EX = 0$. □

Proposition 2 *The fuzzy solution $X \in \mathbb{F}(C^1(\mathbb{R})^n)$ to (2) is unique.*

Proof: Since $L : \mathbb{R}^n \to S$ is bijective, the same is true of the extension $L : \mathbb{F}(\mathbb{R}^n) \to \mathbb{F}(S)$ by functoriality. If $X \in \mathbb{F}(C^1(\mathbb{R})^n)$ is a solution, it belongs to $\mathbb{F}(S)$ by the Lemma and hence is uniquely determined by the initial data. □

We now turn to the common situation that the parameters $a = (a_1, \cdots, a_n)$ describing the data of the problem are given as separate fuzzy numbers $\vec{A} = (A_1, \cdots, A_n) \in (\mathbb{F}(\mathbb{R}))^n$. In this case, $L^{\otimes}\vec{A}$ is an appropoate candidate for a solution to the differential equation. We split the restriction operator R in its components $R_j = \Pi_j \circ R : C^1(\mathbb{R})^n \to \mathbb{R}, j = 1, \cdots, n$, and extend it to an operator $\vec{R} = R_1 \times \cdots \times \mathbb{R}_n : \mathbb{F}(C^1(\mathbb{R})^n) \to (\mathbb{F}(\mathbb{R}))^n$.

Definition 2 *An element $X \in \mathbb{F}(C^1(\mathbb{R})^n)$ is called a componentwise fuzzy solution with initial data $\vec{A} \in (\mathbb{F}(\mathbb{R}))^n$, if*

$$EX = 0 \text{ in } \mathbb{F}(C(\mathbb{R})^n),\ \vec{R}X = \vec{A} \text{ in } (\mathbb{F}(\mathbb{R}))^n\,. \tag{3}$$

Proposition 3 *Given $\vec{A} \in (\mathbb{F}(\mathbb{R}))^n, X = L^{\otimes}(\vec{A})$ is a componentwise fuzzy solution to (3).*

Proof: Let $A = \otimes\vec{A}$. Then $X = L(A)$ satisfies $EX = 0, RX = A$ by Propostion 1. As noted in Section 2, $R_jX = \Pi_j RX = \Pi_j A = A_j$, so X takes the initial data in the required sense. □

Componentwise fuzzy solutions are no longer unique, as can be seen from simple examples. However, we have the following result which was first proved in [7] in the case of computation of antiderivatives:

Proposition 4 *$L^{\otimes}(\vec{A})$ is the componentwise solution with maximal membership degree.*

Proof: Let X be a componentwise fuzzy solution. Then

$$m_{A_j}(r) = m_{R_jX}(r) = \sup\{m_X(y) : r = R_j(y)\} = \sup\{m_X(y) : r = y_j(0)\}\,.$$

If X belongs to $C^1(\mathbb{R})^n$, then

$$m_X(x) \leq \sup\{m_X(y) : y \in C^1(\mathbb{R})^n, x_j(0) = y_j(0)\} = m_{A_j}(x_j(0)) \text{ for } j = 1, \cdots, n\,.$$

Thus

$$\begin{aligned} m_X(x) &\leq \min(m_{A_1}(x_1(0)), \cdots, m_{A_n}(x_n(0))) \\ &\leq \sup\{\min(m_{A_1}(a_1), \cdots, m_{A_n}(a_n)) : x = L(a_1, \cdots, a_n)\} = m_{L^{\otimes}\vec{A}}(x)\,, \end{aligned}$$

noting that $x = L(x_1(0), \cdots, x_n(0))$. □

4 Summary

The results of Sectiou 3 show that from both the viewpoint of fuzzy solutions and componentwise fuzzy solutions, the extension of the classical solution operator provides an appropriate (unique respectively maximal) solution with fuzzy data. In addition, the interpretation using α-level cuts shows that this concept is in accordance with risk assessment ideas: If the initial parameters vary in certain intervals $A_j^\alpha, j = 1, \cdots, n$ at risk level α, the variations in the solution are covered precisely by the family of $\mathcal{C}^1$-functions with membership degree α to the fuzzy solution $X = L^{\otimes}(\vec{A})$. This information can be processed further by evaluating functionals of the solution of interest. As an example, we can consider the fuzzy pointvalue at a later point of time t, say $X(t)$, with membership function

$$m_{X(t)}(r) = \sup\{m_X(x) : r = x(t)\} .$$

Numerically, these fuzzy sets can be computed by evaluating the set theoretic image $L(A_1^\alpha, \cdots, A_n^\alpha)(t)$. In case $A_1^\alpha, \cdots, A_n^\alpha$ are fuzzy numbers, this is just the interval $[\min\{L(a)(t) : a \in A_1^\alpha \times \cdots \times A_n^\alpha\}, \max\{L(a)(t) : a \in A_1^\alpha \times \cdots \times A_n^\alpha\}]$. Less costly search algorithms have been initiated in [7] and are currently being investigated. These results will be published elsewhere.

Typical applications we have in mind are the systems of differential equations arising in economics, biology, and engineering, where information on the input coefficients can often best be represented by fuzzy numbers. As a very simple example, consider a Verhulst population model

$$\dot{x}(t) = ax(t)(b - x(t)), \; x(0) = c$$

where the initial population c is known precisely, while the growth rate a and especially the value of the limiting population b is subject to various assumptions on the future development and thus can be described only up to variations in certain ranges. Subjective assessments of these parameters can be assembled as outlined in the Introduction (and the references quoted there) to produce a representation by means of fuzzy numbers. The fuzzy solution will faithfully describe the population at a later point of time $t > 0$, given the information available initially.

References

1. Bandemer, H. and Näther, W., Fuzzy Data Analysis. Kluwer, Dordrecht, 1992.
2. Bontempi, G., Modeling with uncertainty in continuous dynamical systems: the probability and possibility approach. Preprint, Université libre de Bruxelles, 1995.
3. Diamond, P. and Kloeden, P., Metric Spaces of Fuzzy Sets. World Scientific, Singapore, 1994.
4. Dubois, D. and Prade, H., Possibility Theory. Plenum Press, New York, 1988.
5. Kandel, A., Fuzzy Mathematical Techniques with Applications. Addison-Wesley, Reading, 1986.
6. Lessmann, H., Mühlögger, J. and Oberguggenberger, M., Netzplantechnik mit unscharfen Methoden. Bauingenieur 69 (1994), 469 - 478.
7. Pittschmann, S., Lösungsmethoden für Funktionen und gewöhnliche Differentialgleichungen mit unscharfen Parametern. Diplomarbeit, Universität Innsbruck, 1996.
8. Puri, M. and Ralescu, D., Differentials of fuzzy functions. Journal of Mathematical Analysis and Applications, 91 (1983), 552 - 558.
9. Terano, T., Asai, K. and Sugeno, M., Fuzzy Systems Theory and its Applications. Academic Press, Boston, 1992.
10. Zadeh, L., Fuzzy sets. Information and Control, 8 (1965), 338 - 353.
11. Zimmermann, H.-J., Fuzzy Set Theory and its Applications. Kluwer, Dordrecht, 1991.
12. Zimmermann, H.-J., Fuzzy Sets, Decision Making, and Expert Systems. Kluwer, Dordrecht, 1993.

FINITE ELEMENT METHOD WITH FUZZY PARAMETERS

Thomas Fetz
Institut für Mathematik und Geometrie, Universität Innsbruck
Technikerstr. 13, A-6020 Innsbruck
E-mail: fetz@mat1.uibk.ac.at

Abstract. Values of soil parameters entering soil mechanical models are only *vaguely* determined. In spite of this, the *finite element* computations which are in common use are performed with *crisp* data and yield results which only seem to be exact. In modelling this uncertainty, the theory of *fuzzy sets* can be viewed as an extension of interval arithmetic, which in addition provides a possibility theoretic valuation. So information on vague data can be processed in cases where no probability distributions are known or applicable. The purpose of this paper is to develop an algorithm in order to solve finite element problems with fuzzy parameters and to discuss the problems arising in the interpretation and visualization of the solution.

Introduction

Uncertainties and a lack of information are important problems arising in soil mechanical modelling. At the beginning there is often a vague linguistically formulated description of the characteristics of the soil, e.g. "the soil consists of clay, which is wet but not very wet" and so on. On the other hand the civil engineer, who has to determine the dimensions of a foundation needs crisp precise data for performing his finite element computation. We enumerate already used methods for handling uncertainty:

1. Mean values: Today the usual proceeding is to estimate a mean value out of the vague data for further computation. It is obvious, that the reliability of the result is not very high, because the vagueness is fully neglected.

2. Calculations with different input values: This is an improvement of the latter, but commonly one just takes all combinations of the lowest and highest possible input values. If the result does not monotonically depend on the input parameters, by these means we get an incorrectly small variation of the output values.

3. Interval arithmetic: Here we omit the latter problem using all possible combinations (in practice often 'enough' combinations). But there is no valuation of the input, so additional information about the input, e.g. some values seem to be more preferable than other ones, is not used. For an introduction to interval analysis see [11].

4. Stochastic methods: Using probability measures requires to have a probability theoretic model and enough and adequate statistical data. These can be provided for the materials of a building itself, but not for the soil, on which it is built. In soil mechanics there are often too few statistical information and too few comparable situations for using stochastic methods. Nevertheless an introduction to the stochastic finite element method is [9].

In this paper we employ *fuzzy numbers* [1, 3, 4, 18] for an improvement of (3). A fuzzy number can be viewed as a set of valued intervals, where the lower valued intervals contain the higher valued ones. So we get what was lacking in (3). In addition, fuzzy set theory allows to translate linguistic expressions such as "medium grained" and "high elastic modulus" directly into fuzzy soil parameters. Applying this theory we can extend commonly used crisp models to models which can handle vagueness in an appropriate way.

There are already some papers about modelling uncertainty in several applications in civil engineering with the fuzzy set theory [2, 10]. In [16, 17] there is also a method for using fuzzy numbers in combination with the finite element method described, but in these papers problems which arise in interval and fuzzy arithmetic are unfortunately neglected.

[5] is a report about a workshop at the technical faculty in Innsbruck concerning fuzzy sets in soil mechanical models of tunnels.

Fuzzy numbers

Let X be a universe of discourse. Membership in a classical subset A of X is often viewed as a characteristic function m_A from X to $\{0,1\}$ such that $m_A(x) = 1$ if $x \in A$ and $m_A(x) = 0$ if $x \notin A$. Consider now a subset $\tilde{A}$ of X, where the transition between membership and nonmembership is gradual rather than abrupt. The so called *fuzzy* set $\tilde{A}$ has no well-defined bounderies and it is defined as a map (cf. [4])

$$m_{\tilde{A}} : X \longrightarrow [0,1]. \tag{1}$$

The following interpretations of the map $m_{\tilde{A}}(x)$ are possible:

1. $m_{\tilde{A}}(x)$ is the degree of membership. The closer the value $m_{\tilde{A}}(x)$ is to 1, the more x belongs to A.
2. $m_{\tilde{A}}(x)$ is the degree of possibility. The closer the value $m_{\tilde{A}}(x)$ is to 1, the higher the possibility (not probability) is, that the variable $\tilde{A}$ represents the value x.
3. $\tilde{A}$ consists of subsets (α-cuts) which are valued by $m_{\tilde{A}}(x)$.

An important role for the handling of fuzzy sets plays the α-cut

$$[\tilde{A}]_\alpha = \begin{cases} \{x \in X, m_{\tilde{A}}(x) \geq \alpha\} & \alpha > 0 \\ \operatorname{supp} m_{\tilde{A}}(x) & \alpha = 0, \end{cases} \tag{2}$$

which is an ordinary set. Let X be the set of real numbers. A fuzzy number $\tilde{x} \in \mathbb{F}(\mathbb{R})$, where $\mathbb{F}(\mathbb{R})$ is the set of all fuzzy numbers, is defined by an upper semi-continous map $m_{\tilde{x}}$, such that the α-cuts $[\tilde{x}]_\alpha = \left[[\tilde{x}]_\alpha^L, [\tilde{x}]_\alpha^R\right]$ are single intervals in $\mathbb{R}$.

For handling more than one fuzzy variable we need the Cartesian product. Let $\tilde{x}_1, \ldots, \tilde{x}_n$ be fuzzy numbers. The Cartesian product $\tilde{x}_1 \times \cdots \times \tilde{x}_n$ is defined as

$$m_{\tilde{x}_1 \times \cdots \times \tilde{x}_n}(x_1, \ldots, x_n) = \min\left(m_{\tilde{x}_1}(x_1), \ldots, m_{\tilde{x}_n}(x_n)\right). \tag{3}$$

We evaluate a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ on n fuzzy arguments in $\mathbb{F}(\mathbb{R})$ by means of the *extension principle*. The extension principle introduced by Zadeh is one of the most basic ideas of fuzzy set theory. It provides a general method for extending crisp mathematical concepts in order to deal with fuzzy quantities. Here it allows us to induce from n fuzzy numbers $\tilde{x}_i$ a fuzzy vector $\tilde{y} \in \mathbb{F}(\mathbb{R}^m)$ through f such that

$$m_{\tilde{y}}(y) = \sup_{\substack{x_1, \ldots, x_n \\ y = f(x_1, \ldots, x_n)}} \min\left(m_{\tilde{x}_1}(x_1), \ldots, m_{\tilde{x}_n}(x_n)\right) \tag{4}$$

$$m_{\tilde{y}}(y) = 0 \quad \text{if} \quad f^{-1}(y) = \emptyset. \tag{5}$$

$m_{\tilde{y}}(y)$ is the greatest among the membership values $m_{\tilde{x}_1 \times \cdots \times \tilde{x}_n}(x_1, \cdots, x_n)$ of the realizations of $y \in \mathbb{R}$ using n-tupels $(x_1, \ldots, x_n)$, cf. [4].

Caution: We have started with an object in $\mathbb{F}(\mathbb{R})^n$ and get an object in $\mathbb{F}(\mathbb{R}^m)$ and not in $\mathbb{F}(\mathbb{R})^m$. $\tilde{x} \in \mathbb{F}(\mathbb{R})^n$ is a n-dimensional vector with components in $\mathbb{F}(\mathbb{R})$, but $\tilde{y} \in \mathbb{F}(R^m)$ is a more complicated fuzzy quantity, because there may be *interactions* between the components. For better handling $\tilde{y}$ is often replaced by the hull $\tilde{z} \in \mathbb{F}(\mathbb{R})^m$ which is the smallest fuzzy quantity in $\mathbb{F}(\mathbb{R})^m$ containing $\tilde{y}$. In terms of α-cuts: $[\tilde{y}]_\alpha$ is replaced by the smallest Cartesian product $[\tilde{z}_1]_\alpha \times \ldots \times [\tilde{z}_m]_\alpha$ containing $[\tilde{y}]_\alpha$ where

$$[\tilde{z}_i]_\alpha = [\min\{y_i : y \in [\tilde{y}]_\alpha\}, \max\{y_i : y \in [\tilde{y}]_\alpha\}]. \tag{6}$$

Neglecting interactivity in such a way means treating each output variable separatly from the others.

Evaluating a continous function f by the extension principle amounts to using α-cuts:

$$[f(\tilde{x}_1, \ldots, \tilde{x}_n)]_\alpha = f([\tilde{x}_1]_\alpha, \ldots, [\tilde{x}_n]_\alpha). \tag{7}$$

So fuzzy calculation is reduced to interval calculation on the α-cuts.

A simple linear elastic problem

Let us introduce the combination of the *Finite Element Method* and *Fuzzy Set Theory* using a simple example similar to [16], which shows the calculation and also the interpretation and visualization of the solution. Consider a cross-section through a linear elastic soil medium, on which an uniformly distributed load is applied.

The situation including load, boundary conditions and finite element mesh is shown in figure 1. For the fuzzy elastic modulus and the fuzzy Poisson's ratio we use triangular fuzzy numbers drawn in figure 2.

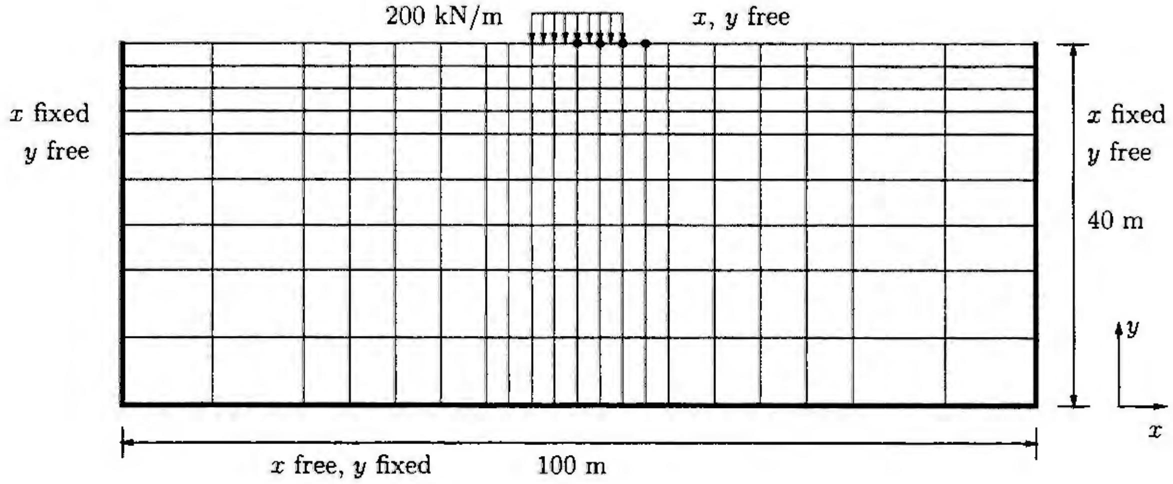


Figure 1: cross-section and finite element mesh

First have a look at the crisp case. The solution of a linear elastic problem using finite elements leads to the following system of linear equations [7, 12, 13]:

$$K(E,\nu) \cdot d(E,\nu) = f, \tag{8}$$

where $K(E,\nu)$ is the stiffness matrix,

$$d(E,\nu) = \Big(x_1(E,\nu), y_1(E,\nu), \ldots, x_n(E,\nu), y_n(E,\nu)\Big)^T \tag{9}$$

the displacement vector with x- and y-displacement for each node and f the force vector. K and d are depending on the elastic modulus and the Poisson's ratio. We can take the elastic modulus out of the matrix K and write equation (8) as follows:

$$E \cdot K(1,\nu) \cdot d(E,\nu) = f. \tag{10}$$

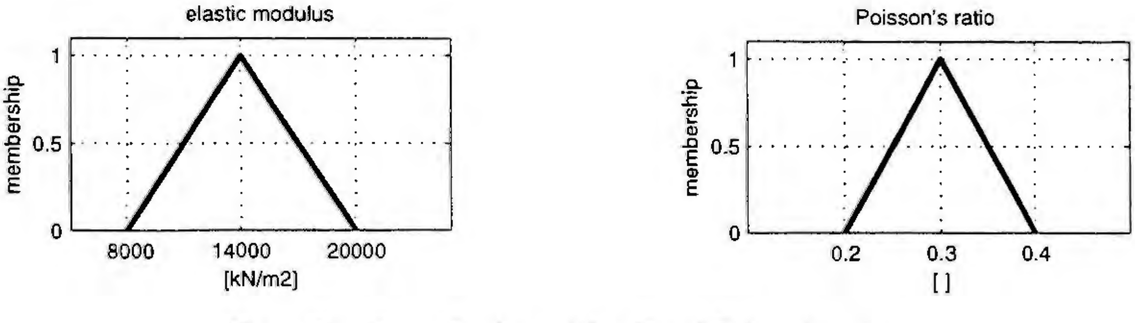


Figure 2: fuzzy elastic modulus E and Poisson's ratio ν

If $d(1,\nu)$ is a solution of $K(1,\nu) \cdot d(1,\nu) = f$, we have

$$d(E,\nu) = d(1,\nu)/E. \tag{11}$$

We have to solve equation (8) for fuzzy parameters. For that purpose equation (11) is evaluated for fuzzy $\tilde{E}$ and $\tilde{\nu}$ taking the α-cuts $[\tilde{E}]_\alpha$ and $[\tilde{\nu}]_\alpha$. Because the function $d(E,\nu)$ is continous for the values of E and ν in question we can write:

$$[d(\tilde{E},\tilde{\nu})]_\alpha = d([\tilde{E}]_\alpha,[\tilde{\nu}]_\alpha) = d(1,[\tilde{\nu}]_\alpha)/[\tilde{E}]_\alpha. \tag{12}$$

Visualizing the solution

The result $[d(\tilde{E},\tilde{\nu})]_\alpha$ is a surface in $\mathbb{R}^n$ parametrized by $E \in [\tilde{E}]_\alpha$ and $\nu \in [\tilde{\nu}]_\alpha$, which cannot be visualized as a whole in one plot, as in the crisp case. So we have to restrict our view to single nodes and in general to single displacement directions. We define $[\tilde{x}_i]_\alpha^L = \min\{x_i(1,\nu) : \nu \in [\tilde{\nu}]_\alpha\}$ and $[\tilde{x}_i]_\alpha^R = \max\{x_i(1,\nu) : \nu \in [\tilde{\nu}]_\alpha\}$. For the displacement in x-direction of node i we get from (12):

$$[x_i(\tilde{E},\tilde{\nu})]_\alpha = x_i(1,[\tilde{\nu}]_\alpha)/[\tilde{E}]_\alpha = \left[[\tilde{x}_i]_\alpha^L,[\tilde{x}_i]_\alpha^R\right]/[\tilde{E}]_\alpha = \left[[\tilde{x}_i]_\alpha^L/[\tilde{E}]_\alpha^R,[\tilde{x}_i]_\alpha^R/[\tilde{E}]_\alpha^L\right], \tag{13}$$

cf. [11] for interval division.

All these intervals, computed for x- and y-directions, are the components of the hull of $[d(\tilde{E},\tilde{\nu})]_\alpha$, cf. (6). These components are not comparable, because we have neglected the interactivity in this process. Then the membership function of $x_i(\tilde{E},\tilde{\nu})$ is put together from the α-cuts.

However in this simple case we can visualize the displacement of a node i in two dimensions accurately by

$$\left[\begin{pmatrix} x_i(\tilde{E},\tilde{\nu}) \\ y_i(\tilde{E},\tilde{\nu}) \end{pmatrix}\right]_\alpha = \left\{\frac{1}{E} \cdot \begin{pmatrix} x_i(1,\nu) \\ y_i(1,\nu) \end{pmatrix} : E \in [\tilde{E}]_\alpha, \nu \in [\tilde{\nu}]_\alpha\right\}. \tag{14}$$

$[(x_i(1,\tilde{\nu}),y_i(1,\tilde{\nu}))^T]_\alpha$ is a curve in $\mathbb{R}^2$ parametrized by $\nu \in [\tilde{\nu}]_\alpha$. Multiplying the curve by $1/[\tilde{E}]_\alpha$ is a homothetic transformation of the curve parametrized by $E \in [\tilde{E}]_\alpha$.

In figure 4 the displacements of four nodes, which are marked by bullets (cf. figure 1) and numbered from left to right, are drawn in the two ways discussed above. In the latter version the degree of membership is indicated by a gray-scale. White means 0, light gray low and dark gray high degree of membership and at last black means 1.

Some values $f(x,y)$ on the cross-section, e.g. the tension, are visualized using contour plots. Areas such as $A = \{(x,y) : c_1 \leq f(x,y) \leq c_2\}$ are often coloured using the colours of the rainbow for indicating lower or higher values of f. Let $\tilde{f}(x,y)$ be the fuzzy value at an arbitrary point (x,y) on the cross-section provided by an interpolation using the values at the nodes or Gauss points. We define the degree of membership of the fuzzy value $\tilde{f}(x,y)$ to the crisp interval $C = [c_1,c_2]$ by

$$m_C(\tilde{f}(x,y)) = \sup_{a \in C} m_{\tilde{f}(x,y)}(a), \tag{15}$$

cf. figure 3 where $m_C(\tilde{f}(x,y)) = 0.5$.

So we get a fuzzy area which indicates the membership of $\tilde{f}(x,y)$ to the interval C at each point (x,y). This is a fuzzy extension of an often used visualizing concept.

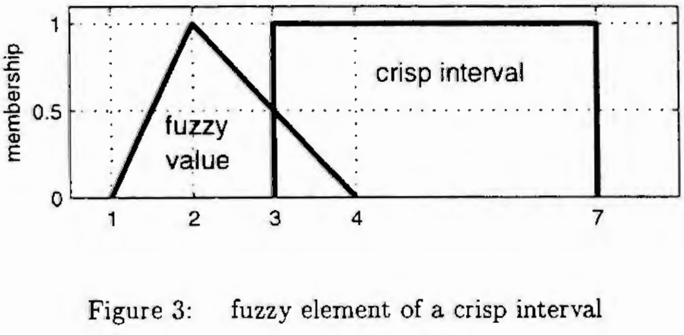


Figure 3: fuzzy element of a crisp interval

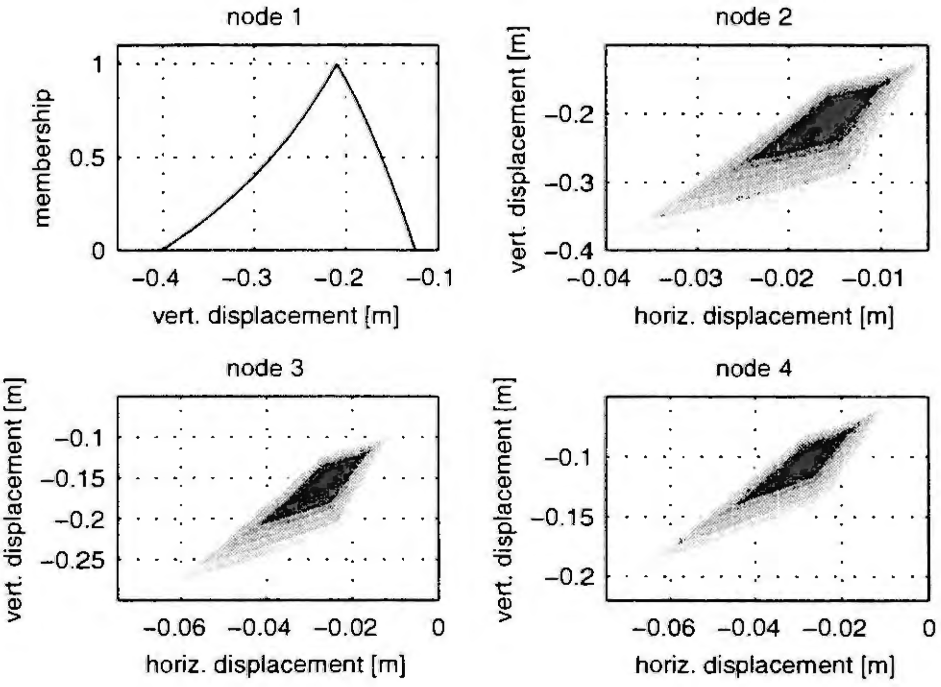


Figure 4: fuzzy displacements

Algorithm

In general the function $x_i(1,\nu)$ (resp. $y_i(1,\nu)$) is not monotonic. We have to calculate the set theoretic image $x_i(1,[\tilde{\nu}]_\alpha)$ by solving a global optimization problem. This has to be done for each node, for each displacement direction and also for each α-cut.

To reduce such a big computational effort we compute $d(1,\nu)$ for $\nu \in P = \{0, h, 2h, \dots, 0.5-h\}$ (valid values of ν are in $[0, 0.5)$) with h about 0.01 up to 0.05. In other words the finite element problem is solved for $E = 1$ and $\nu \in P$. For this purpose any finite element software package can be used.

So we get an approximation $\hat{d}(1,\nu)$ of $d(1,\nu)$ using linear interpolation. The components of $\hat{d}(1,\nu)$ are $\hat{x}_i(1,\nu)$ and $\hat{y}_i(1,\nu)$. Let $[\tilde{\nu}]_\alpha$ be an α-cut of $\tilde{\nu}$ and S the discrete set

$$S = \{x_i(1,\nu) : \nu \in [\tilde{\nu}]_\alpha \cap P\} \cup \{\hat{x}_i(1,[\tilde{\nu}]^L_\alpha), \hat{x}_i(1,[\tilde{\nu}]^R_\alpha)\} \tag{16}$$

Then $[\tilde{x}_i]^L_\alpha$ is approximately equal to the smallest element of S and $[\tilde{x}_i]^R_\alpha$ to the biggest element of S.

For calculating $[x_i(\tilde{E},\tilde{\nu})]_\alpha$ we proceed as in (13) and for the visualizing of the two-dimensional displacements we also use the already computed solutions and an appropriate visualization program. It is obvious, that this method works well only if the number of fuzzy input parameters is small.

Conclusion

The finite element method with fuzzy parameters is in any case an improvement of the current state in modelling vagueness in soil mechanics. Research concerning this method is only at an initial state. Further research has to be divided into a mathematical branch and into one involving the cooperation with engineers in geotechnics and strength of materials.

The goals of mathematical research are:

1. The minimization of the high computional effort.
2. The finding of 'self-validating' methods to improve the reliability of solutions which are currently only approximative. The theory of self-validating methods for function space problems described

in [8] seems to be a reasonable way for solving parameter depending equations. The results of such a computation are (interval) polynomials in the parameters. These polynomials are used for further calculation with fuzzy or interval numbers, for which already some methods exist in [6, 11].

The goals of research in cooperation with civil engineers are:

1. The application of the theory to more realistic soil mechanical models, which is already in progress for models developed by R. Stark [14, 15].
2. The transformation of linguistic expressions describing soil properties into fuzzy soil parameters used in the applied model, possibly accumulated in data banks.
3. Finding appropriate visualizations of the fuzzy solution, which should be as convenient and affirmative as possible for the civil engineer , because here the visualization of the solution is quite different from what an engineer nowadays gets from an output of a finite element package.

References

1. Bandemer, H. and Gottwald, S., Einführung in Fuzzy-Methoden, Berlin: Akademie Verlag 1989
2. Caligiana, G., Fuzzy logic in engineering applications, Österr. Ing.- und Arch.-Zeitschr. 9 (1995)
3. Dubois, D. and Prade, H., Possibility Theory, Plenum Press, New York, 1988
4. Dubios, D. and Prade, H., Fuzzy Sets and Systems, Theory and Applications, Academic Press, San Diego, 1980
5. Fetz, Th., Hofmeister, M., Hunger, G., Jäger, J., Lessman, H., Oberguggenberger, M., Rieser, A. and Stark, R., Tunnelberechnung – fuzzy?, Bauingenieur, to appear
6. Hammer, R., Hocks, M., Kulisch, U. and Ratz, D., C++ Toolbox for Verified Computing, Springer, Berlin, 1995
7. Hughes, Th. J. R., The Finite Element Method, Prentice-Hall, New Jersey, 1987
8. Kaucher, W. E. and Miranker, W. L., Self-Validating Numerics for Function Space Problems, Academic Press, Orlando, 1984
9. Kleiber, M. and Hien, T. D., The Stochastic Finite Element Method, Wiley, Chichester, 1992
10. Lessmann, H., Mühlögger, J. and Oberguggenberger, M., Netzplantechnik mit unscharfen Methoden. Bauingenieur 69 (1994), 469-478
11. Neumaier, A., Interval Methods for Systems of Equations, Cambridge University Press, Cambridge, 1990
12. Schwarz, H. R., Methode der finiten Elemente, Teubner, Stuttgart, 1991
13. Smith, I. M. and Griffiths, D. V., Programming the Finite Element Method, Wiley, Chichester, 1988
14. Stark, R. F., Boden-Bauwerk Interaktion bei inhomogenem Boden und nichtlinearem Baugrundverhalten, Projektbericht J0963-TEC, Innsbruck, 1995
15. Stark, R. F. and Booker, J. R., Surface Displacements of a Nonhomogeneous Elastic Half-Space Subjected to Uniform Surface Tractions, Int. J. for Numer. and Analyt. Meth. in Geomech., to appear
16. Valliappan, S. and Pham, T. D., Fuzzy finite element analysis of a foundation on an elastic soil medium, Int. J. for Numer. and Analyt. Meth. in Geomech., 17 (1993), 771-789
17. Valliappan, S. and Pham, T. D., Elasto-plastic finite element analysis with fuzzy parameters, Int. J. for Numer. and Analyt. Meth. in Geomech., 38 (1995), 531-548
18. Zadeh, L. A., Fuzzy Sets, Information and Control, 8 (1965), 338-353

FUZZY DECISIONS IN DISCRETE EVENT SIMULATION

M. Lingl[1], **F. Breitenecker**[2]
Dept. Simulation Techniques
Technical University Vienna
Wiedner Hauptstraße 8-10, A-1040 Wien
[1] mlingl@osiris.tuwien.ac.at, [2] Felix.Breitenecker@tuwien.ac.at

Introduction

Since its development in 1965, the fuzzy theory has become a well accepted tool of engineering, especially fuzzy controllers are widely spread. But there is more potential in the theory of fuzzy sets and of fuzzy logic. This paper wants to point out how the fuzzy theory can be used to make decisions in discrete modelling and simulation.

Decisions in discrete event simulation

Whatever level of programming the user chooses in modelling of discrete systems (e.g. event lists, process oriented, object oriented), in every model there comes a point when decisions have to be made how to proceed. In other words, there is a ramification where the flow of the system can proceed onto several different ways.
Traditionally, there are three possibilities to make those decisions:

- deterministic (predefined)
- algorithmic (calculated)
- stochastic (by accident)

Fuzzy logic can be used to implement a fourth way to make decisions. It can be considered either an extension of the algorithmic method or a mixture between the algorithmic and the stochastic method, depending on whether random numbers are used to interprete the resulting fuzzy values or not.

Reasons to use fuzzy logic

There are several reasons to use fuzzy logic in discrete simulation, and they are very much the same as the reasons to use fuzzy logic in controllers:

- **fuzzy data:** Data collection often is a difficult as well as an important task to be done for a simulation. Unfortunately, most data are inaccurate. Using fuzzy logic is one way to handle these inaccuracies.
- **fuzzy rules:** The information about the system to be modelled provided by other people is often inaccurate and biased as well. The way rules are formulated in fuzzy logic makes it easier to bring lingual information into a form readable by the computer.
- **clear parameters:** Even people who are not inflicted in computer science, mathematics, or simulation may understand the meaning of fuzzy sets and fuzzy rules if they are interested. This makes it easier for the simulation specialist to work together with other scientists in adjusting the parameters.
- **standardization of inaccuracy:** Many ways to handle inaccuracies of data and rules can be imagined. But if simulationists agree on fuzzy logic as a standard tool, programs and models can be much easier handed over to somebody else. Standardization simply grants better readability.

Fuzzy logic in control engineering

A controller can be seen as an *n*-dimensional function of an *m*-dimensional parameter, which is the same as a controller with *m* inputs and *n* outputs. By describing the function with the means of fuzzy logic we get a fuzzy controller, where the inputs and the outputs are linked by fuzzy rules. We name the inputs in*i*, the outputs out*i*, and the fuzzy sets set*i* to describe how a fuzzy controller works. Later we will refer to this description in order to point out the differencies between using fuzzy logic in continuous or in discrete systems.
Some typical fuzzy rules:

- out1:=(set1), (weight), if: in1==(set2) AND in2==(set3);
- out1:=(set4), (weight), if: in1==(set3) AND in3==(set5);
- out2:=(set1), (weight), if: in2==(set3) AND in3==(set2);

These rules are worked out by the computer in four steps:

- evaluating the if-conditions with the AND-concurrences and the weight factor

- OR-concurrence of the output fuzzy sets
- defuzzyfication
- setting the actual output values

Evaluating the if-conditions just means to look up the fuzzy value for a given input value in the correct table function. The fuzzy values are concurred by a fuzzy AND operator. The result is then multiplied with the weight factor, which describes the importance of the rule.

The output fuzzy sets are weighed with the corresponding results of the if-conditions (i.e. they are multiplied), and sets that are assigned to the same output are concurred by a fuzzy or operator.

To get a value that can be assigned to an output, the resulting fuzzy sets must be defuzzyfied. Two methods are commonly used:

- **maximum:** Problems may occur when there are more than one abscissa values giving the same maximum.
- **centre of gravity:** This method works only with finite fuzzy sets.

Fuzzy logic in discrete systems

The idea is to apply the fuzzy theory to any logical statement where the data are inaccurate. In the case of a discrete decision we have an if-then-statement, which is different to the rules of a fuzzy controller in the following points:

- The if-part of the statement may contain any fuzzy expression (not only AND-concurrences), i.e. AND, OR, NOT, or any other general fuzzy operator.
- Output values are assigned a discrete value, not a fuzzy set.
- Normally there is no natural order of the output values, although they may be represented by real numbers.

Hence, new methods of defuzzyfication have to be found. To explain these new methods in detail, let us consider a case where the path of a discrete model is split into four paths. At this ramification a decision has to be made. It is assumed that it is possible to choose one path, more than one, or none at all. Normally, we would name the four paths 1, 2, 3, and 4, but in order to emphasize the fact that they are not ordered, we call them A, B, C, and D. The rules we know must be written down in the form

- if (fuzzy condition a) then path:=A
- if (fuzzy condition b) then path:=B
- if (fuzzy condition c) then path:=C
- if (fuzzy condition d) then path:=D

Defuzzyfication in discrete systems

Each fuzzy condition gives a value in [0,1]. These values are stored, and then we have several possibilities to calculate the actual path to be chosen:

- Compare each of the values with a certain boundary value and choose the corresponding path if the fuzzy value is greater than the boundary value. This method may return any number of paths from 0 to 4.
- Choose one path by taking a random number, letting the fuzzy values be the probabilities for each path. Thus we will always get one path.
- Take the path with the greatest fuzzy value. As the maximum may occur more often than once. We may get one or more paths.
- If, and only if, the paths have a natural order, and we have a mathematical reason to assign the value 1 (or any other value) to the first path, the value 2 to the second path, and so on, then we may also want to use an analogon to the centre-of-gravity-method. We multiply the identification number of each path with its fuzzy value, sum them up, devide by the number of paths, and round the result.

Implementation in a discrete simulator

The discrete simulator Micro Saint was chosen for a case study. User defined functions were written in order to evaluate fuzzy expressions. Any fuzzy expressions can be used, even recursively, but they have to be encoded in an array variable. This is shown in figure 1, which not only gives technical information for encoding the expressions, but moreover shows the structure of a fuzzy expression, which is very much the same as the structure of any logical term. Especially important is that a fuzzy value may contain other fuzzy values as input values for a fuzzy operation.

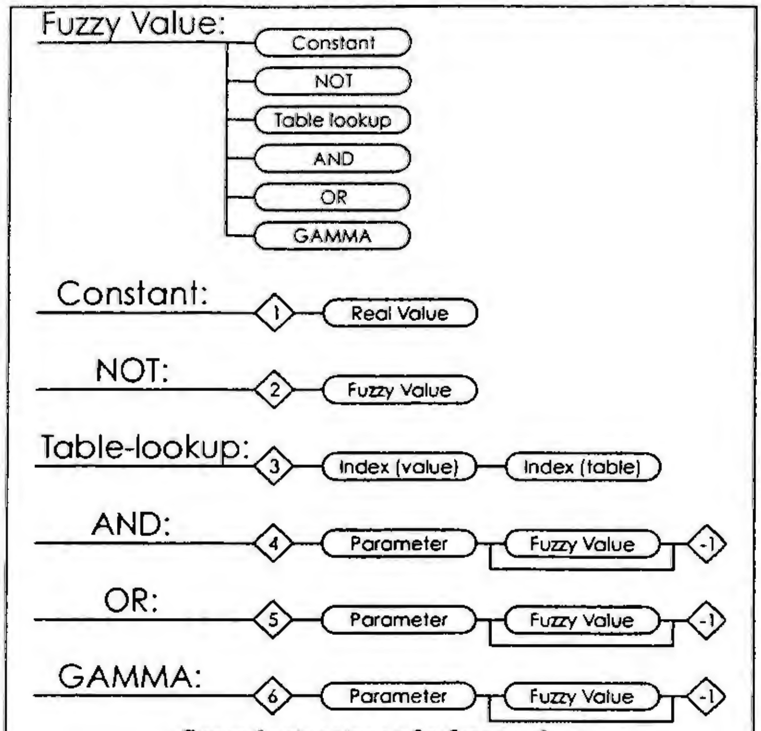


figure 1: structure of a fuzzy value

1. Constants might be useful for testing purposes. They also offer the possibility to "cut" a fuzzy set by using AND or OR.
2. NOT amounts to the fuzzy complement of the fuzzy value specified, i.e. 1-(fuzzy value).
3. Table lookup takes a value (specified by the first parameter) and returns the according fuzzy value of a certain fuzzy set, which is defined by a table function (specified by the second parameter).
4. AND is the fuzzy-AND operator and works with any number of arguments. The AND statement is closed by an element containing the value -1.
5. OR operates like AND, but for fuzzy-OR.
6. GAMMA operates like AND, but for fuzzy-GAMMA.

The case study

The purpose of the case study was to test whether fuzzy logic can really be used to make decisions in discrete event simulation. Therefore an artificial model was created. It consists of an entrance that generates entities, ten tasks that process the entities, and an exit that protocols data. Although it is an artificial model it could be considered to represent some imaginable post office or bank or something like that.

Entities arrive only one at a time and with exponentially distributed interarrival times with a mean value of 1. They are then led to ten counters which have an exponentially distributed working time with a mean of 10. Thus the capacitiy of the counters is fitted exactly for the arriving entities. Each counter can process only one entity at a time.

What changed in the system was the way the entities behaved in selecting a counter and in waiting if the counter was not free:

System 1: Each counter has a queue for itself. New entities enter the model in task 111 („Decide"). After this task, the entity is sent to one of the counters by a probabilistic decision. All counters have the same probability. At the counter the entity waits in the queue until the counter is free.

System 2: The second way to model this system is to build just one queue for all the counters together. This would be the best solution in the sense that there will be the shortest waiting times, but the problem is that human beings simply do not behave that way.

These two ways to model queues are quite sufficient for most problems. Nevertheless they cannot represent the whole truth. It makes a difference whether there are just workpieces waiting to be processed or there are human beings. In an assembly line it is not important whether there are pieces waiting for a long time, as

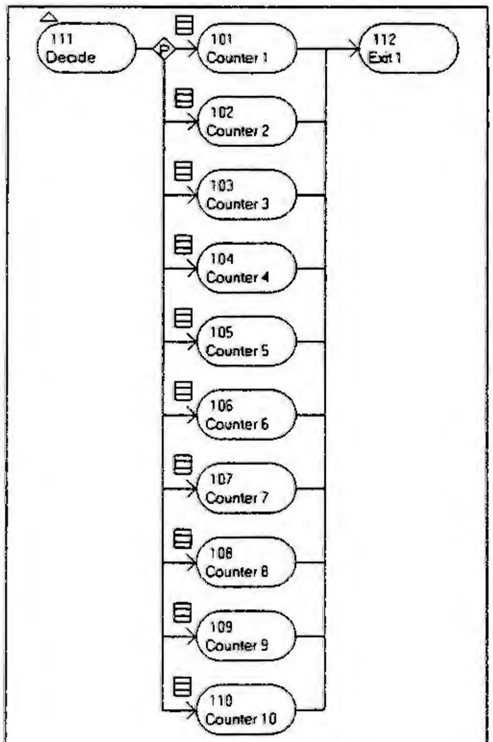


figure 2: sample model for the case study

long as the utilization rate of the whole system is good. Human beings will not want to wait any longer than absolutely necessary. They do not care about the costs of a counter and whether it is utilized well or not. They want to be served as quickly as possible. In order to simulate the system more accurately, this behaviour has to be taken into consideration. An easy way to do so was chosen in the systems 3 and 4.

System 3: Here the entity comes into the „Decide"-task, where the shortest queue is selected. The tactical decision then sends the entity to the first counter of those with the shortest queue (if more than one).

System 4: It works very similar to system 3, but here the counter is chosen randomly if there are more than one with the shortest queue.

In the systems 3 and 4 individual behaviour is considered slightly, but an automatic control system of an assembly line could still do the same.

System 5: Here fuzzy logic is used to gain a probability for each counter. These probabilities are calculated in the „Decide"-task. The following probabilistic decision then sends the entity to a counter. Note that the counter is chosen randomly, but the probabilities for the choice were given by a fuzzy expression.

System 6: The sixth system is an extension of system 5. Here the entities can change between the queues while they are waiting. Every time an entity leaves the system, all remaining entities check whether they will stay in the queue or change to another queue. They use fuzzy logic to judge the situation.

In order to introduce another probabilistic element into the model, the counters were given different working speeds. Such a situation may occur in technical systems as well, but it is typical for human behaviour. Not all clerks in a post office have the same working speed. It depends on their age, their experience, their physical and psychical condition, and maybe some more things.

Results of the case study

Expectations were fulfilled in the sense that the waiting times in the fuzzy models ranged between those of the worst model (longest waiting times, system 1) and the best model (shortest waiting times, system 2). Careful examination of the result files showed that not only the statistical results were correct, but also the behaviour of any single entity was reasonable. In short: Fuzzy logic in discrete simulation works!

A single run with 10000 entities in each system produced the following results:

	System 1	System 2	System 3	System 4	System 5	System 6
waiting 1)	9334	8501	8003	7978	8118	8944
waittime 2)	366.12	20.62	26.37	26.77	47.6	51.09
waiting 2)3)	312.14	20.87	25.61	25.87	41.18	49.24

1) absolute overall number for the whole simulation run

2) average values, excluding zeros

3) at the moment; A snapshot was done every time an entity left the system.

References:

[1] Zadeh L.A.: Outline of a new approach on the analysis of complex system and decision processes
IEEE Trans. syst. Man., Cybern., 1973

[2] Breitenecker F.: Diskrete Simulationssysteme
Script, TU Vienna 1991

[3] Tilli T.: Fuzzy-Logik, Grundlagen, Anwendungen, Hard- und Software
Franzis-Verlag, Munich 1993

[4] Tilli T.: Automatisieren mit Fuzzy-Logik
Franzis-Verlag, Munich 1992

[5] Tilli T.: Mustererkennung mit Fuzzy-Logik
Franzis-Verlag, Munich 1993

[6] Möller D.P.F.: Fuzzy Systems in Modelling and Simulation
Proc. EUROSIM '95, pp.65-74, Elsevier Science, Amsterdam 1995

[7] Zadeh L.A.: Fuzzy Sets
Informat. Control, Vol. 8, pp. 338-353, 1965

[8] Wang L.X.: Adaptive Fuzzy Systems and Control
Prentice Hall International, 1994

Advanced Neural Network Training for Discrete Time Nonlinear Dynamic System Modeling

Jennie Si and Guian Zhou
Department of Electrical Engineering
Arizona State University
Tempe, AZ 85287-5706

Abstract. In the present paper we address the problem of discrete time nonlinear dynamic system modeling using feedforward neural networks and advanced training methods for obtaining good approximation models. Most of neural network applications rely on the fundamental approximation property of feedforward networks. Supervised learning is a means of implementing this approximate mapping. In a realistic problem setting, a mechanism is needed to devise this learning process based on available data, starting from choosing an appropriate set of parameters in order to avoid overfitting, to an efficient learning algorithm measured by computation and memory complexities, as well as the accuracy of the training procedure measured by the training error, and not to forget testing and cross-validation for generalization. A comprehensive supervised learning mechanism is introduced in this paper which addresses the issues raised above. We use several simulations to demonstrate the applicability of supervised learning and feedforward networks in various problem settings and to evaluate the effectiveness of the various algorithms.

Introduction

Nonlinear dynamic systems often arise in engineering applications. As is well known, the analysis of these systems can be complicated. For this reason it is frequently desirable to have available simple approximate models that can be used for synthesis, analysis, and identification. In the present paper, we consider feedforward networks as simple approximate models to these nonlinear dynamic systems.

The feedforward network $g(\cdot)$ which implements the following system (1) is the focus of the present discussion.

$$\hat{y}(k+1) = g[y(k), \cdots, y(k-n_y), x(k+1), \cdots, x(k-n_x)] \tag{1}$$

In the above network, the inputs are $x(k+1), \cdots, x(k-n_x), y(k), \cdots, y(k-n_y)$, and the network output is $\hat{y}(k+1)$. Common examples of feedforward neural network implementations include multi-layer perceptrons and radial basis function networks, as well as others [11]. They both have been proved to be universal approximators [7] [10] [12] [14]. Other than the apparent global vs. local tuning characteristics for sigmoid and radial basis networks, respectively, and the faster learning capability by radial basis networks, another interesting aspect regarding the two networks is that when the input dimension is low, radial basis function networks tend to be more efficient in terms of network complexity than sigmoid. However, when the input dimension is high, the above observation is reversed ([2], chapter 5 in [1]). Therefore a careful selection of the centers and widths of the radial basis functions become critical which is in general a nontrivial task.

Multi-layer feedforward networks have been applied successfully to solve some difficult and diverse problems, including nonlinear system identification and control, financial market analysis, signal modeling, power load forecasting, etc., by training them in a supervised manner [1] [13] [16] [11]. In supervised learning, we start with a training set and use certain numerical procedures, which usually solve the nonlinear optimization problem deduced by supervised learning, to compute the network parameters (weights) by loading training data pairs into the network. The hope is that the network so designed will generalize, meaning that the input-output relationship computed by the network is within certain expectation as measured by testing error, for some input output data pairs which were never used in training the network. A typical bad generalization may occur due to overfitting. Generalization is generally affected by three factors, the size and efficiency of the training data set, the complexity of the network, and the complexity of the physical problem at hand. In most neural network application problems, the size of the training set is given, and herein is assumed to be representative of the system. The issue to be addressed then is how to determine a set of network parameters for achieving good generalization. This may include choosing the right network size, an efficient algorithm of determining the weights for the network to achieve desired accuracy for training and testing.

In the following, we first address the issue of existence of feedforward neural networks as approximate models to nonlinear dynamic systems. In particular, we consider feedforward neural networks with input-output representations as approximations to a class of discrete time nonlinear dynamic systems. The inputs to the feedforward neural network model are composed of the inputs to the plant and tapped delays from previous outputs. Then we devise a comprehensive procedure to construct such a feedforward neural network model from input-output data measurements. The training mechanism discussed in this paper are applicable to either static or dynamic, sigmoid or radial basis networks, as will be shown in the following sections.

Existence of a Feedforward Network Model

We will be concerned with discrete-time systems which can be represented by input-output difference equations. They take the form

$$y(k+1) = \Phi[y(k), y(k-1), \cdots, y(k-m), u(k), u(k-1), \cdots, u(t-n)] \tag{2}$$

where $u(\cdot)$, and $y(\cdot)$ are discrete time sequences.

When Φ in (2) is unknown, the problem of identification arises. The problem is described in the following. The input and output of a time-invariant, causal discrete-time dynamic plant are $u(\cdot)$ and $y(\cdot)$, respectively, where $u(\cdot)$ is a uniformly bounded function of time. The plant is assumed to be stable with a known parameterization but with unknown values of the parameters. The objective is to construct a suitable identification model which when subjected to the same input $u(k)$, produces an output $\hat{y}(k)$ which approximates $y(k)$ in the sense that $\| \hat{y}(k) - y(k) \| < \varepsilon$, for $k \in K$, K is a finite interval of interest, and for some desired $\varepsilon > 0$ and a suitably defined norm.

In order to represent a general nonlinear plant of the form (2), we need to use a class of models that can represent the system function Φ. It is a common knowledge that the continuous mapping Φ can be approximated arbitrarily well by various regular simple structures, e.g., sigmoid networks or radial basis networks, *etc.*. If the neural model inputs and outputs are arranged as in (2), the above description can be summarized as:

For a nonlinear, time invariant, causal, discrete-time system of the form (2), there exists a feedforward neural network $\hat{N}$ (with parameters w) such that

$$\| \hat{y}(k) - y(k) \| < \varepsilon,$$

for $k \in K$, and for some desired $\varepsilon > 0$, where

$$\hat{y}(k+1) = \hat{N}[y(k), y(k-1), \cdots, y(k-m), u(k), u(k-1), \cdots, u(t-n)] \tag{3}$$

and $y(\cdot)$ is from (2).

This result can be justified by the universal approximation property of a feedforward neural networks, either a sigmoid or a radial basis [11].

Neural Network Model Contrstruction from Data Measurements

Training a neural network is an optimization problem. In particular, the objective of training a neural network is to associate input-output training pairs $\{(x^1, t^1), \cdots, (x^\xi, t^\xi), \cdots, (x^p, t^p)\}$ by properly adjusting the weights $\mathbf{w} \in R^q$ in the network such that an error measure $E(\mathbf{w})$ is minimized. A sum of squared error function is a commonly used measure.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{\xi,i}(e_i{}^\xi)^2 = \frac{1}{2}\sum_{\xi,i}(t_i{}^\xi - o_i{}^\xi)^2, \tag{4}$$

where o^ξ represents the network output when the input is x^ξ.

Various training algorithms have been developed to adapt the weights $\mathbf{w}$ in the network recursively to reduce the error defined in (4). Two major types of algorithms have been widely used. On one hand there are several *ad hoc* techniques, *e.g.*, varying the learning rates in the backpropagation algorithm, adding a momentum term to the basic backpropagation algorithm [15]. Another school uses standard numerical optimization techniques like conjugate gradient method or quasi-Newton (secant) methods [8]. Some nonlinear least squares methods have been tested and practically provide better performance than the previous two types of approaches [4].

Back-propagation is a commonly used training technique where parameters are moved in the opposite direction to the error gradient. Each step down the gradient results in smaller error until an error minimum is reached. The computational complexity of back-propagation is contributed mostly by calculating each of the partial derivatives which is only of $O(n)$. However this gradient descent based method is only linearly convergent and it usually is very slow. As another heuristic technique of similar complexity to the backpropagation algorithm is the gradient algorithm with momentum. It makes changes proportional to a running average of the gradient. It may be considered as an approximation of conjugate gradient method. In general, this technique can decrease the probability that the network gets stuck in a shallow minimum on the error surface and thus decrease training times. However the learning rate and the momentum parameter have to be carefully chosen which is a highly heuristic process.

Newton's method provides good local convergence properties. It is q-quadratically convergent when a minimum is approached. Modifications are needed before it can be used at points that are remote from the solution. Calculation of the Hessian is desired to be avoided. The Levenberg-Marquardt algorithm is a combination of gradient descent and Gauss-Newton methods which has the advantages of the local convergence properties of the Gauss-Newton method and the global properties of gradient descent. A comparison study is reported in [4] where the Levenberg-Marquardt method significantly outperforms the conjugate gradient and the back-propagation with variable learning rate [6] in terms of training time and accuracy.

The systematic feedforward neural network training approach introduced in this paper is based on the observation of Jacobian rank deficiency. It was noted in [3] that the Jacobian matrix in the Gauss-Newton algorithm is commonly rank deficient and the degree of rank deficiency is usually very high for neural network problems because of saturation characteristics of node sigmoid functions, linear dependencies among node outputs, linear dependencies among local gradients, *etc.* Experiments have revealed that the rank of a deficient Jacobian matrix is about 60 ~ 80% of the size of the Jacobian on average, and it may reach as low as 20%. Rank deficient Jacobian matrices on one hand result in the Gauss-Newton and certain high order algorithms not applicable, and on the other hand indicate that some weights in the network are redundant. Some modifications, for example, the Levenberg-Marquardt algorithm, are to make the cross-product matrices of the Jacobian matrix positive definite. Although satisfactory convergence properties are obtained [4], the weight redundancy is ignored.

In the present paper, we only update the network weights which correspond to a more "efficient" Jacobian matrix (less rank deficient) [17]. This on one hand overcomes overfitting introduced by a highly complex network and on the other hand reduces computation and memory complexities in network training. The algorithm presented in this paper is derived from the format of the Gauss-Newton method. It has similar convergence properties to the Levenberg-Marquardt algorithm. The details of the derivations of the training algorithm is given in the next subsection.

Derivation of the Systematic Algorithms

When the Gauss-Newton update rule is employed, the weight change $\Delta \mathbf{w}^{(k)}$ from $\mathbf{w}^{(k)}$ is computed by

$$(J^T J)\Delta \mathbf{w}^{(k)} = -J^T \epsilon \tag{5}$$

where $J \in \mathcal{R}^{p \times q}$ is the Jacobian matrix at the k^{th} iteration, ϵ is the k^{th} error vector. Obviously, when J is rank deficient, which is often the case for neural network problems, $J^T J$ will not be invertible and (5) can not be applied directly.

To solve for $\Delta \mathbf{w}^{(k)}$, we make a diagonal pivoting triangular factorization (chapter 5 in [9]) on $J^T J$, that is, $(JP)^T(JP) = L^T DL$, and thereby obtain

$$L^T DL\Delta \tilde{\mathbf{w}}^{(k)} = -(JP)^T \epsilon \tag{6}$$

where $P \in \mathcal{R}^{q \times q}$ is a pivoting matrix, $\Delta \tilde{\mathbf{w}}^{(k)} = P^{-1}\Delta \mathbf{w}^{(k)}$, $L \in \mathcal{R}^{q \times q}$ is an upper triangular matrix with unit diagonal elements, $D \in \mathcal{R}^{q \times q}$ is a diagonal matrix of the form

$$D = diag(d_1, d_2, \cdots, d_r, 0, \cdots, 0), \quad (d_i \geq d_j, \quad \text{for} \quad i < j)$$

where r is the rank of J.

Let $\mathbf{b} = (L^T)^{-1}(JP)^T\epsilon$, we have

$$DL\Delta\tilde{\mathbf{w}}^{(k)} = -\mathbf{b}. \tag{7}$$

Since D is not full rank, we represent D, L, $\Delta\tilde{\mathbf{w}}^{(k)}$ and $\mathbf{b}$ in appropriate block matrix forms as

$$D = \begin{bmatrix} D_1 & \\ & 0 \end{bmatrix}, \quad L = \begin{bmatrix} L_1 & L_2 \\ & L_3 \end{bmatrix}, \quad \Delta\tilde{\mathbf{w}}^{(k)} = \begin{bmatrix} \Delta\tilde{\mathbf{w}}_1^{(k)} \\ \Delta\tilde{\mathbf{w}}_2^{(k)} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \tag{8}$$

where $D_1 \in \mathcal{R}^{r\times r}$ is a non-zero diagonal matrix, $L_1 \in \mathcal{R}^{r\times r}$ is an upper triangular matrix, and other sub-matrices and sub-vectors have proper dimensions. Substituting the block matrices in (8) into (7), one can verify that $\mathbf{b}_2 = 0$.

Equation (7) then becomes

$$D_1(L_1\Delta\tilde{\mathbf{w}}_1^{(k)} + L_2\Delta\tilde{\mathbf{w}}_2^{(k)}) = -\mathbf{b}_1. \tag{9}$$

The above derivation, which is in the spirit of the Gauss-Newton algorithm under Jacobian rank deficiency, divides the network weights into two subsets: $\Delta\tilde{\mathbf{w}}_1^{(k)}$ and $\Delta\tilde{\mathbf{w}}_2^{(k)}$. The former is a principle component subset extracted from the entire weight set and therefore has major influence on the network at the k^{th} iteration; the later is insignificant in the sense that L_2 is usually dependent upon L_1. In the following, we provide means of updating $\Delta\tilde{\mathbf{w}}_1^{(k)}$ in the meanwhile eliminating $\Delta\tilde{\mathbf{w}}_2^{(k)}$. The details are not provided due to space limitation.

Algorithm I. Network tuning and parameter tuning (NTPT):

1. $\Delta\tilde{\mathbf{w}}_2^{(k)} = -\tilde{\Lambda}(k)\ \tilde{\mathbf{w}}_2^{(k)}$;
2. $\Delta\tilde{\mathbf{w}}_1^{(k)} = -L_1^{-1}(D_1 + \mu I_1)^{-1}(\mathbf{b}_1 + D_1L_2\Delta\tilde{\mathbf{w}}_2^{(k)})$.

Algorithm II. Parameter tuning and then network tuning (PTNT):

1. $\Delta\tilde{\mathbf{w}}_1^{(k)} = -L_1^{-1}(D_1 + \mu I_1)^{-1}\mathbf{b}_1$ (with an implicit assumption $\Delta\tilde{\mathbf{w}}_2^{(k)} = 0$ at this step);
2. $\Delta\tilde{\mathbf{w}}_2^{(k)} = -\tilde{\Lambda}(k)\ \tilde{\mathbf{w}}_2^{(k)}$.

where $\tilde{\Lambda}(k)$ is a diagonal matrix with diagonal elements $0 \le \tilde{\lambda}_i(k) \le 1$, $i = 1, \cdots, n-r$.

It can be seen that L_2 does not have to be exactly and entirely computed in *algorithm II* (PTNT) because L_2 was "swept out", which might provide certain reduction in computation and memory complexities.

Applications and Performance Evaluations

In the following, we use nonlinear system modeling examples to demonstrate the applicability of the modeling procedure obtained in the above sections, and also to evaluate quantitatively the effectiveness of various algorithms, in terms of training time, training accuracy, network complexity and generalization capability. As a basis of comparison, the back-propagation with momentum (BPM) as considered in the Neural Network Toolbox of Matlab, the weight-elimination algorithm (WE) [16], and the Levenberg-Marquardt (LM) [4] are used to train the same problems, whose updating rules are summarized in the following:

(1) BPM: $\Delta\mathbf{w}^{(k)} = \alpha\Delta\mathbf{w}^{(k-1)} - \beta\frac{\partial E(\mathbf{w}^{(k)})}{\partial\mathbf{w}^{(k)}}$;

(2) WE: $\Delta\mathbf{w}^{(k)} = \alpha\Delta\mathbf{w}^{(k-1)} - \beta\frac{\partial(E(\mathbf{w}^{(k)}) + \eta\sum_{i=1}^{n}(w_i^{(k)})^2/(1+(w_i^{(k)})^2))}{\partial\mathbf{w}^{(k)}}$;

(3) LM: $\Delta\mathbf{w}^{(k)} = -(J^TJ + \mu I)^{-1}J^T\epsilon$

where α is a momentum constant, β is an adaptive learning rate; η is a regularization parameter. All algorithms are coded in Fortran 77 and implemented on a Sun Sparc 5 computer.

Neural networks considered in this paper are two-layer feedforward type with sigmoid hidden nodes and linear output nodes. Network weights are initialized with uniformly distributed random numbers between [-0.5,0.5]. The training process is terminated if the error function in (4) is less than a tolerance $\varepsilon = 10^{-7}$ or the epochs reach 10,000 for BPM and WE or 100 for LM and PTNT.

Example 1: Modeling a third order nonlinear system. In this example we train neural networks to model a nonlinear dynamic process, the system was considered in [13]. The inputs and outputs of the system satisfy the following equation,

$$y(i+1) = \frac{y(i)y(i-1)y(i-2)u(i-1)(y(i-2)-1) + u(i)}{1 + y^2(i-1) + y^2(i-2)}.$$

The input to the network was formulated as $x = (y(i), y(i-1), y(i-2), u(i), u(i-1))^T$ and the target output $t = y(i+1)$. The training data was generated from a uniform distribution on [-1, 1]. 700 samples were used for training the network and another set of 300 samples were used for testing.

A two-layer feedforward network $\mathcal{N}_{[5\times20\times1]}$ was trained with different algorithms. The 5 network inputs consist of 2 system inputs and 3 delayed outputs of the system. 20 experimental runs were carried out by the algorithms for 20 different arbitrary initial weights. The average initial error $E(\mathbf{w}^{(0)})$ of the 20 runs was 332.0 and the training goal of the error function was 0.03.

The 20 runs all converged by LM, PTNT and PT, respectively. None of the simulations converged to the desired accuracy by using BPM. WE gave even poorer performance than BPM. Therefore its training result is taken out from the following table. Table 1 summarizes the average training and testing results over the 20 runs.

Table 1 Simulation results for a third order nonlinear system

	Epochs	Rank	Removed weights	Final error	Testing error	CPU (s)
BPM	10,000	N/A	N/A	0.0663	N/A	8,028
LM	9.4	141.0	N/A	≤ 0.03	0.00695	180
PTNT	31.1	63.8	62.3	≤ 0.03	0.00689	267
PT	15.8	67.8	N/A	≤ 0.03	0.01202	177

As can be seen from Table 1, the convergence property and training accuracy of BPM was poor. LM significantly outperformed BPM for the example in terms of training time and accuracy. But the memory complexity of LM ($O(n^2)$) is higher than that of BPM ($O(n)$) to store the lower triangular part of J^TJ for at least two operations in μ and the upper triangular factor of $(J^TJ + \mu I)$.

PT has much better convergence property than BPM. PT required fewer iterations than PTNT, but more than LM. However due to the fact that PT was operating on a lower rank matrix than LM, the computation time for PT was still lower than LM. PT has similar memory complexity to PTNT, much less than half of LM. In particular, PTNT cost more time than LM but removed about 62 weights on average, with a maximum of 79 weights removed. PT cost a little less time than LM. The memory complexity of PT or PTNT is much less than half of that of LM, i.e., less than half of (141^2 x 4) bytes in this example.

Example 2: A distillation process identification. A distillation column is the system to be considered in this example. The column separates methannol-isopropanol binary mixtures. A mathematical model of the 26-tray methannol-isopropanol column is used to generate data for training neural networks. The inputs are generated from Gaussian distributions with modulated magnitude.

When a feedforward network $\mathcal{N}_{[20\times20\times1]}$ is used as the identification model, the input was formulated as $x = (y(i), \cdots, y(i-9), u(i+1), u(i), \cdots, u(i-8))^T$, and the target output $t = y(i+1)$. 900 training patterns and another set of 500 testing patterns were used in the example.

20 experimental runs were carried out by PTNT, PT and LM, respectively, for 20 different arbitrary initial weights. BPM and WE were taken out from the comparison because of poor convergence properties. The average initial error $E(\mathbf{w}^{(0)})$ over the 20 runs was 1778.9 and the training goal was to make the error function below 0.04. The averaged results over the 20 runs are tabulated in Table 2.

Table 2 Simulation Results for a distillation process

	Epochs	Rank	Removed w.	Final error	Testing error	CPU (s)
LM	13.8	441.0	N/A	≤ 0.04	0.0467	3,610
PTNT	36.3	187.4	167.9	≤ 0.04	0.0568	3,806
PT	19.3	188.0	N/A	≤ 0.04	0.0584	1,754

Similar observations to *Example 1* still hold in *Example 2*. Although PTNT cost more time than LM but removed 168 weights in the network on average, with a maximum of 193. PT still cost much less time than LM. The memory complexity of PT or PTNT is also much less than half of (441^2 x 4) bytes used by LM.

Conclusions

This paper has focused its discussion on how to use feedforward neural networks for nonlinear dynamic system modeling. We first provide an justification on the feasibility of using feedforward neural network for this purpose. We then devised a practical procedure which construct this feedforward neural network model from input output data measurements. Unlike many results in the literature, our approach achieves multiple objectives, i.e., high training accuracy, low memory and computation costs, and finally selecting networks with right sizes to overcome overfitting, in an integrated and systematic manner.

Acknowledgement. This research is supported in part by NSF under grant ECS-9553202, by EPRI under grant RP8015-03, and by Motorola.

References

[1] White, D. A. and Sofge, D. A. (Eds.), Handbook of Intelligent Control, Neural, Fuzzy, and Adaptive Approaches. Van Nostrand Reinhold, 1992.

[2] Liu, B. and Si, J., The Best Approximation to C^2 Functions and Its Error Bounds Using Regular-Center Gaussian networks. IEEE Trans. on Neural Networks, 5 (1994), 845-847.

[3] Saarinen, S., Bramley, R.B., and Cybenko, G., The Numerical Solution of Neural Network Training Problems, CRSD Report No. 1089. Center for Supercomputing Research and Development, University of Illinois, Urbana, 1991.

[4] Hagen, M.T., and Menhaj, M.B., Training Feedforward Networks with the Marquardt Algorithm, IEEE Trans. on Neural Networks, 6 (1994), 989-993.

[5] Battiti, R., First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method, Neural Computation, 4 (1992), 141-166.

[6] Kollias, S., and Anastassiou, D., An Adaptive Least Squares Algorithm for the Effective Training of Artificial Neural Networks, IEEE Trans. on Circuits and Systems, 8 (1989), 1092-1101.

[7] Cybenko, G. Approximation by Superpositions of a Sigmoidal Function, Mathematics of Control, Signals, and Systems, 2 (1989), 303-314.

[8] Dennis, J.E., and Schnabel, R.B., Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice Hall, Englewood Cliffs, NJ, 1983.

[9] Dongarra, J.J, Bunch, J.R., Moler, C.B., and Stewart, G.W., LINPACK: Users' Guide. SIAM Philadelphia, 1979.

[10] Funahashi, K. On the Approximate Realization of Continuous Mappings by Neural Networks, Neural Networks, 2 (1989), 183-192.

[11] Haykin, S., Neural Networks: A Comprehensive Foundation. New Jersey: Macmillan Publishing Company, 1994.

[12] Hornik, K., Stinchcombe, M., and White, H. , Multi-layer Feedforward Network Are Universal Approximators, Neural Networks, 2 (1989), 359-366.

[13] Narendra, K.S., and Parthasarathy, K. , Identification and Control of Dynamical Systems Using Neural Networks, IEEE Trans. on Neural Networks, 1 (1990), 4-27.

[14] Park, J., and Sandberg, I. W., Universal Approximation Using Radial Basis Function Networks, Neural Computation, 3 (1991), 246-257.

[15] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., Learning Representations by Backpropagating Errors, Nature, 323 (1986), 533-536.

[16] Weigend, A.S., Rumelhart, D.E. , and Huberman, B.A., Generalization by Weight-elimination with Application to Forecasting. In: Advances in Neural Information Processing Systems 3, (Eds.: Lippmann, R.P., Moody, J.E., and Touretzky, D.S.) Morgan Kaufmann, San Mateo, CA, 1991, 857-882.

[17] Zhou, G., and Si, J., Improving Neural Network Training Based on Jacobian Rank Deficiency. In: Lecture Notes in Computer Science, Artificial Neural Networks-ICANN96 (Eds.: von der Malsburg, C., von Seelen, W., Vorbruggen, J. C., and Sendhoff, B.) Springer Verlag, 1996. 857-882.

UNCERTAINTIES IN MODELLING. APPLICATION TO HEAT TRANSFERT IN BUILDINGS, COUPLING RADIATIVE AND CONVECTIVE EXCHANGES.

P. Aude, G. Rusaouen , P. Depecker

Centre de Thermique de Lyon, Equipe Thermique du Bâtiment (CETHIL/TB),
INSA Bât. 307, 20 av. Albert Einstein 69621 VILLEURBANNE. France
Phone : (33) 04 72 43 84 61, Fax : (33) 04 72 43 85 22, email : aude@parga.insa-lyon.fr

Abstract

The problem developed and analyzed in this paper is the estimation of the uncertainty associated with the results obtained by numerical simulation codes of physical systems induced from input data. Implicitly posed by the authors is the delicate question concerning the use of the output obtained by calculation codes used in prediction situations, and the reliability which can be attributed to such ouput. One example of thermal behaviour of simple physical systems is treated, serving as an illustration.

Two classical methods are presented. The first is a probabilistic method, the Quasi Monte Carlo method and the other one, a determinist method, the Finite Differences Differential Analysis. These two methods are tested on a non-linear heat transfert model. This model is extracted fraction of larger model of building thermal behaviour, allowing a simplified presentation of the methods proposed.

The comparison of the two methods leads to conclusions in favour of differencial analysis, which is clearly more calculation time saving and which makes it possible to identify sensitive data with significant bearing on output uncertainty. Nevertheless, it is emphasized that for this method it is essential to enter into the calculation code formalism in order to express the partial derivatives of the transfer function. Globally, a relative superiority of the differential analysis is shown, particularly in the case of large codes where the use of Monte Carlo method would be prohibitive in calculation time.

Introduction

Let us consider a model of building in which all the output can be grouped in a vector S with components such as walls surface temperatures, air room temperatures, air velocities in rooms, air humidities, ... This vector can be expressed in form of a linear or non linear relation $\mathscr{F}$ of the model's input data and the control parameters according to the very general expression represented by relation (1). Thus :

$$S = \mathscr{F}(E, C) \tag{1}$$

where $S = \{s_1, s_2, ..., s_q\}$ is the output vector with dimension q, and $E = \{e_1, e_2, ..., e_n\}$ the input data vector with dimension n. E contains the whole thermal and geometric data of the building. For example : south area windows, east area windows, ..., insulation thermal conduction coefficients, concrete thermal conduction coefficients, material thicknesses ....

$C = \{c_1, c_2, ..., c_m\}$ is the control parameters vector, with dimension m. For example: reference temperature in rooms, maximum heating power, ... E and C constitute the data. The $\mathscr{F}$ function is a function representative of the model, known analytically, numerically or even experimentally.

The problem of estimation of the reliability of results, i.e. error analysis, consists therefore in evaluating the effects on the components of the output S of perturbations generated on the elements of E and C. The most direct method is to associate a variation interval with each component e_i and c_i :

$$e_i \in [\, e_i - \Delta e_i \,,\, e_i + \Delta e_i \,] \quad \text{and} \quad c_i \in [\, c_i - \Delta c_i \,,\, c_i + \Delta c_i \,] \ , \text{where } i = 1, ..., n \text{ or } m.$$

Likewise, resolving the problem of error analysis will consist in reaching a fluctuation interval on the output s_j :

$$s_j \in [\, s_j - \Delta s_j \,,\, s_j + \Delta s_j] \qquad , \text{where } j = 1, ..., q.$$

The variable of output s_j may be considered as a function F_j (j jth component of $\mathscr{F}$). For instance, in the elementary case of 2 dimensions , $\mathscr{F}$ is a surface. The uncertainty analysis is perform according to the domain defined by the different variation limits. It is therefore expected that the optimal combination which produces the maximal amplitude variation of the response S be located in this zone.

Determination of output uncertainty interval

Reference method : Monte Carlo method (MC)
In the Monte Carlo method, a probability density is assigned to all of the input data which may be affected with uncertainties. For each simulation carried out, a value is randomly selected for all uncertain data according to their respective probability density, and all the uncertain parameters are simultaneaously disturbed. The Monte Carlo method thus makes it possible to take full account of the various interactions taking place among all input data of the model. The simulation product is then saved, and the process is reinitialised, using a unique and different set of input data for each operation.

The total uncertainty of simulation results can be expressed by the standard deviation :

$$s_D(p_j) = [\frac{1}{(N-1)}(\sum P_{jk}{}^2 - N.(m(P_{jk}))^2)]^{\frac{1}{2}} \tag{2}$$

where N is the number of simulations and m the average of the output values and P_{jk} the probability value of the p_j parameter.

An estimation of s_D can be deduced from each simulation and the precision of this estimation can be determined by using a distribution of χ^2 to calculate an interval of reliability for s_D [1] . The accuracy of s_D only depends on the number of simulations carried out, as shown by relation (2). The main inconvenience is that the sensitivity of the predictions related to the individual variations of each parameter is not accessible, since all the input data vary simultaneaously.

Another inconvenience is the large time computation when using MC method for complex models as building thermal behaviour simulation or fluid mechanic simulation. Studying variance variation with number of computations $s_D^2 = f(N)$ in case of thermal system model, we observe number of computations must be greater than 100 for a confidence interval less than 15 % of final s_D value (i.e. for a very great number of computations). The applications studied in this paper involved about 800 computations to reach 5% for final s_D value.

Finite Differences Differential Analysis method (FDDA)
Two cases can thus be seperated :

- The function $\mathscr{F}$ has no singularity in the domain of fluctuation of the disturbed elements of the model. An analysis is done in first order of each output s_k. If the function $\mathscr{F}$ is differentiable to the point considered, we have :

$$\Delta s_k = \sum_{j=1}^{m}\left|\frac{\partial F_k}{\partial e_j}\right|\Delta e_j + \sum_{j=1}^{n}\left|\frac{\partial F_k}{\partial c_j}\right|\Delta c_j \tag{3}$$

where m, n are the numbers of disturbed input data and parameters. F_k corresponds to the expression of $\mathscr{F}$ relative to the output s_k.

- If the function $\mathscr{F}$ exhibits one or more singularities, it is necessary to proceed to a case by case study. The calculation of partial derivatives can be developed according to various approaches due to the complexity of the model studied. The ideal case lies in the possibility of analytically establishing these primary derivatives. The direct differential analysis then proceeds to the derivation of equations of the model. Thus, designating by S ={s_1, s_2,..., s_q} the output vector of a system of equations (which can be non-linear) :

$$f(s^n,\dots,s^1,\dot{s},\alpha_1,\dots,\alpha_m) = 0 \tag{4}$$

with $\dot{s} = \frac{\partial s}{\partial t}$, α_i grouping together the data (control parameters and input data of the system considered). The s^n, ... , s^j, ... s^1 denote derivatives of space of the order *n*,..., *j*, ...*1*. Introducing as a new solution to the problem the function of sensitivity of output $\chi = \left.\frac{\partial s}{\partial \alpha_i}\right|_{\alpha_0}$, its differentiation gives the following system [3] :

$$\left.\frac{\partial f}{\partial s^n}\right|_{\alpha_0}\chi^n + \dots + \left.\frac{\partial f}{\partial s}\right|_{\alpha_0}\chi = -\left.\frac{\partial f}{\partial \alpha}\right|_{\alpha_0} \tag{5}$$

These equations of output sensitivity give the sensitivities of all the output according to the input data observed. This approach may be adopted whenever the elements of the above equation (5) are easily accessible. However this opportunity is not frequent due either to the high degree of complexity of the equations, or to the fact that the model is not explicit (numerical or experimental model). In addition, the use of this approach automatically makes numerous changes of the theoretical model necessary to be able to estimate the derivatives. This continues to be difficult to carry out on complex models.

The possibility that remains is to approximate calculations using the method of finite differences. Each partial derivative will be evaluated using the following relation :

$$\frac{\partial F_k}{\partial e_j} \approx \frac{F_k(E+\delta E, C) - F_k(E, C)}{\delta e_j} \qquad \text{with,} \qquad \delta E = [0,....,0,\delta e_j,0,...0]^T \qquad (6)$$

where δE is equal to a weak perturbation (compared to 1) to the j jth position, and zero everywhere else. For a first order calculation , δe_j is included between 10^{-3} and 10^{-6} . Therefore m + n + 1 evaluations of $\mathcal{F}$ are necessary to calculate the Δs_k. Other authors have described Finite Difference method in a sensitivity analysis context. It is important to note that the Finite Difference method (marked FD) [4] is different of Finite Differences Differential Analysis method (marked FDDA). Indeed, FD method gives an approximate value for the derivative $\partial F_k / \partial e_j$ for a central value $e_{j,o}$, between $e_{j,o} - \Delta e_j$ and $e_{j,o} + \Delta e_j$, where [$e_j - \Delta e_j$, $e_j + \Delta e_j$] is the data variation interval. Our calculation reaches a numerical estimation derivative value with a very small displacement δe_j around $e_{j,o}$ value (6). An approximative derivative calculation is nevertheless correct for linear function F_k with smooth variations, but is not efficient for non-linear cases with rapid variations which are not studied here.

Application to coupled heat transfert in building thermal non-linear model.

We propose to treat a simple example of a non-linear model which consists in studying radiative and convective exchanges of a room in a dwelling unit.

We are interested in the evolution of the temperature of each side according to the intensity of the physical inputs (figure 1). Face 1 receives a surface energy E_0, while it simultaneously disperses heat by convection (exchange coefficient h_c , and reference temperature T_{ref}). It then radiates towards the other surfaces. Faces 2 and 3 are subjected to coupling between radiation and convection. Faces 4 are assumed to have very low convective exchange coefficients which are then neglected in the thermal balance equation of the face.

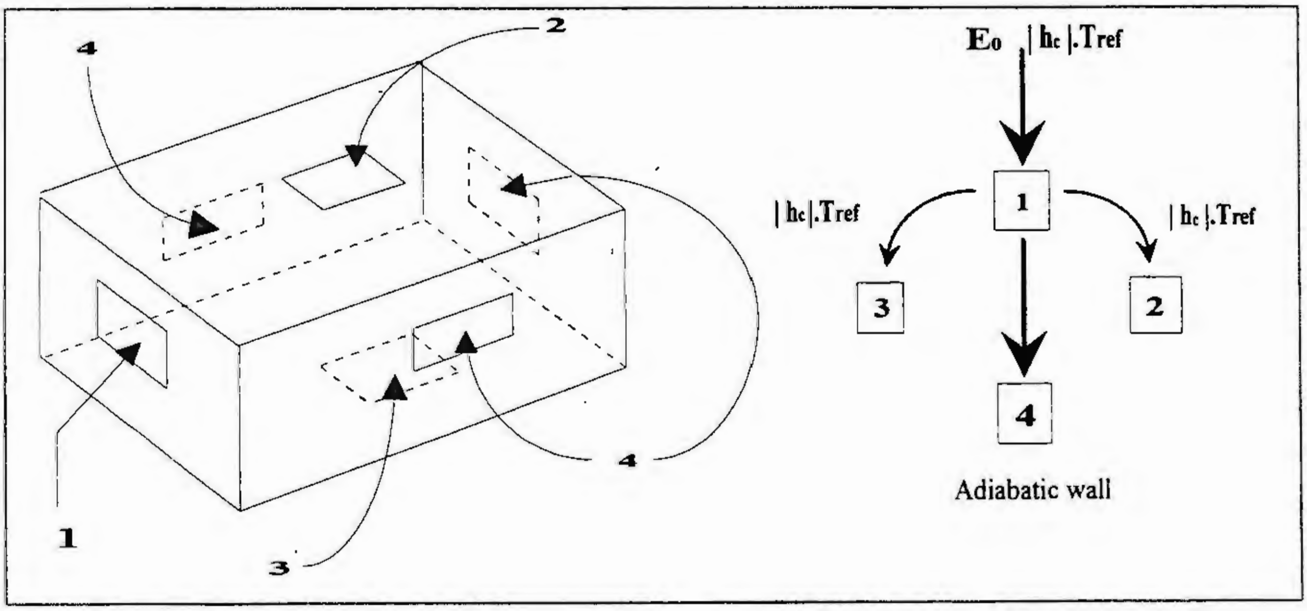


Figure 1 - View of the enclosure and the applied input

Table 1 groups together the values of the thermophysical parameters, as well as their respective fluctuations,

8 data values	Surface energy Eo (W/m²)	Envelope dimensions (m) $L_1 \times L_2 \times L_3$	View factors	. Convective coefficient (W / m². K)	Reference temperature (K)
Reference value	$E_o = 400$	6 x 3 x 3	$F_{31} = 0.217$, $F_{32} = 0.292$	$h_c = 12.0$	$T_{ref} = 291$
Uncertainty	± 10 %	± 2 %	± 8 %	± 9 %	± 5 %

Table 1 - Values of thermophysical parameters and the corresponding uncertainty.

according to the specifications generally sought by experimenters on measurable sizes. After assesment of the 4 faces studied, the following system of nonlinear equations is obtained:

$$\begin{aligned}
&\frac{h_C}{\sigma_0}T_1 + T_1^4 - F_{12}T_2^4 - F_{13}T_3^4 - F_{14}T_4^4 = \frac{E_o}{\sigma_o} + \frac{h_C}{\sigma_0}T_{ref} \\
&-F_{21}T_1^4 + \frac{h_C}{\sigma_0}T_2 + T_2^4 - F_{23}T_3^4 - F_{24}T_4^4 = \frac{h_C}{\sigma_0}T_{ref} \\
&-F_{31}T_1^4 - F_{32}T_2^4 + \frac{h_C}{\sigma_0}T_3 + T_3^4 - F_{34}T_4^4 = \frac{h_C}{\sigma_0}T_{ref} \\
&-F_{41}T_1^4 - F_{42}T_2^4 - F_{43}T_3^4 + (1-F_{44})T_4^4 = 0
\end{aligned} \tag{7}$$

associated as well with the relationships of complementarity and symmetry for the calculation of view factors not coming from the monogram reading. That is :

$$S_i\, F_{ij} = S_j\, F_{ji}\,, \text{ with } \sum_{j=1}^{n} F_{ij} = 1$$

where S_i and S_j represents the different surfaces of the envelope. In this example, data vector is :

$$E = \left[E_0, L_1, L_2, L_3, F_{12}, F_{13}, \ldots\ldots, F_{43}, F_{44}, h_c, T_{ref}, \sigma_o\right] \text{ with 20 components.}$$

Except for σ_o and F_{ij} for which the uncertainty is supposed negligible, an uncertainty interval is assigned to other E components (Table 1). Nevertheless, F_{ij} view factors having a major importance in radiant heat exchange, we arbitrary introduce uncertainties for only two factors F_{31} and F_{32}, in order to test their influence. So, ΔE vector is reduced to 8 components.

Relation $\mathscr{F}$ here corresponds to the solution of the non-linear system (7), providing solution T:

$$T = \mathscr{F}(E) = [\,T_1,\ T_2,\ T_3,\ T_4\,]^T \qquad \textbf{\textit{requiring 1 computation.}}$$

Using relation (3), we have :

$$\Delta T_k = \left|\frac{\partial T_k}{\partial E_o}\right|\Delta E_0 + \sum_{j=1}^{j=3}\left|\frac{\partial T_k}{\partial L_j}\right|\Delta L_j + \left|\frac{\partial T_k}{\partial F_{31}}\right|\Delta F_{31} + \ldots\ldots\ldots\ldots + \left|\frac{\partial T_k}{\partial T_{ref}}\right|\Delta T_{ref} \tag{8}$$

For each data (with attached uncertainty intervals), we compute 9 vectors T:

$$\frac{\partial T_k}{\partial E_o} = \frac{\left(T_k + \delta T_k\big|_{\delta E_o}\right) - T_k}{\delta E_o} \quad \ldots\ldots\ldots \quad \frac{\partial T_k}{\partial T_{ref}} = \frac{\left(T_k + \delta T_k\big|_{\delta T_{ref}}\right) - T_k}{\delta T_{ref}} \qquad \textbf{\textit{requiring 9 computations}}$$

The resolution of the system of nonlinear equations $T = \mathscr{F}(E)$ was carried out through the use of the Levenberg-Marquard algorithm [2]. It should be noted however that the estimation of the Jacobian matrix according to the finite differences formula was completed by a relationship which makes automatic the generation of an adaptive step *hj* as a function of input data considered e_j and the numerical computer accuracy [5]. A global systematic *hj* value has not been used as it was the case in the study of linear systems. Figure 2 superimposes the projections of the values obtained by Monte Carlo simulations and the boundaries of fluctuations of the corresponding output variables.

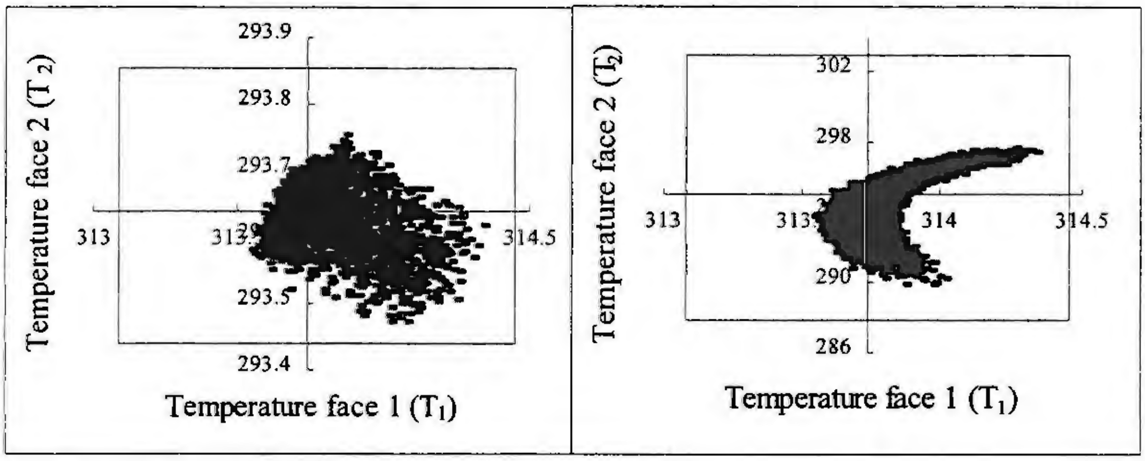


Figure 2 - Results obtained for the nonlinear model of coupled exchanges : Monte Carlo values and uncertainty of the Finite Differences Differential Analysis. Temperature are given with Kelvin.

As for time computation performance, time rate is about 1 to 100, between the FDDA and the MC method. The differential analysis which we have implemented is fully correct despite the totally nonlinear nature of the physical process studied. To complete this presentation of results, table 2 sums up for each surface temperature, the fluctuation resulting from the uncertainties on the parameters and the input data of the model.

Temperatures (K)	Non disturbed values	FDDA Uncertainties (9 simulations)	FDDA Uncertainties (%)	MC Uncertainties (800 simulations)*	MC Uncertainties (%)
Face 1 : T_1	313.74 K	± 0.7 K	± 0.2 %	- 0.2 K + 0.61 K	- 0.06 % + 0.2 %
Face 2, 3: T_2	293.64 K	± 0.2 K	± 0.07 %	- 0.17 K + 0.09 K	- 0.06 % + 0.03 %
Face 4 : T_4	295.01 K	± 7.7 K	± 2.6 %	- 5.15 K + 2.73 K	- 1.75 % + 0.92 %

Table 2 - Uncertainties associated with temperatures on each face of the envelope.
(* number of computation must be greater than 100 for a reliability interval less than 15% of final standard deviation s_D value. This case involved about 800 computations to reach 5% of final s_D value).

Comparatively, differential analysis restricted to first order therefore offers the possibility of constructing a relatively exact framing of the fluctuations of the output of the model, both linear and nonlinear. The consistency of the results with the Monte Carlo simulations is all the more favourable as the cost of the calculation involved in setting up the differential analysis remains quite acceptable, and much below than that of the Monte Carlo method. It should be remembered however that the two methods share the quality of being able to consider the uncertainties on the input data parameters *with few restrictions* on their amplitude. Differential analysis does however offer the additional possibility of an estimation of the sensitivity of all of the output with respect to *each* uncertain parameter or piece of input data. Figure 3 summarises the incidence of each input data on the variations of different output, leaving aside the assumed input data fluctuation interval.

It can therefore be observed that the view factor F_{31} is the most influential parameter with a major influence on all of the temperatures. The other influential parameters are, in the order of importance, the reference temperature, the exchange coefficient, and the surface energy received by the wall.

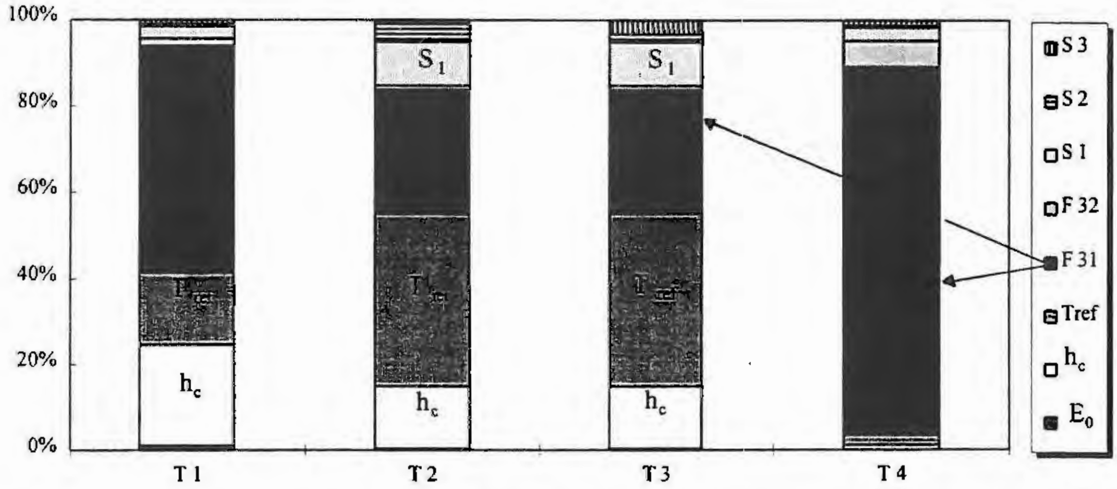


Figure 3 - Influence of each piece of input data on the fluctuation of output in the case of a highly nonlinear model of coupled radiative and convective exchanges. The preponderant influence of the F_{31} view factor on the temperatures can be observed, particularly with T_1 and T_4.

Conclusion

We have compared, through the coupled heat exchange example in buildings, two uncertainty evaluation methods. These two methods involve different approaches. The first is a probabilistic type (Monte Carlo, MC), the second a deterministic type (Finite Differences Differential Analysis, FDDA).

The FDDA method has to be particularly effective. Regarding to robustness, we have observed that the uncertainty interval described by this method almost systematically includes the values computed by the MC method. In addition, this framing of the MC results by first order approximation of FDDA does not lead to an excessive extension of the area of uncertainty of the results, but on the contrary, narrows the extrema of the cloud. The interval which the FDDA method leads to is therefore always more pessimistic than that obtained by the MC method, but the difference does not exceed 2%. The FDDA first order approximation therefore proves to be satisfactory.

Regarding to effectiveness, the calculation times necessary to obtain the uncertainty interval are far below those of the MC method. As for performance, calculating time rate is 1 to 100, showing that FDDA method is more time saving than the probabilist reference MC method. The only precaution necessitated by the FDDA method is within the determination of the calculation step of partial derivatives. This is a particularity of the FDDA method which must be entered in the calculation code. But it is likely that in deterministic type approaches, a sequential analysis of the numerical and mathematical treatment of the model is essential. Nevertheless, when this is done, and as the partial derivatives are coded, the uncertainty of the output vector is easily computed. Another considerable advantage of the FDDA method is the possibility of estimating explicitly the sensitivity of each output element on all of the input data. Therefore it is possible to envisage using it inversely by transmitting information back to the necessity of reducing the uncertainty of certain data.

Finally we note that this method imposes few *a priori* restrictions concerning the nature and the amplitude of the uncertainties associated with data.

References

[1] **KREYSZIG, E.** (88): Advanced Engineering Mathematics, Wiley, New-York

[2] **IMSL Math/Library Users Manual**(89) : IMSL Inc., 2500 City West Boulevard, Houston, TX 77042

[3] **FRANK, P.M.** (78) : Introduction to System Sentivity Theory, Academic Press, New-York

[4] **LOMAS K. J, EPPEL H,** (92) : Sensitivity analysis techniques for building thermal simulations programs, *Energy and Buildings, 19 (1992) 21-44*

[5] **DENNIS, J.E, SCHNABEL, R.B.** (83) : Numerical Methods for Unconstrained Optimisation and Nonlinear Equations, NJ: Prentice Hall, Englewood Cliffs

FUZZY MODELING FOR THE BIOSYNTHESIS CONTROL

L. Rusinov, V. Holodnov, G. Panov
St. Petersburg State Technological Institute (State University)
26, Moscowsky pr. 198013, St. Petersburg, Russia

Abstract. Untraditional approach for industrial biosynthesis control is proposed. It implies appliance the continuous diagnostics of the current process state and using obtained information for determination of the necessary control operations. The diagnostics is based on the two level hierarchical diagnostic model which includes both expert and deep knowledge. The diagnostic model consists of the following kinds of models formalizing this knowledge: frame-net at upper level and sets of production rules and mathematical model at lower level.

Introduction

The modern biotechnological processes in most cases have a high level of the uncertainty, so there are problems in the control of these processes for their efficiency and stability improvement. It is caused by a lot of reasons. First of all it is the complexity of the modeling processes with living objects, when it is essential to take into account individual character of the strain growth and vital functions which may be different even at the identical mode of the process running. The conventional mathematical models suitable for the use in control systems very often fail to describe deviations in the process flow caused by individual peculiarity of biological agents (microorganisms, funguses etc.). So it is necessary to use the adaptive models and control algorithms that will change the set of their basic parameters with reference to a current situation in fermenter [2, 4 - 6].

Another reason which causes the high level of uncertainty is the lack of sensors that can provide automatic measurement of the concentrations of metabolism product components in on-line mode. At the same time the duration of the laboratory analyses of these parameters is rather great. It means that the results of the laboratory analyses reflect a situation which occurred in fermenter some hours ago. So the results cannot be used directly in the control systems. Moreover for successful operation of the process it is necessary to consider such hard formalizable factors as the appearance of biomass, its color, the value of its air saturation and others, that give important information about the process current state. All above-mentioned problems are well illustrated by the Citric Acid biosynthesis.

Target process description

A simplified scheme of the process is shown in Fig. 1. The sterile nutrient medium (water solution of sugar and mineral salts) and inoculum (germinated mycelium of the Aspergilus Niger) supplied into the fermenter (stream Medium In) before the beginning of the Citric Acid fermentation process. The temperature in the fermenter is stabilized by controller TC (1) which affects the cooling system. Air stream (Air In) is continuously supplied for breathing and mixing the medium. This input flow is regulated by the flow controller FC (2). The fermenter pressure is maintained by manipulating Air Out stream using controller PC (3). The increase of the foam level in the fermenter above the critical point invokes the supply of Antifoam Agent (controller LC (4)). The Feed stream entering process loop is controlled by flow controller FC (5). Moreover the Control System (CS) allows to monitor the values of mass (WR (6)), heat evolution (AR (7)) of the medium and specific speed of oxygen consumption by the medium in the fermenter (BR (8)).

Such CS can control the process quite successfully in the case of its normal functioning. But if the fermentation process develops with some deviations (caused for example by peculiarities of the concrete micelium) the CS may fail. We cannot use the laboratory information on concentrations of Citric Acid, biomass and sugar because of the large delays in obtaining of the laboratory analyses results. The measurement errors here can reach large values (about 30%) in relation to the moment when samples were taken. The CS also fails if some faults in the measurement or control apparatus occur.

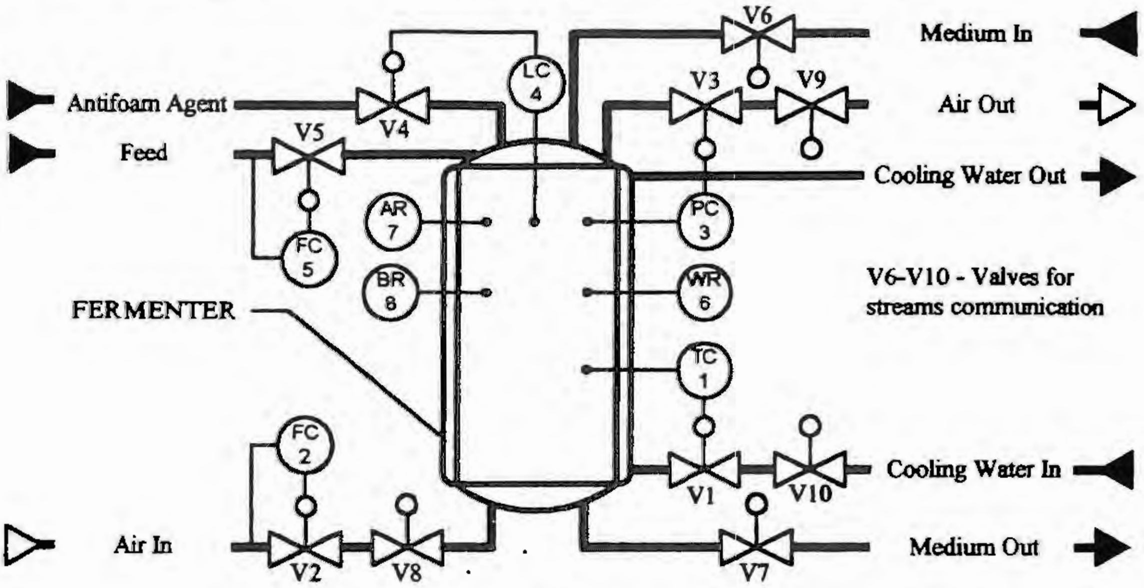


Fig. 1 Simplified technological scheme of the Citric Acid fermentation

To overcome these difficulties and make the level of uncertainty maximally low it is suggested to control the process by means of the information obtained from the knowledge-based Subsystem of Monitoring and Continuos Diagnostics (SMCD) of the process current state. The algorithm of the subsystem functioning takes the Diagnostic Model (DM) of the process as the basis. DM allows to identify a situation actually existing in the process at any given moment and to take necessary steps timely.

The Structure of the Diagnostic Model

For the synthesis of the DM that would be valid at different stages of the process it is necessary to use both the expert (empirical) and deep (theoretical) knowledge [1, 3, 7]. This way we can avoid drawbacks of the systems using one type of knowledge. Unlike usual mathematical model (MM) DM describes mainly the abnormal states of the process and includes MM as its integral part. In our case DM is applied for identification current situations in the process and early detection possible deviations from the normal process flow and its causes. On the basis of this information the SMCD generates the forecast of given situation development, defines the control actions corresponding this situation and passes them to the CS.

At the same time the MM describes effects of the biomass and target product accumulation and sugars consumption. It is used for rational on-line control of the feeding flow into fed-batch reactor as well as for forecast the general way of the fermentation process development. On the basis of this forecast economically expedient moment when the fermentation process should be ceased is calculated and fact of the medium infection in fermenter is recognized.

Thus the system Knowledge Base must include not only different type of knowledge but also different forms of its presentation. For the effective unification of this heterogeneous knowledge it is proposed to apply hierarchic two-level DM with the frame-situation model of net structure on its upper level (Fig. 2). The arcs of the net show interrelations between typical technological operations (let us call them macrosituations) for accomplishment of the whole technological process. The frames in the nodes (frames of macrosituations FrS_i) have the same set of slots which includes information describing both these situations and control operations for the normal process flow. More detailed information about any abnormal situations (faults) in the process (let us call them micrositutions) and control operations in these cases is gathered in the affiliated frames FrC_{ij}.

The decomposition of the Citric Acid biosynthesis process into the technological operations (macrosituations) allows to consider them one by one and to reduce dimension of lower level models without the increasing of upper level model dimension and thus to simplify computation.

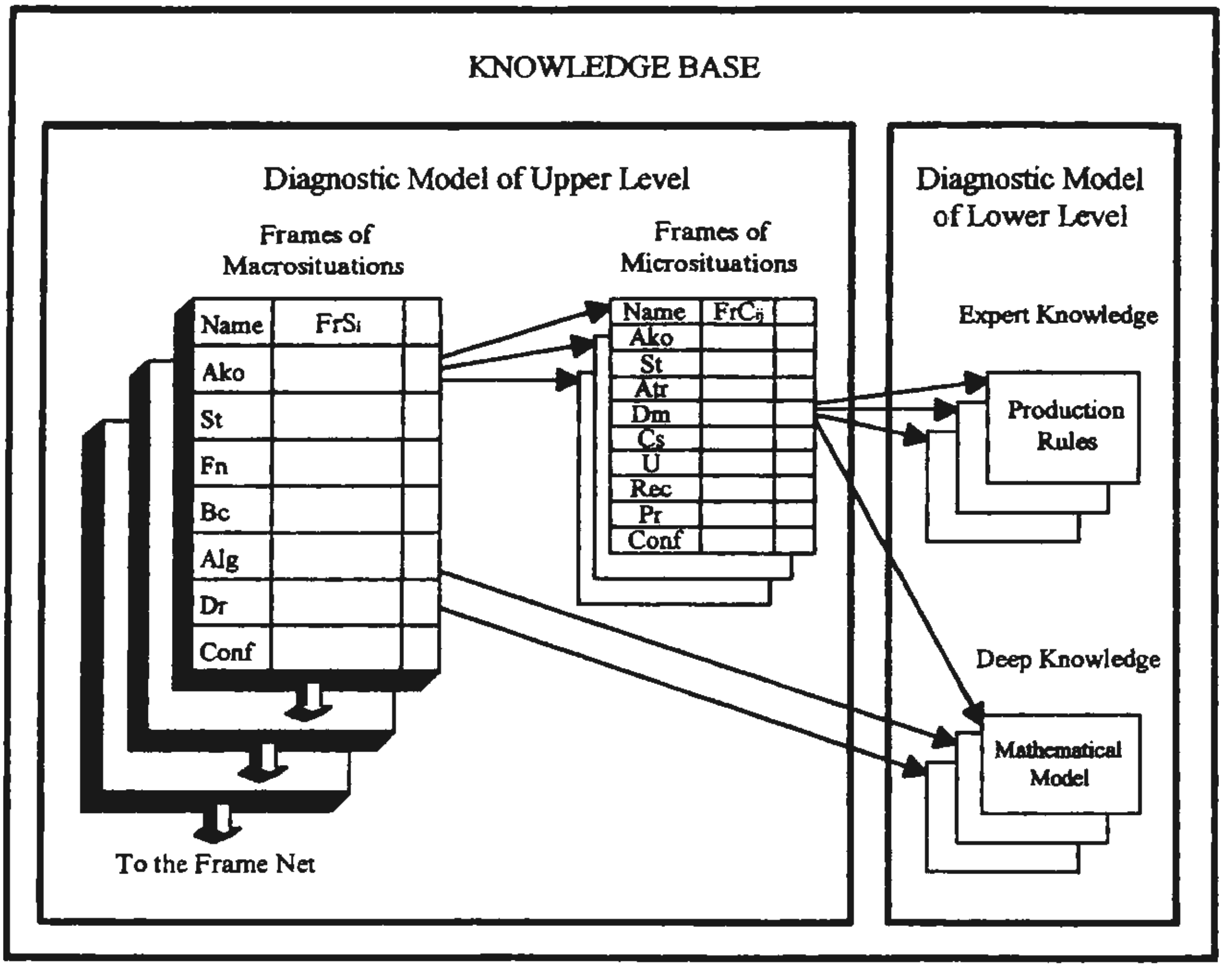


Fig. 2 The structure of the Diagnostic Model

The frames of macrosituations and microsituations

The typical frame FrS_i of macrosituation S_i may be shown in the following way (the slots of the frame are listed in braces):

FrS_i = {*Name, Ako, St, Fn, Bc, Alg, Dr, Conf*} (1).

Each slot contains either quantitative or semantic information which relates to a particular technological operation and describes one or another its aspect as pointed out below:

Ako indicates the names of the affiliated frames that are connected with this frame;

St contains information about a status of the macrosituation determined by a special procedure Pc_i. This slot has two values: *St=1* (if this operation runs) and *St=0* (if not);

Fn is the number of the fermenter where the process under control is conducted ;

Bc represents a set of initial parameters necessary for the running of the technological operation (set points for the controllers, the positions of valves and so on);

Alg contains the algorithms of running the technological operation in the case of its normal flow;

Dr has an information about the duration or conditions of ceasing of the technological operation;

Conf points at the frame which is the next in the frame net and toward which it is necessary to move after fulfillment of the given technological operation..

Let us examine FrS_4 "Fermentation" as an example of a macrosituation frame (Fig. 3). It is necessary to note that the realization of this technological operation in the case of its normal flow is the independent functioning of several algorithms (slot *Alg*): the algorithms of change depending on time the set points of temperature TC (1) and air flow FC (2) controllers; adaptive algorithms for pressure stabilization by means of controller PC (3) and for foam removal and environment pollution prevention using controller LC (4); algorithm of an optimum feeding profile determination on the basis of MM and rational feeding flow control by means of flow controller FC (5). Economically expedient moment when the fermentation process should be finished (slot *Dr*) may be forecasted approximately in a hour beforehand by means of special algorithm Alg_6 by application of MM (it is the instant when potential profit from the target product accumulation will be less than the expenses for the further process realization).

The faults that occur during the fermentation as a technological operation (slot *Ako*) may be divided into three groups, associated with: 1) sensors faults (frames FrC_{401}-FrC_{408}); 2) discrepancy between actual value of the parameter and its set point (frames FrC_{409}-FrC_{416}); 3) deviations from the normal biotechnological process flow that are not displayed at the monitored parameters supplied by automatic sensors (FrC_{417}-FrC_{427}).

The causality between faults and its consequences (symptoms) was elicited for successful fault diagnostics and appears the meaning of the lower level of DM. The simplest way to formalize this knowledge is to use the production rules or fuzzy sets, but in the latter case we need some more information to synthesize the membership functions so we choose the previous one as the method for knowledge presentation.

The typical frame FrC_{ij} describing a microsituation C_{ij} and belonging to the macrosituation frame FrS_i may be delineated by the following expression:

FrC_{ij} = {*Name, Ako, St, Atr, Dm, Cs, U, Rec, Pr, Conf*} (2),

where *Ako* indicates the name of the root frame connected with this one, whose values of attributes it inherits;

St is a status of the microsituation. This slot has 3 values: detected, possible and not detected. The microsituation is considered as detected if all symptoms connected with it are observed in the process at the given moment. It is considered as possible if one from all symptoms is not observed or an extra one is observed . In other cases the value of status of the microsituation is not detected. There is the special procedure Pc_2 for definition of the microsituation status.

Atr contains 2 groups of parameters: list and total number of symptoms specifying this fault (microsituation) and list and number of them actually detected in the process;

Dm is a fragment of the process DM which is applied to the concrete microsituation and performed by means of rules used in the fault analysis. The left parts of the rules contain conditions (set of the fault symptoms), whose fulfillment (coincidence with the symptoms actually arisen in the process) should be accompanied by activization of the right parts of rules. The last ones explicate the reasons of the fault, define the necessary control actions and give them out to the object, provide personnel with the recommendations and prognosis of the process development;

Cs has information about the causes of the fault occurrence;

U contains control actions that are necessary to give out to the object through the CS (amendment of the controllers set points, change-over the channels of control of some parameters, activization or cancellation of special algorithms and so on);

Rec represents recommendations to the personnel about the maximally efficient operations in the case of this fault occurrence;

Pr calculates prognosis of the current process state development;

Conf points at the names of the frames of competitive microsituations, that are necessary to take into account for successful determination of the most probable fault, explaining the received set of symptoms.

The structure of a microsituation frame is illustrated on the frame FrC_{423} "Infection of the medium in the fermenter " (Fig. 4). In this case the fragment of DM (slot *Dm*) contains 5 production rules for fault diagnosis. The following symbols are put into practice here:

A, A1 and *B, B1* are actual and limit values of the medium heat evolution and specific speed of oxygen consumption respectively in the case of fermentation normal functioning;

t, t1 are the actual time of the operation and an agreed moment of the shift from the biomass growth stage to fermentation in the full sense respectively;

$D7=1$ and $D8=1$ denote that the channels of measurement the heat evolution and specific speed of oxygen consumption are in order;

MI is the result of the microscope investigation of the medium ($MI=0$ if infection is not detected and $MI=1$ if infection is detected);

$\Delta s/\Delta t$, *DS* and $\Delta p/\Delta s$, *DPS* are the actual and limit values of sugar consumption speed and the quotient of the target product accumulation speed to sugar consumption speed accordingly (these values are calculated on the basis of MM;

p, p1 are the actual value of the target product concentration and the economically expedient target product contents in the medium for the Citric Acid purification.

Mathematical Model

Inexplicit character of the fermentation process (as it was mentioned above) necessitates the utilization of both the empirical and deep knowledge. The last one is presented in the Knowledge Base of SMCD by MM. All aforesaid functions of MM in the SMCD call for using the dynamic MM with adaptation of its parameters to the real conditions of the particular fermentation process.

The structure of MM of the Citric Acid fermentation can be formed from the following theoretical reasoning. Specific growth rate of the biomass is limited by low sugar concentration and by high biomass concentration (effect of the space lack). Sugar acts as both limiting and inhibiting factor for specific production rate. The sugar is spent for the biomass growth, maintenance of its vital activity and for Citric Acid synthesis. For this purposes the sugar solution is put into bioreactor not only before fermentation but throughout it also (the Feed stream at Fig.1). MM of the Citric Acid fermentation is represented by three differential equations reflecting changing rates of biomass, target product and sugar concentrations:

$$\frac{dx}{dt}=\frac{k_1\cdot s\cdot x}{(k_2+s)\cdot(k_3+x)}-\frac{F}{V}\cdot x \qquad (3),$$

$$\frac{dp}{dt}=\frac{k_4\cdot s\cdot x}{(k_5+s)\cdot(k_6+s)} \qquad (4),$$

$$\frac{ds}{dt}=-k_7\cdot\frac{k_1\cdot s\cdot x}{(k_2+s)\cdot(k_3+x)}-k_8\cdot\frac{k_4\cdot s\cdot x}{(k_5+s)\cdot(k_6+s)}-k_9\cdot x+\frac{F}{V}\cdot(s_F-s) \qquad (5),$$

where *x, p, s* - concentrations of the biomass, target product and sugar accordingly; k_1 - k_9 - model factors; *F* and s_F are the flow rate and sugar concentration in the feed stream; *V*- volume of the medium. The parametric identification of the model is carried out using the data obtained in the passive experiment conducted on the Belgorod Citric Acid Plant, Russia. The results of this identification are given in Table 1.

Table 1

The values of Mathematical Model parameters

Parameter of MM	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9
Unit	g / l·h	g / l	g / l	g / l·h	g / l	g / l			h^{-1}
Value	1.15	98	31.2	15.1	24.8	3.9	2.7	0.71	0.12

Slots	Values	Notes
Name	Fermentation	FrS_4
Ako	A fault in the channel of measurement: the temperature in fermenter; the input air flow; the pressure in fermenter; the foam level in fermenter; the feed flow; the mass of the medium; the heat evolution; specific speed of the oxygen consumption.	FrC_{401} . . FrC_{408}
	Temperature in the fermenter is elevated.	FrC_{409}
	Temperature in the fermenter is lowered.	FrC_{410}
	Input air flow is elevated.	FrC_{411}
	Input air flow is lowered.	FrC_{412}
	Pressure in the fermenter is elevated.	FrC_{413}
	Pressure in the fermenter is lowered.	FrC_{414}
	Feed flow in the fermenter is elevated.	FrC_{415}
	Feed flow in the fermenter is lowered.	FrC_{416}
	Reduction of the biomass activity.	FrC_{417}
	Lack of oxygen during the biomass growth.	FrC_{418}
	Lack of sugar at the stage of the biomass growth.	FrC_{419}
	Lack of sugar during the fermentation.	FrC_{420}
	Lack of oxygen during the fermentation.	FrC_{421}
	Excessive growth of the biomass.	FrC_{422}
	Infection of the medium in the fermenter.	FrC_{423}
	Level of the medium in the fermenter is increased.	FrC_{424}
	Level of the foam in the fermenter is increased.	FrC_{425}
	Ejection of the medium from the fermenter.	FrC_{426}
	Leak of the medium from the fermenter.	FrC_{427}
St	St=1 / St=0	Pc_1
Fn	0.. .n	
Bc	Set of the initial parameters	
Alg	Algorithm of fermenter temperature control.	Alg_1
	Algorithm of input air flow control.	Alg_2
	Algorithm of pressure stabilization.	Alg_3
	Algorithm of foam removal.	Alg_4
	Algorithm of rational feeding flow control.	Alg_5
Dr	Determination of the operation cease moment	Alg_6
Conf	Thermal inactivation of the medium	FrS_5

Fig. 3 Macrosituation frame FrS_4 "Fermentation"

Slots	Values	Notes
Name	Infection of the medium in the fermenter	FrC_{423}
Ako	Fermentation	FrS_4
St	Detected / Possible / Not detected	Pc_2
Atr	A, B, t, D7, D8, MI, $\Delta s/\Delta t$, $\Delta p/\Delta s$, p	

Dm	if $A>A1 \& B>B1 \& t>t1 \& D7=1 \& D8=1$, then Cs1,Rec1,Pr1;
	if $MI=0 \& \frac{\Delta s}{\Delta t} > DS1 \& \frac{\Delta p}{\Delta s} < DPS1$,then Cs2, U1;
	if $MI=1 \& \frac{\Delta s}{\Delta t} > DS1 \& \frac{\Delta p}{\Delta s} < DPS1$, then Cs3, Rec2, U2;
	if $MI=1 \& \frac{\Delta s}{\Delta t} > DS2 \& \frac{\Delta p}{\Delta s} < DPS2 \ \& p>p1$, then Cs4, U3, Rec3, Pr2;
	if $MI=1 \& \frac{\Delta s}{\Delta t} > DS2 \& \frac{\Delta p}{\Delta s} < DPS2 \ \& p<p1$, then Cs4, U3, Rec4, Pr3	

Cs	Cs1: either infection of the medium in the fermenter or excessive growth of the biomass; Cs2: excessive growth of the biomass is possible; Cs3: moderate medium infection in the fermenter; Cs4: the medium in the fermenter is entirely infected	
U	U1: examination of competitive microsituation (slot Conf); U2: to make adversity for strange microorganism; U3: to move on the next macrosituation frame FrS_5	
Rec	Rec1: to carry out additional laboratory analyses and microscope investigation (*MI*), input new data in the SMCD; Rec2: to identify the species of the strange microorganism; Rec3: to rout infected medium for the product purification; Rec4: to use infected medium as substratum in other process	
Pr	Pr1: the medium infection is possible; Pr2: be ready to receive medium for the purification; Pr3: be ready to receive medium for substratum preparation	
Conf	Excessive growth of the biomass	FrC_{422}

Fig. 4 Microsituation frame FrC_{423} "Infection of the medium in the fermenter"

The parameters of MM have to be adjusted to the current fermentation conditions in order to predict development and execute advanced control of the particular process. However it is necessary to adjust not all parameters but only those ones the sensitivity of Discrepancy Function to their change is especially high. The results of calculation of Static Sensitivity Coefficients (SSC) for each parameter are given in Table 2.

Table 2

The values of Static Sensitivity Coefficients

Parameter of MM	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9
Value of SSC	0.71	0.09	0.69	0.11	0.09	0.23	0.19	0.76	0.12

As anybody can see from the Table 2 only k_1, k_3 and k_8 have to be adjusted. Since the determined parameters enter in the MM linearly and one by one in the different differential equations, the adjusting procedure becomes much simplified. Such adjusting takes place every 4 hours after reception of the new data of laboratory analyses. In addition the quantities of received feed and medium picking out are taken into account.

MM allows to determine the feeding general strategy which is one of the basic factors for improvement of the process economy: to calculate the sugar concentration $\bar{s} = \sqrt{k_5 \cdot k_6}$ which ensures the maximum fermentation speed; to predict the instance $\bar{t}$ when this concentration would be reached and to find out rational feeding flow *F (t)* which would maintain this concentration and therefore a maximum fermentation speed (last algorithm from slot *Alg* in Fig. 3).

Summary

The suggested method for the control of fed-batch bioreactor can work successfully in the conditions of high level of uncertainties that take place in the processes with living objects. The application of Diagnostic Subsystem as an integral part of the Control System allows to get the robust control in the case of on-line information deficit. To overcome the problem the Diagnostic Model uses both empirical and deep knowledge about the process. Two level hierarchic Diagnostic Model with frame net at upper level and set of production rules and mathematical model at lower level allows to decrease the dimension and thus to simplify computation.

This approach was developed first of all for the Citric Acid fermentation process but it can be applied to other biotechnological processes as well.

References

1. Basila, M.R., Stefanek, G. and Cinar, A. A., Model-object based supervisory expert system for fault tolerant chemical reactor control. Computers and chemical engineering, 14 (1990), 551 - 560.
2. Berber , R., Control of batch reactors . Chemical engineering research and design, 74, pt. A (1996), 3 - 20.
3. Dhurjati, P.S., Lamb, D.E. and Chester, D., Experience in the development of an expert system for fault diagnosis in commercial scale chemical process. In: Foundations of computer aided process operations, (Eds.: Reklaitis, G.V. and Spriggs, H.D.) Elsevier, Amsterdam, 1987, 589 - 625.
4. Lee, S.C., Hwang, Y.B., Chang, H.N., Chang, Y.K., Adaptive control of dissolved oxygen concentration in bioreactor. Ibid, 37 (1991), 597 - 607.
5. Mou, D.G. and Cooney, C.I., Growth monitoring and control through computer aided on-line mass balancing in a fed-batch penicillin fermentation. Biotechnology and Bioengineering, 25 (1983), 225 - 255.
6. Shioyn , S., Optimization and control in fed-batch reactors. In: Advances in biochemical engineering and bioreactors, Springer, Berlin, 46 (1992), 111 - 142.
7. Venkatasubramanian, V. and Rich, S.H., An object-oriented two-tier architecture for integrating compiled and deep knowledge for process diagnosis. Ibid, 12 (1990), 903 - 921 .

TENSOR ANALYSIS BASED SYMBOLIC COMPUTATION FOR MECHATRONIC SYSTEMS

K. Schlacher, A. Kugi and R. Scheidl
Johannes Kepler University Linz
Altenbergerstraße 69, A–4040 Linz, Auhof

Abstract. This contribution presents methods for the mathematical modeling of mechatronic systems based on tensor analysis in combination with graph theory. Tensor analysis is an effective and universal tool for the common description of electrical and mechanical systems in a geometric way. Efficient algorithms for time–dependent Lagrangian systems with nonholonomic constraints are developed as well as an extension of the theorem of Brayton–Moser to general n–port networks. Therefore the combination of electrical and mechanical systems is achieved in a straightforward way. The so obtained methods for setting up the mathematical models are optimized for treatment by computer algebra as well as for numerical simulation.

Introduction

Using the well known notation of tensor algebra in the first part of the paper we discuss the fundamentals of Lagrangian systems. An efficient algorithm for the derivation of the equations of motion for a system of rigid bodies connected together by different types of joints is proposed. This algorithm is designed in such a way, that time dependent Lagrangian systems and non holonomic constraints can be taken into consideration, as well as the costs for a symbolic treatment and numerical calculations are minimized. The second part of the paper concerns the modeling of electrical systems based on Kirchhoff networks. Analogously to Lagrangian systems a geometric oriented description based on tensor analysis is given. However, the most important result of these considerations is the extension of the theorem of Brayton–Moser [2] to a general class of electrical n–port networks. Due to the common description of Lagrangian systems and electrical networks, the mathematical modeling of electromechanical systems is achieved in a straightforward manner in the third part. Furthermore, we show that under certain integrability conditions the well known principle of coenergy functions can be directly deduced from our results. All proposed algorithms are especially useful for numerical simulation as well as for computer algebra programs in combination with object oriented design strategies [3]. On this manner we can overcome the restriction of the over–polynomial increase of the calculation time and memory of symbolic computation, and this makes these methods applicable for practical problems.

Lagrangian Systems

Modeling of systems of rigid bodies is based on the theory of Lagrangian systems here. Therefore we need effective methods to describe these system as well as methods to form complex systems by connecting simple ones. An n–dimensional Lagrangian system in canonical coordinates (x, v) is described by the equations

$$\frac{\mathrm{d}}{\mathrm{d}t}x_i = v_i \ , \qquad \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial v_i}L - \frac{\partial}{\partial x_i}L = F_i \ , \qquad i = 1, \dots, n \, , \tag{1}$$

where $L(t, x, v)$ is the Lagrangian function and F_i are generalized forces. It is well known, that the action $\int L\mathrm{d}t$ is extremized for $F_i = 0$.

To achieve a geometric description of Lagrangian systems, we take an $n + 1$ dimensional manifold $\mathcal{M}$ with coordinates (t, x) as base manifold. Let $\gamma(\tau) : R \to \mathcal{M}$ be a curve with $\gamma(\tau) = (T, X)$, where upper cases are used for a curve, then its tangent vector is given by $\gamma' = T'\partial_t + \sum_{i=1}^{n} X_i'\partial_{x_i} \in \mathcal{TM}$ with $X_i' = \frac{\mathrm{d}}{\mathrm{d}\tau}X_i$. We assign to each point γ of the base manifold $\mathcal{M}$ the linear space $\Delta(\gamma) = \mathrm{span}\{\gamma'\}$. The annihilator $\Delta^{\perp}$ of Δ is the set of a all 1–forms α_i^L such that $\gamma^*\alpha_i^L = \gamma' \rfloor \alpha_i^L = 0$ with the pullback operation $\gamma^* : \Lambda\mathcal{M} \to \Lambda R$ and the interior product $\rfloor$ of a vector and a form. If $T' \neq 0$ holds, a basis of $\Delta^{\perp}$ is given by $\{\mathrm{d}x_i - v_i\mathrm{d}t\}$, $i = 1, \dots, n$, where d denotes the exterior derivative [1]. Taking $T = t = \tau$ we get velocity type vectors $\dot{\gamma} = \partial_t + \sum_{i=1}^{n} \dot{X}_i\partial_{x_i}$ with t–component equal to 1. Now a curve $\gamma(t)$ can be lifted from $\mathcal{M}$ to $\mathcal{TM}$ by $\tilde{\gamma}(t) = \left(\gamma, \partial_t + \sum_{i=1}^{n} \dot{X}_i\partial_{x_i}\right) \in \mathcal{TM}$, on the other hand a curve $\gamma(t) = (\gamma, \partial_t + \sum_{i=1}^{n} V_i\partial_{x_i}) \in \mathcal{TM}$ is the lift of a curve on $\mathcal{M}$, iff

$$\dot{\gamma} \rfloor \alpha_i^L = 0 \ , \qquad \text{with} \qquad \alpha_i^L = \mathrm{d}x_i - v_i\mathrm{d}t \ , \qquad i = 1, \dots, n \tag{2}$$

holds for $\dot{\gamma} \in \mathcal{T}\mathcal{T}\mathcal{M}$ or $\dot{X}_i = V_i$, where the forms α_i^L are transferred from $\Lambda^1\mathcal{M}$ to $\Lambda^1\mathcal{T}\mathcal{M}$. Here we have constructed a special fiber bundle, that is called a line element contact bundle with contact forms α_i^L. This bundle has coordinates (t, x, v) at least locally. Taking the 1–forms $\beta_i^L = \mathrm{d}\partial_{v_i}L - \partial_{x_i}L\mathrm{d}t$, we can easily see that a curve $\gamma(t) = (t, X, V) \in \mathcal{T}\mathcal{M}$ describes a motion of the Lagrangian system, iff its tangent $\dot{\gamma} = \partial_t + \sum_{i=1}^n \dot{X}_i\partial_{x_i} + \dot{V}_i\partial_{v_i} \in \mathcal{T}\mathcal{T}\mathcal{M}$ meets the conditions

$$\dot{\gamma}\rfloor\left(\alpha_i^L \wedge \beta_i^L\right) + F_i\alpha_i^L = 0\ , \qquad \dot{\gamma}\rfloor\alpha_i^L = 0\ , \qquad i = 1, \ldots, n\ . \tag{3}$$

It is worth to mention, that on the line element contact bundle $L\mathrm{d}t$ is a well defined 1–form and (2), (3) are valid for time dependent Lagrangian L, too.

Of special interest are transforms $f : \mathcal{M}_1 \to \mathcal{M}_2$ of the type $(t, y) \to (t, x(t, y))$ between the bases of the fiber bundles. If $\alpha_i = \mathrm{d}x_i - v_i\mathrm{d}t$ are the contact forms on $\mathcal{M}_2$, then their pullback $f^*\alpha_i$ to $\mathcal{M}_1$ is given by $f^*\alpha_i = \sum_j \partial_{y_j}x_i\mathrm{d}y_i - (v_i - \partial_t x_i)\,\mathrm{d}t$. Comparing them with the contact forms $\mathrm{d}y_i - u_i\mathrm{d}t$ on $\mathcal{M}_1$, we see that the contact structure is preserved, iff $v_i = \partial_t x_i + \sum_j \partial_{y_j}x_iu_i$ holds. Therefore, the lift of f to $\tilde{f} : \mathcal{T}\mathcal{M}_1 \to \mathcal{T}\mathcal{M}_2$ is given by $(t, y, u) \to \left(t, x(t, y), \partial_{(t,y)}x(t, y)u\right) = (t, x, v)$, and a curve $\gamma_1 \in \mathcal{T}\mathcal{M}_1$ describes a motion, iff

$$\dot{\gamma}_1\rfloor\left(\tilde{f}^*\alpha_i \wedge \tilde{f}^*\beta_i\right) + \tilde{f}^*F_i\alpha_i = 0\ , \qquad \dot{\gamma}_1\rfloor\tilde{f}^*\alpha_i = 0\ , \qquad i = 1, \ldots, n \tag{4}$$

holds, where $\tilde{f}^*$ pulls the forms from $\mathcal{T}\mathcal{M}_2$ back to $\mathcal{T}\mathcal{M}_1$. These equations are equivalent to using $\frac{\mathrm{d}}{\mathrm{d}t}y = u$, $\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial}{\partial u}L_1 - \frac{\partial}{\partial y}L_1 = F_2\partial_y x$ with $L_1(t, y, u) = L_2\left(t, x(t, y), \partial_t x + \partial_y xu\right)$, the pullback of L_2 to $\mathcal{T}\mathcal{M}_1$. Also of interest are more general transforms of the kind $\tilde{f} : \mathcal{T}\mathcal{M}_1 \to \mathcal{T}\mathcal{M}_2$ with $(t, y, u) \to (t, x(t, y), v(t, y, u))$, which do not necessarily preserve the Lagrangian structure. Here the equations of motion are given by (4), the pullback of (3), but they cannot be recovered from the pullback of the Lagrangian alone in general.

Lagrangian systems can be built up from simpler ones in a straightforward manner. Let us look at two systems with Lagrangian L^a, L^b, forms $\beta_i^{L_a}$, $\beta_i^{L_b}$ and generalized forces F_i^a, F_i^b, respectively, which are modeled on the same manifold $\mathcal{M}$ with contact forms α_i^L. For each system the equations of motion follow from (3). The equations of the compound system with Lagrangian $L = L^a + L^b$ and general forces $F_i = F_i^a + F_i^b$ are obviously given by

$$\dot{\gamma}\rfloor\left(\alpha_i^L \wedge \left(\beta_i^{L_a} + \beta_i^{L_b}\right)\right) + \left(F_i^a + F_i^b\right)\alpha_i^L = 0\ , \qquad \dot{\gamma}\rfloor\alpha_i^L = 0\ . \tag{5}$$

This result is a direct consequence of $\beta_i^L = \beta_i^{L_a} + \beta_i^{L_b}$.

The Rigid Body Package

The rigid body package offers models for rigid bodies as well as a selection of different joints, springs and dampers. A rigid body is modeled by a Lagrangian system and joints are considered as additional restrictions of the system only. Therefore non holonomic restrictions are allowed as well as time dependent Lagrangian systems.

If one adds restrictions of the type $h(t, x) = 0$ to a Lagrangian system (3), one restricts the movement to a submanifold of the base manifold. Under some conditions $h(t, x) = 0$ is equivalent to $x = f(t, y)$ due to the implicit function theorem for some y. In this case $x = f(t, y)$ induces a simple, structure–preserving transform, which is given by (4). Of course these considerations apply to more general restrictions of the type $g(t, x, v) = 0$, too. For the sake of simplicity we assume, they are equivalent to $x = f(t, y)$ and $v = g(t, y, u)$ for some y, u, then it is easy to see that the induced coordinate transform is described by (4), too. But this transform does not preserve the structure in general.

A system of rigid bodies is formed by connecting rigid bodies, joints and springs, etc.. Its modeling is based on Lagrangian systems. A graph $G = (N, B)$ is related to the system. The node set N contains the rigid bodies, and the branch set contains joints, dampers, springs, etc.. Nodes are Lagrangian systems, and branches generate restrictions like $g(t, x) = 0$, $g(t, x, v) = 0$ or generalized forces F_i. Since it is straightforward to add generalized forces to the system, we do not pursue this case here.

At the first stage we assume that G is a simple chain of length $n + 1$, starting with the system 0. Let x^{i-1}, x^i denote the coordinates of the systems of the nodes $i - 1$, i connected by the branch i, which generates a restriction of the type $x^i = f^i(t, y^i, x^{i-1})$ with the joint coordinates y^i. Let us assume that the joint variables y can be used as generalized coordinates. Since the transforms of the last section are based on the pullback operation only, we can setup the equation of the whole system by the following simple algorithm. Start with $G_0 = \{N_0, B_0, \}$, $B_0 = \{\}$, $N_0 = \{0\}$ and coordinates (t, x^0) and repeat the following steps until you reach the node n: 1) Take the system of node i and derive its model using the coordinates $(t, x^0, y^1, \ldots, y^i, x^i)$. 2) Use (5) to combine this model with the extension of the model of G_{i-1} from $(t, x^0, y^1, \ldots, y^{i-1})$ to $(t, x^0, y^1, \ldots, y^i, x^i)$. 3) Pull this result from $(t, x^0, y^1, \ldots, y^i, x^i) = (t, x^0, y^1, \ldots, y^i, x^i(x^0, y^1, \ldots, y^i))$ back to $(t, x^0, y^1, \ldots, y^i)$ by (4) and set $B_i = B_{i-1} \cup \{i\}$, $N_i = N_{i-1} \cup \{i\}$, $G_i = \{N_i, B_i\}$. It is easy to see that this algorithm works for time dependent Lagrangian systems as well as for non holonomic constraints. It is straightforward to extend this algorithm from simple chains to trees. If the graph $G = (N, B)$ is not a tree, we construct a suitable spanning tree $G' = \left(N, B'\right)$, $B' \subset B$. Whenever it is possible to solve the additional equations induced by the branches $B \setminus B'$ algebraically, this is done by the package. Otherwise, numerical methods or the Lagrange multiplier technique must be used. Furthermore, this algorithm offers an effective strategy to minimize the number of operations necessary for numerical simulation or further investigations with computer algebra systems.

Kirchhoff Networks

Modeling of electrical networks is based on the theory of Kirchoff networks. The circuit is formed by connecting terminals. A graph $G = (N, B)$ is related to the network. N is the set of *nodes* of cardinality n, and B is the set of branches with cardinality b. A branch has exactly two end points which must be nodes. For the sake of simplicity only connected graphs are considered here. The current i_l and the voltage u_l are assigned to the branch l, and the potential v_j is assigned to the node j. The voltage and current law can be expressed as

$$\sum\nolimits_i v_i d_{ij} = v_j d_{jl} + v_k d_{kl} = -u_l \quad \text{and} \quad \sum\nolimits_j d_{ij} i_j = 0\ , \tag{6}$$

where nodes j and k are connected by the branch l, and d_{ij} is defined by

$$d_{ij} = \begin{cases} 1 & \text{if node } i \text{ and branch } j \text{ are connected, the direction of } i_j \text{ to the node } i \text{ is positive} \\ -1 & \text{if node } i \text{ and branch } j \text{ are connected, the direction of } i_j \text{ off the node } i \text{ is positive} \\ 0 & \text{otherwise}. \end{cases}$$

A current $i^T = (i_1, \ldots, i_b) \in R^b$ and a voltage $u = (u_1, \ldots, u_b) \in \left(R^b\right)^*$ are said to be admissible, if they obey the current law and the voltage law, respectively. Using the map $D : R^b \to R^n$, given by $\sum_i d_{ij} i_i = x_j$, one can easily show, that i is admissible, iff $i \in \operatorname{Ker} D$ and $u \in \operatorname{Im} D^*$. This justifies the choice $u \in \left(R^b\right)^*$. Let $R = \{r_i\}$ be a basis of $\operatorname{Ker} D$ and $S = \{s_i\}$ a basis of $\operatorname{Im} D^*$, respectively. The equations

$$i = Rx\ , \quad u = yS \quad \text{and} \quad uR = 0\ , \quad Si = 0 \tag{7}$$

are equivalent to both the laws of Kirchhoff. Now Telegens [5] theorem, which proves $ui = 0$, whenever u and i are admissible, follows directly from the equations above. If we take $(i, u) \in R^b \times \left(R^b\right)^*$, then the configuration manifold $\mathcal{M}$ of the network is given by

$$\mathcal{M} = \left\{(i, u) \in R^b \times \left(R^b\right)^* \mid uR = 0,\ Si = 0\right\}\ . \tag{8}$$

It is a well known fact that R and S can be constructed from a subgraph, a tree, $G' = \left(N, B'\right)$, $B' \subset B$, which contains no loops. Suitable algorithms based on the search method "depth first" are implemented in the package.

Electrical Networks

Electrical networks are special Kirchhoff networks with graph $G = (N, B)$, therefore their configuration manifolds $\mathcal{M}$ are given by (8) with coordinates (i, u). We consider here general n-ports, which include dynamic elements like inductors and capacitors with n-ports, as well as static elements like resistors, voltage and current sources, amplifiers, etc.. For the sake of simplicity the graph $G = (N, B)$ is finite and connected and we consider time invariant networks only.

First we subdivide the set B into 3 disjoint sets L, C and F of subsets in a way, such that the set $i \in L$, C, F contains all ports of the inductor i, capacitor i or static terminal i, respectively. Clearly, $B = \bigcup_{i\in L} L_i \cup \bigcup_{i\in C} C_i \cup \bigcup_{i\in F} F_i$ holds. The differential equations of the dynamic elements are given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_j^i(\ldots, i_k, \ldots) = u_j \ , \quad j,k \in i \in L \qquad \text{and} \qquad \frac{\mathrm{d}}{\mathrm{d}t}q_j^i(\ldots, u_k, \ldots) = i_j \ , \quad j,k \in i \in C \ , \tag{9}$$

and ψ_j denotes the flux linkage and q_j the charge linkage of the j–th port. The static element i is described by equations of the type

$$f_j^i(\ldots, u_k, \ldots, i_k, \ldots) = 0 \quad \text{with} \quad j,k \in i \in F \ . \tag{10}$$

Now we make the following *strong assumption* for the sake of simplicity: The variables (i_j, u_k), $j \in i \in L$, $k \in i \in C$ form a chart, a system of coordinates, of the configuration manifold $\mathcal{M}$. Then we can determine the following functions $i_l = i_l(\ldots, i_j, \ldots, u_k, \ldots)$, $u_l = u_l(\ldots, i_j, \ldots, u_k, \ldots)$, $l \in B$ with $j \in i \in L$, $k \in i \in C$. E.g. one can use the Kirchhoff package described in the last section to set up this set. If this assumption is not met, we have to extend the presented methods to the general case. Since this can be done in a simple way and has no effect on the following consideration, we limit us to those networks which meet the strong assumption.

Analogously to Lagrangian systems, we look for a geometric–oriented description of the network and start with the manifold $\mathcal{M} = R \times R^b \times \left(R^b\right)^* \times R$ with coordinates (t, i, u, p) and $p = \sum_{i\in B} u_i i_i$. Any solution $\gamma(t) = (t, I_i, U_i)$ must satisfy (7), therefore it lies in the submanifold $\mathcal{M}_0$ with $p = 0$ because of Telegen's theorem. If the tangent vector $\dot\gamma = \partial_t + \sum_{j\in B}(\dot I_j \partial_{i_j} + \dot U_j \partial_{u_j})$ satisfies $\dot\gamma \rfloor \mathrm{d}p = 0$, then y remains in $\mathcal{M}_0$, whenever this is true for one point $\gamma(t_0)$. This structure is a special fiber bundle again, which is called a hypersurface element contact bundle with contact forms $\mathrm{d}p = \sum_{j\in B} u_j \mathrm{d}i_j + i_j \mathrm{d}u_j$ for all admissible voltages u and currents i. Because of (7), the contact forms $\mathrm{d}p$ have a finite basis. The equations of motion follow from the pullback of the special 1-forms β_j^I, β_j^U

$$\beta_j^I = \mathrm{d}\psi_j^i - u_j \mathrm{d}t \ , \quad \beta_j^U = \mathrm{d}q_j^i - i_j \mathrm{d}t \qquad \text{and} \qquad \dot\gamma \rfloor \beta_j^I = 0 \ , \ j \in k \in L, \quad \dot\gamma \rfloor \beta_j^U = 0 \ , \ j \in k \in C \tag{11}$$

induced by the curve $\gamma(t) = (t, I_i, U_i)$. It is worth to mention that these results can be extended to time variant networks in a straightforward manner.

Although (11) together with the strong assumption describes the network completely, it is still possible to reduce the number of equations. By our strong assumption we have a map

$$f : (t, i_i, u_k) \rightarrow (t, i, u) \qquad \text{with} \qquad j \in i \in L, k \in i \in C \ , \tag{12}$$

and we will use its pullback f^* to transfer the equations from $\mathcal{M}$ back to $\mathcal{M}_0$. By a little abuse of notation we see, that $f^*(\beta_j^I) = \mathrm{d}\psi_j^i - f^*(u_j)\mathrm{d}t$ and $f^*(\beta_j^U) = \mathrm{d}q_j^i - f^*(i_j)\mathrm{d}t$ is fulfilled, and we have to express u_j, i_j as functions of the new coordinates only. Of course we also have $f^*(p) = 0$, $f^*(\mathrm{d}p) = 0$, as well as $f^*(u_j \mathrm{d}i_j) = f^*(u_j)\mathrm{d}i_j$ for inductors and $f^*(i_k \mathrm{d}u_k) = f^*(i_k)\mathrm{d}u_k$ for capacitors. Since $f^*(\mathrm{d}p^I)$, $f^*(\mathrm{d}p^U)$ are obviously linear independent, $f^*(\mathrm{d}p^I) = f^*(\mathrm{d}p^U) = 0$ must hold, too. Combining these equations we get

$$\sum\nolimits_{j\in i\in L} f^*(u_j)\mathrm{d}i_j = -f^*\left(\mathrm{d}\sum\nolimits_{j\in i\in C} u_j i_j + \sum\nolimits_{j\in i\in F} u_j \mathrm{d}i_j\right) + \sum\nolimits_{j\in i\in C} f^*(i_j)\mathrm{d}u_j \ .$$

If $\mathrm{d}\omega = 0$ with $\omega = f^*\left(\mathrm{d}\sum_{j\in i\in C} u_j i_j + \sum_{j\in i\in F} u_j \mathrm{d}i_j\right)$ is fulfilled, then we can find a function $f^*(\hat q)$ with $\mathrm{d}f^*(\hat q) = \omega$, and we get immediately

$$\sum\nolimits_{j\in i\in L}\left(f^*(u_j) + \frac{\partial}{\partial i_j} f^*(\hat q)\right)\mathrm{d}i_j = \sum\nolimits_{j\in i\in C}\left(f^*(i_j) - \frac{\partial}{\partial u_j} f^*(\hat q)\right)\mathrm{d}u_j \, .$$

Since the forms $\mathrm{d}i_j$, $\mathrm{d}u_j$ in the equation above are linear independent, the terms in parentheses must vanish. $\hat q$ can be easily calculated, if we define for each static terminal $i \in F$ the functions

$$\hat p^i = \sum\nolimits_{j\in i}\int^{(i)} u_j \mathrm{d}i_j \qquad \text{and} \qquad \check p^i = \sum\nolimits_{j\in i}\int^{(u)} i_j \mathrm{d}u_j \, , \tag{13}$$

assuming that the integrability conditions $\mathrm{d}\sum_{j\in i} u_j \mathrm{d}i_j = \mathrm{d}\sum_{j\in i} i_j \mathrm{d}u_j = 0$ are met. This is possible for terminals like resistors, sources, amplifiers, etc.. The sum $\hat{p}^i + \check{p}^i = p_F^i$ is clearly the flow of power into the terminal i. From $\hat{q} = \sum_{i\in C} p_i + \sum_{i\in F} \hat{p}^i$, we get immediately $-\hat{q} = \sum_{i\in L} p_i + \sum_{i\in F} \check{p}^i$. With the abbreviations $p_K^i = \sum_{j\in i} u_j i_j$, $K \in \{L, C, F\}$ for the flow of power in the terminal i we have the final result:

Theorem 1 *The equation of motion of an electrical network (9), (10), which meets the strong assumption and the integrability conditions* $\mathrm{d}\sum_{j\in i} u_j \mathrm{d}i_j = 0$ *for* $i \in F$, *are given by*

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_j^i = -\frac{\partial}{\partial i_j} f^* \left(\sum\nolimits_{i\in C} p_C^i + \sum\nolimits_{i\in F} \hat{p}^i\right) \ , \quad \frac{\mathrm{d}}{\mathrm{d}t} q_j^i = -\frac{\partial}{\partial u_j} f^* \left(\sum\nolimits_{i\in L} p_L^i + \sum\nolimits_{i\in F} \check{p}^i\right) \tag{14}$$

with $\hat{p}^i$, $\check{p}^i$ *from (13) and* f *from (12).*

It is worth to mention that these results are an extension of the theorem of Brayton–Moser [2] to a wider class of networks. Here we emphasize the symmetry between inductors and capacitors, too.

Of further interest is the stored energy in the dynamic elements. E.g. the flow of power into an inductor $i \in L$ or capacitor $i \in C$ is given by

$$w_L^i = \int_\gamma \sum\nolimits_{j\in i} i_j \mathrm{d}\psi_j^i \ , \qquad w_C^i = \int_\gamma \sum\nolimits_{j\in i} u_j \mathrm{d}q_j^i \tag{15}$$

for a curve $\gamma(t)$, $t_0 \leq t \leq t_1$. If the values of w_L^i, w_C^i are path independent, or $\mathrm{d}\sum i_j \mathrm{d}\psi_j^i = 0$, $\mathrm{d}\sum_{j\in i} u_j \mathrm{d}q_j^i = 0$ holds, then we are able to find the functions w_L^i and w_C^i such that $\mathrm{d}w_L^i = \sum_{j\in i} i_j \mathrm{d}\psi_j^i$, $\mathrm{d}w_C^i = \sum_{j\in i} u_j \mathrm{d}q_j^i$ are met.

The energies $w_L = \sum_{i\in L} w_L^i$, $w_C = \sum_{i\in C} w_C^i$ together with the functions $\hat{p} = \sum_{i\in F} \hat{p}^i$, $\check{p} = \sum_{i\in F} \check{p}^i$ allow a simple stability analysis of the network. Here we assume that all integrability conditions are met. Let the function $w = w_L + w_C$ be positive definite for an equilibrium of the network, then w can be used as a candidate for a Liapunov function of the network. From

$$\frac{\mathrm{d}}{\mathrm{d}t} w = -f^* \left(\hat{p} + \check{p}\right)$$

one can conclude that w is a Liapunov function of the network, iff $f^* \left(\hat{p} + \check{p}\right)$ is positive semi definite.

Electromechanical Systems

Here modeling of electromechanical systems is based on Lagrangian theory and the electrical networks presented in the former sections. The mathematical modeling starts with the cross product of the line element and hypersurface contact bundle with coordinates (t_m, x, v, t_e, i, u), with t_m the time of the mechanical part and t_e the time of electrical part, respectively. Setting $t = t_m = t_e$ we get the coordinates (t, x, v, i, u) for the compound system and we can extend the exterior derivative d and the interior product $\rfloor$ from the parts to the configuration manifold in a straightforward manner. We denote the exterior derivative operating in the variables (t, x, v) and (t, i, u) with d_m and d_e, respectively.

We need some assumptions about the way to combine the two systems. A curve γ on the configuration manifold is given by $\gamma = (t, X, V, I, U)$ of course. We assume that the energy transfer between the mechanical part and the electrical part occurs in the inductors and capacitors only. Let the sets L, C and F be the same as above. We define the set M_i for each dynamic terminal of the electrical network, such that $j \in M_i$ means, the variable x_j of the mechanical part is associated with the terminal i of the electrical part. We assume, that the generalization of (9) is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_j^i(\ldots, i_k, \ldots, x_l) = u_j \ , \ j, k \in i \in L \ , \ l \in X_i \ , \quad \frac{\mathrm{d}}{\mathrm{d}t} q_j^i(\ldots, u_k, \ldots, x_l) = i_j \ , \ j, k \in i \in C \ , \ l \in X_i \, . \tag{16}$$

It is easy to see that the pullpack of $\beta_j^I = \mathrm{d}\psi_j^i - u_j \mathrm{d}t$, $\beta_j^U = \mathrm{d}q_j^i - i_j \mathrm{d}t$ induced by γ is equivalent to (16). It is worth to mention that the effect of the mechanical part on the electrical part is linear in the variables v. To find the effect of the electrical part on the mechanical part, we make the following assumption: The relation

$$\dot{\gamma} \rfloor \sum\nolimits_{j\in i} u_j i_j \mathrm{d}t = \dot{\gamma} \rfloor p_K^i \mathrm{d}t + \dot{\gamma} \rfloor \sum\nolimits_{k\in X_i} F_k^i \mathrm{d}x_k \ , \qquad K \in \{C, L\} \tag{17}$$

holds for any curve $\dot{\gamma}$. $\sum_{j\in i} u_j i_j$ is the electrical power, $\sum_{j\in X_i} F_j^i v_j$ is the mechanical power with generalized coupling forces F_j^i, and p_K^i is the flow of power stored in the terminal i. E.g. from (14) we get the relation

$$p_L^i = \frac{\mathrm{d}}{\mathrm{d}t} w_L^i = \frac{\mathrm{d}}{\mathrm{d}t} \int_\gamma \sum\nolimits_{j\in i} i_j \mathrm{d}_e \psi_j^i = \dot{\gamma}\rfloor \sum\nolimits_{j\in i} i_j \mathrm{d}_e \psi_j^i + \dot{\gamma}\rfloor \sum_{k\in M_i} \left(\frac{\partial}{\partial x_k} \int_\gamma \sum\nolimits_{j\in i} \mathrm{d}_e \left(i_j \psi_j^i \right) - \psi_j^i \mathrm{d}i_j \right) \mathrm{d}x_k$$

for inductors. The combination of this result with (11), (17) leads to

$$\dot{\gamma}\rfloor \sum\nolimits_{k\in X_i} F_k^i \mathrm{d}x_k = \dot{\gamma}\rfloor \sum\nolimits_{k\in M_i} \mathrm{d}x_k \frac{\partial}{\partial x_k} \int_\gamma \sum\nolimits_{j\in i} \psi_j^i \mathrm{d}i_j \ .$$

Since this equation must hold for any curve γ, we get immediately the expressions

$$F_k^i = \frac{\partial}{\partial x_k} \int_\gamma \sum\nolimits_{j\in i} \psi_j^i \mathrm{d}i_j \ , \ i \in L \ , \ k \in M_i \ , \qquad F_k^i = \frac{\partial}{\partial x_k} \int_\gamma \sum\nolimits_{j\in i} q_j^i \mathrm{d}u_j \ , \ i \in C \ , \ k \in M_i \qquad (18)$$

for the generalized coupling forces F_j^i of the inductors as well as for the generalized coupling forces F_j^i of the capacitors because of analogous considerations. If (18) meets the integrability conditions $\mathrm{d}_e \sum_{j\in i} \psi_j^i \mathrm{d}i_j = 0$ and $\mathrm{d}_e \sum_{j\in i} q_j^i \mathrm{d}u_j = 0$, respectively, which are equivalent to the integrability conditions of (15), we can define the functions

$$\tilde{w}_L^i = \int_\gamma \sum\nolimits_{j\in i} \psi_j^i \mathrm{d}i_j \qquad \text{and} \qquad \tilde{w}_C^i = \int_\gamma \sum\nolimits_{j\in i} q_j^i \mathrm{d}u_j \ . \qquad (19)$$

$\tilde{w}_L^i$, $\tilde{w}_C^i$ are called coenergies, too.

Finally it is easy to set up the equations of the total system. The electrical part is given by (16), and the generalized coupling forces F_k^i for the mechanical part follow from (18). If it is possible to calculate the coenergies (19), we can modify the forms β_i^L of (3) to $\beta_i^L = \mathrm{d}\partial_{v_i}\tilde{L} - \partial_{x_i}\tilde{L}\mathrm{d}t$, $\tilde{L} = L + \tilde{w}_L + \tilde{w}_C$ with $\tilde{w}_L$ and $\tilde{w}_C$ as the sum of coenergies of all electrical terminals, which are connected with the Lagrangian system.

Summary

In this paper we have presented some efficient methods for the mathematical description of mechatronic systems. Since the modeling of mechatronic systems requires the knowledge of various fields of research, we have focused our attention to a common geometric oriented description and notation based on tensor analysis. We have shown that by means of a simple algorithm the equations of motion of time variant Lagrange systems with non holonomic constraints can be easily derived. As an important result for electrical systems an extension of the well known theorem of Brayton-Moser to n-port networks is given. Furthermore, the connection of electrical and mechanical systems to electromechanical systems is achieved in a straightforward way. The proposed methods can be easily implemented in any computer algebra system. The special implementation in MAPLE V, described in [4], has now been extended with the proposed algorithms. It is worth to mention, that this package can handle hydraulic networks, too.

References

1. Burke W.L., Applied Differential Geometry, Cambridge University Press, 1994.
2. Hirsch M.W. and Smale St., Differential Equations, Dynamical Systems and Linear Algebra, Academic Press, Inc., 1974.
3. Kugi A., Schlacher K. and Kaltenbacher M., Object Oriented Approach for Large Circuits with Substructures in the Computer Algebra Program Maple V, In: Software for Electrical Engineering Analysis and Design, Ed. P.P. Silvester, 1995, 491–500.
4. Schlacher K. and Scheidl R., Modeling of Mechatronic Systems by Symbolic Computation, In: Proc. of the 1995 Eurosim Conference, Vienna, 1995, 657–662.
5. Zemanian A. H., Infinite Electrical Networks, Cambridge University Press, 1991.

BOND GRAPH APPROACH FOR RELATIVE DEGREES AND ZERO DYNAMICS ANALYSIS OF LINEAR SYSTEMS

R. Fotsu Ngwompo, S. Scavarda and D. Thomasset
Laboratoire d'Automatique Industrielle de Lyon
Bât.303, INSA, 20, av. A. Einstein
F-69621 Villeurbanne Cedex, France.

Abstract: During the modeling phase in control system design, before deriving and analysing the dynamic equations of the model, it is helpful to determine some properties of the system which do not depend on numerical values of the parameters. This study called structural analysis can be carried out conveniently using a bond graph approach. The advantage of bond graph being that it is close to the physical representation of the system and it provides an interpretation of the results in term of physical phenomena and the interconnections in the system. This paper considers the problem of structural determination of relative degrees and zero dynamics analysis for linear control systems based on their bond graph representation. Some previous results obtained on this topic are recalled and a new approach more general using the bond graph concept of bicausality is presented. This study has some applications in the design of tracking controller and the problem of input-output decoupling.

Introduction

In diverse problems of system control design, it is often necessary to study relative degrees and the zeros dynamics of the system. Unlike the poles of the system which are intrinsic system properties, the relative degree and zero dynamics of the system depend on the chosen input-output pair. They are then, related to the locations of the sensors and actuators on the physical system and they have an effect on the overall stability of the control system. For instance, in perfect output tracking problem, relative degrees indicate how many times the predefined output trajectories should be time-differentiable and stability of zero dynamics is related to the boundedness or the level of oscillations of control actions in performing tracking objectives.

From a practical point of view, it is interesting during modeling and design of control systems to carry out a structural analysis of the system before calculations and to keep as far as possible a physical interpretation of the system properties. Structural properties of a system are then properties which do not depend on the numerical values of its parameters but they are generic properties depending only on the type of elements and the way they are interconnected. The purpose of this work is to present a general bond graph based method for determining relative degrees and analysing zero dynamics of physical systems. In [6], some results on the problem considered here was presented but the proposed method based on conventional bond graph did not provide general results in a systematic approach for all bond graph configurations and particularly for multiple input-output power line systems.

In this paper, a general rule applicable to linear systems modelled by bond graph for the determination of relative degrees is given and our study focuses on particular cases which are exceptions to the proposed rule. Two classes of exceptions are distiguished depending whether they are structural (i.e. independent of the parameters of the elements) or not and two examples are given to illustrate both cases. For zeros dynamics analysis, we use the Mason loop rule and the inverse bond graph model as the zeros of a system are the poles of its inverse and we show that this approach which is more general allows to study in a procedural way the case of multiple power lines between the input and the output variables. Inverse bond graph is constructed using the concept of bicausality introduced by Gawthrop [3] and the procedure for inversion of linear SISO system which was proposed in [2]. Let us recall that the method for application of Mason loop rule to bond graph given by Brown in [1] remains roughly valid for extended causal bond graph with mixed causal and bicausal bonds even if particular configurations due to bicausal bonds may occur.

1. Determination of the relative degree from the direct bond graph

The problem of determining the relative degree of an output of a system modelled by bond graph is considered here. The rule proposed to find the relative degree is an extension to MIMO linear systems of that given in [6]. In all the paper, the direct bond graph denotes the causal bond graph obtained from a procedure such as SCAP [4] or MSCAP [5]. We start by some definitions:

Definition 1: For a m-input m-output sytem, the relative degree d_j of an output y_j is defined as the minimum number d_j so that an input component appears explicitly in the expression of $y_j^{(dj)}$ where $y_j^{(dj)}$ is the d_j-th time-derivative of y_j.

Definition 2: In a direct bond graph, the order $\omega_p(u_i,y_j)$ of an input-output causal path p from the input u_i to the output y_j is: $\omega_p(u_i, y_j) = n_I(p) - n_D(p)$ where $n_I(p)$ (resp. $n_D(p)$) is the number of energy storage elements in integral (resp. derivative) causality on that path p.

As $\omega_p(u_i,y_j)$ indicates the net number of integrations between the input variable u_i and the output variable y_j along the path p, we then state the following result.

Rule1: The relative degree d_k of the output y_k for a m-input m-output system is given in general by:

$$d_k = \min_{i=1,\cdots,m} \left[\min_{p \in P_{ik}} \left(\omega_p(u_i, y_k) \right) \right] \tag{1}$$

where P_{ik} denotes the set of all input-output paths from u_i to y_k.

However, it may occur in some particular cases that the true relative degree denoted d'_k be greater than the number d_k defined by (1). To develop this point, we consider the case of SISO linear systems and notice that from calculus carried out in [2], if d is the output relative degree then it comes that:

$$\text{for } r < d,\ y^{(r)} = \sum_{k=1}^{n} G_{O_{r+1}}(\dot{x}_k, y) x_k$$

$$y^{(d)} = \sum_{k=1}^{n} G_{O_{d+1}}(\dot{x}_k, y) x_k + G_{O_d}(u, y) u$$

where $G_{O_k}(v_i, v_j)$ is the sum of constant factors (or static terms) in the gain of all k-order causal paths from variable v_i to variable v_j and x_k are energy variables associated with energy storage elements k.

The exceptional cases where the relative degree is larger than d may then arise when there exist more than one path with minimal order d so that: $G_{O_d}(u, y) = 0$.

As we show on two examples given below, this condition may be structural (i.e. independent of parameters values) or it may depend on some algebraic constraints between the system parameters.

Example 1: The system represented by the bond graph of Fig.1 is considered in [6] and its structural relative degree is two. It has two minimal-order paths shown on the bond graph and to examine exceptional cases when the relative degree may be larger than two, we have to calculate the sum of static terms in the gains of minimal-order paths. The condition for relative degree to be degenerated is then: $\frac{-R}{L_1 L_2} + \frac{1}{k L_2 C_1} = 0$. Thus, when the system parameters satisfy the equality: $k C_1 R = L_1$, the relative degree is larger than two. However, this condition is not structural as it depends on numerical values of the parameters and a simple pertubation of the system parameter will cause the relative degree to reduce to the structural value predicted from expression (1).

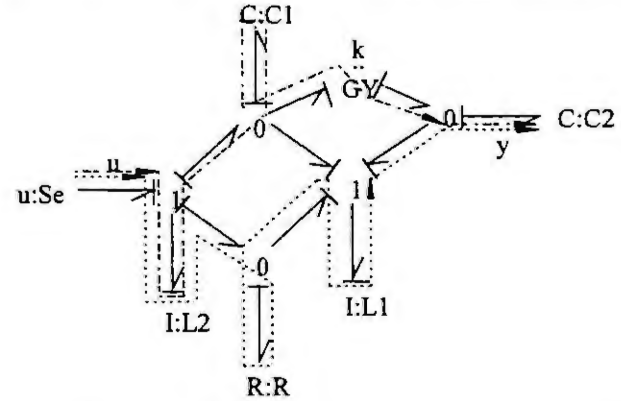
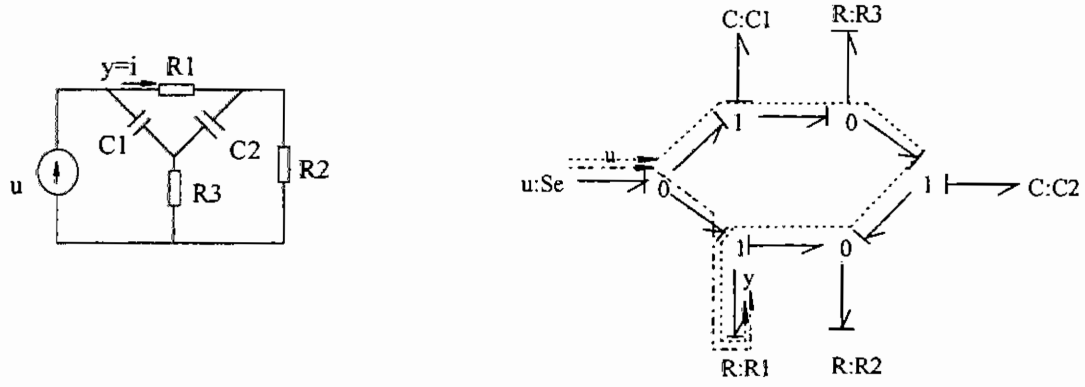


Fig.1 - Example of bond graph with double minimal-order paths

Example 2: Let us consider the bond graph of Fig.2 (b) which is the model of the electrical circuit of Fig.2 (a). According to the Rule 1, the relative degree of the given output y should be zero. But as there are two input-output zero-order causal paths, we have to study the sum of the gains of minimal-order input-output paths. In this case, that sum is null independently from the parameters of the components then the relative degree is structurally larger than zero. To find the structural relative degree, we have to consider paths of higher order and the output structural relative degree is then one. Note that in this last example, to find a one-order causal path, one has to considered not only elementary paths but also the paths containing causal loops.



(a) (b)

Fig.2 - Electrical circuit (a) and its corresponding bond graph (b)

To summarize this section, the relative degrees from the bond graph model is given by (1) when the minimal-order causal path between input components and the considered output variable is unique. Otherwise, one has to examine the sum of the gains of minimal-order input-output causal paths in order to study the conditions in which the relative degree is degenerated.

2. Stability of zero dynamics from the direct bond graph

The zero dynamics of a system are the internal dynamics when the input is chosen so that the output and its successive time-derivatives are kept identically zero. Defining a power line as a serial connection of bonds and junctions, an extensive study of the zero dynamics of systems with single input-output power line is presented in [6] and the following result which gives a sufficient condition for a system to be minimum-phase (i.e. with stable zero dynamics) is established:

Rule 2: The zero dynamics of a linear SISO system are stable or marginally stable if there is a single power line from the input source to the output variable. Moreover, each subsystem off this single power line indepently determines a set of zeros.

An additional result concerning the parameters on which the values of the zeros depend is also given.

Rule 3: Given an input-output pair, the zeros of a linear system are independent of the parameters of the invariant elements i.e I, C or R elements which appear on each path from the input to the output variable.

Remark 1: In the above rule 1 it is supposed that there exist no element with negative parameters as it may occur in bond graph obtained from linearization around some reference state.

The above rules does not involve systems with multiple power lines between input and output variables. As the zeros of a system can be studied as the poles of the inverse system, in the next section, a procedure for construction of inverse bond graph is recalled and then we will show how by analysing causal loops in the inverse bond graph, a general approach for analysing the stability of the zero dynamics can be carried out in some simple cases including multiple input-output power lines system and we will illustrate the method on an example of nonminimum phase system.

3. Construction of the inverse bond graph

In the bond graph representation, a causality assignment is attached to a computational scheme and causal manipulations are analogue to algebraic and differential operations on a mathematical model. Bicausal bonds enlarges the computational possibilities and allows to represent the inverse system in a convenient way [3]. An extended causality assignment procedure to represent the inverse bond graph model of linear SISO systems was given in [2] and the main points of that procedure are presented below:

Procedure 1 (for system inversion):

i) Determine the minimal-order path between the input variable and the output variable on the bond graph in integral causality.

ii) On the acausal bond graph, replace the source associated to the input variable by a SS element and connect a SS element to the output variable.

iii) On the output SS element, assign the flow source / effort source causality and propagate the bicausal information toward the input SS element along the shortest path determined in step 1. Immediately extend the causal implications throughout the bond graph as far as possible.

iv) Choose any energy storage elements (I or C) without causality, and assign its preferred integral causality. Propagate the causal information as far as possible. Repeat this step until all I or C elements are causally completed.

v) Choose any unassigned R element and assign to it an arbitrary causality. Propagate the causal information as far as possible. Repeat this step until the bond graph is causally completed.

Remark 2: SS element is source-sensor element introduced in [3] to extend the concept of sources and detector in the context of bicausality.

4. Zero dynamics analysis from the inverse bond graph

Depending on whether the system is single or multiple input-output power lines, two cases naturally appears when applying step iii) of procedure 1. If there is a single power line between the input and the output variables, then there is only one possibility for bicausal propagation from the output SS element toward the input SS element (Fig.3-4). A choice for bicausal propagation appears in the case of multiple input-output power lines. In the following, according to the number of power lines between the input and the output variable we will use the expression single power line or multiple power lines systems. Thus the same system can be called single power line or multiple power lines depending on the chosen input-output pair. We then distiguish two cases for our study.

4.1 The case of single power line systems:

The following development shows that the above rules stated in [5] can easily be interpreted using the extended causal bond graph of the inverse system.

Let us considered the general form of systems with single power line between input and output (Fig.4).

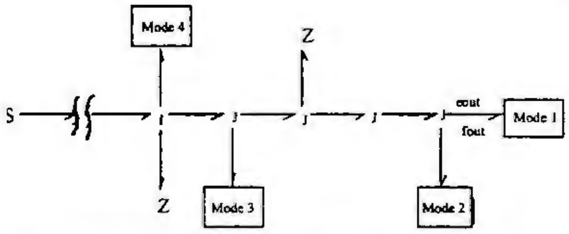


Fig.3 Single power line system general structure with J ∈ {1, 0, TF ,GY}; Z is an invariant element and "Mode i" is the passive subsystem i

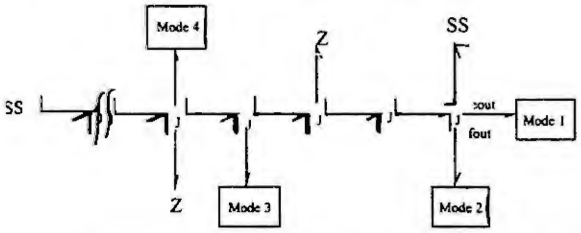


Fig.4 Propagation of bicausality along the power line

The propagation of bicausality from the output SS element to the input SS element leads to the intermediate bond graph shown in Fig.4 as bicausality can not be propagated toward subsystems "Mode i" or Z-elements [2].

The causal completion of the inverse bond graph of Fig.4 leads to the causal configurations of Fig.5 according the type of the J junction in the power line. At each junction J, causality is imposed by the bicausal bond in the power line. A graphical reasoning shows that there is no loop including elements of different subsystems or an element of a subsystem and an invariant element. Then, for i ≠ j, the set of loops of the subsystem i are disjoint from the set of loops of the subsystem j.

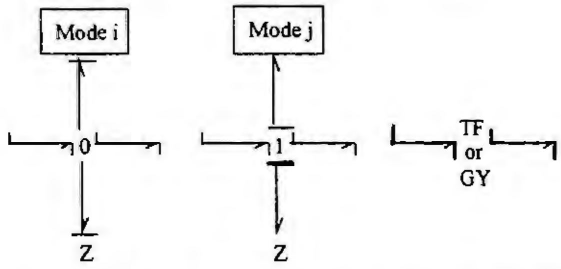


Fig.5 Causal configurations of J junction of the power line in the inverse bond graph

We can then conclude from Mason loop rule applied to bond graph [1] that each subsystem determines a set of zeros and as these systems are passive systems, the zeros are stable or at least marginally stable if they contain no dissipative component. Moreover, a graphical reasoning proves that in the inverse bond graph, invariant elements Z are either R elements or I/C elements in derivative causality [2]. They then have no effect on the zeros dynamics.

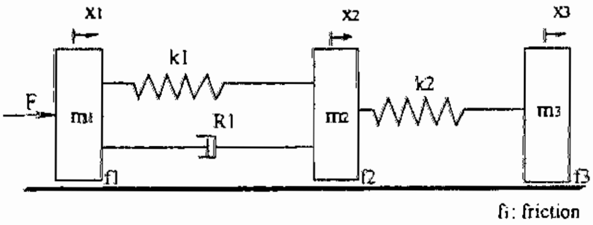


Fig.6 Example of mechanical system

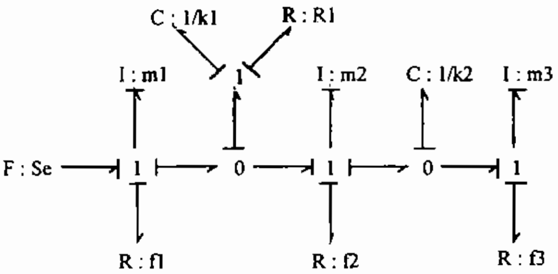


Fig.7 Causal bond graph of the system of fig.4

Example 3: Let us consider the mechanical system of Fig.6. Its causal bond graph is represented on Fig.7 and Fig.8 represents the inverse bond graph when the output is $v_2=dx_2/dt$. By examining that inverse bond graph, we conclude that there are two sets of zeros independently defined by the parameters of each subsystem 1 and 2. The zeros dynamics are independent of m_1, m_2, f_1 and f_2 as I-elements with parameters m_1 and m_2 are invariant elements according to the definition in rule 3 while R-elements with parameters f_1 and f_2 are dissipative subsystems with no effect on the system dynamics.

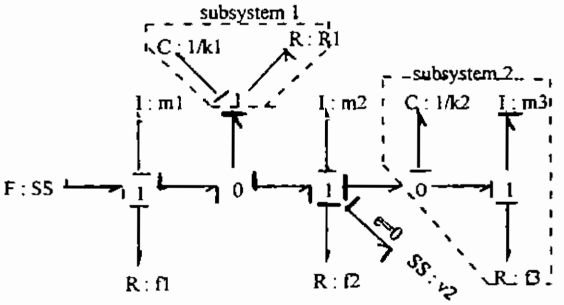


Fig.8 Inverse bond graph of Fig.5 with v2 as output

4.2 The case of multiple power lines systems

When for a chosen input-output pair there are multiple input-output power lines in the system, the rules stated above and the decomposition in subsystems are not straigthforward. In the inverse bond graph of the system, the bicausal bonds may create causal cycles or causal meshes [5] with positive loop gain. Examining the loop gains in the inverse bond graph allows to study the stability of the zero dynamics and in some cases to detect structural nonminimum-phase systems from the positiveness of an isolated loop gain.

To illustrate the method, we considered again an example similar to the one presented in [6] and we show how the instability of the zero dynamics is graphically detected.

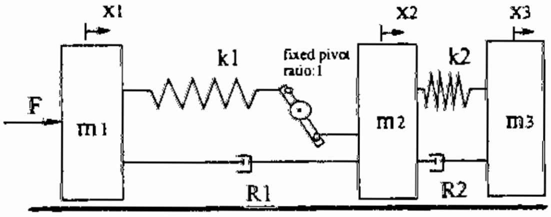


Fig.9 Mechanical system with a lever

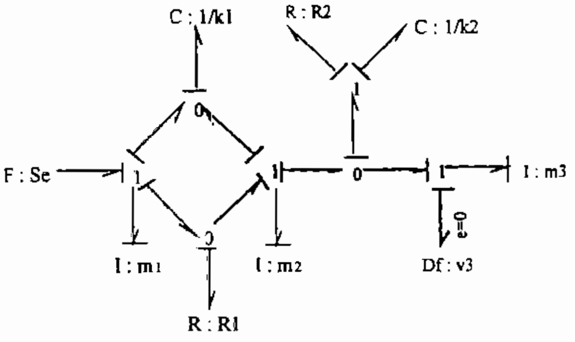


Fig.10 Causal bond graph of the system of Fig.9

Example 4:

This illustrative example of Fig.9 presents a particular case of a nonminimum-phase system. The causal bond graph (Fig.10) shows that there are two power lines between the input F and the output v_3. Application of procedure 1 gives the inverse bond graph of Fig.11 on which we see that there are two zeros independently defined by the subsystem 1 (parameters C_2 and k_2) and the isolated causal mesh which has the positive gain $+k_1/R_1s$. The system is then nonminimum-phase as it has a positive zero $z = +k_1/R_1$.

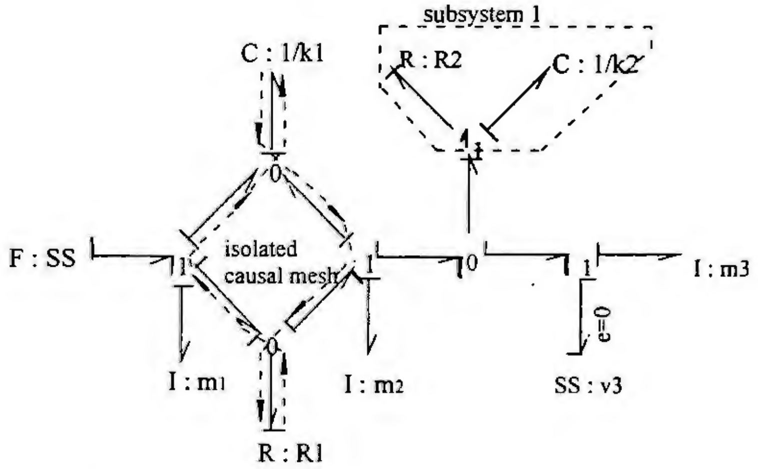


Fig.11 Inverse bond graph of the system of Fig.9 with v_3 as output

***Remark** 3*: Note that if v_1 the velocity of the mass m_1 is taken as output, then the system will be single power line and the rules 1 and 2 can be applied.

Conclusion

A graphical method based on bond graph representation for determining the output relative degrees and analysing the zero dynamics is proposed in this paper. The results presented in [2] are reinterpreted in a more general approach using the bicausality concept and the inverse bond graph model. It is shown that from the structural analysis of the system, some properties concerning the relative degree or the stability of the zeros dynamics and not depending on the system parameters can be deduced by analysing causal paths in the direct bond graph or causal loops in the inverse bond graph.

References

[1] Brown, F. T., Direct Application of the loop rule to Bond Graphs. In ASME Journal of Dynamics Systems, Measurements and Control, Vol. 21, No. 3 (1972), pp. 253-261.

[2] Fotsu Ngwompo, R. Scavarda, S., and Thomasset, D., Inversion of linear time-invariant SISO systems modelled by bond graph. In: Journal of the Franklin Institute, Vol.333(B), N°.2 (1996) pp.157-174.

[3] Gawthrop, P.J., Bicausal Bond Graphs. In: Proceedings of the 1995 International Conference on Bond Graph Modelling and Simulation: ICBGM'95, San Diego, 1995, pp. 83-88.

[4] Karnopp, D.C., Margolis, D.L. and Rosenberg, R.C. System Dynamics - A unified approach, Wiley, 2nd edition, 1990.

[5] Van Dijk, J. and Breedveld, P.C., Simulation of system models containing zero-order causal paths - I. Classification of zero-order causal paths. In Journal of the Franklin Institute, Vol.328, No.5/6 (1991), pp.959-979.

[6] Wu, S. T. and Youcef-Toumi, K., On Relative degrees and zero dynamics from physical systems modeling. In ASME Journal of Dynamics Systems, Measurements and Control, Vol. 117, June 1995, pp. 205-217,.

BOND-GRAPH MODELLING FOR GEOMETRIC DECOUPLING CONTROL

J.M. Bertrand, C. Sueur and G. Dauphin-Tanguy
L.A.I.L., U.R.A. C.N.R.S. D1440, Ecole Centrale de Lille, B.P. 48
59651 Villeneuve d'Ascq cedex, France
Tel: (33) 3.20.33.54.14, fax: (33) 3.20.33.54.18, email: bertrand@lails1.ec-lille.fr

Abstract. In this paper, the structure of linear square bond-graph models is investigated in a particular control goal: input-output decoupling by regular static state feedback. The aim is to identify, on the bond-graph model describing the system, the elements involved in major properties of the control solution. Model decouplability and closed loop fixed modes are thus graphically characterized. Geometric tools and bond-graph methodology are also shown to enable, in a graphical manner, the symbolic computation of the associated decoupling contol law.

1. Introduction

The bond-graph is an appreciated tool for physical systems modelling. Based on power flows representation, it enables to describe the system through energy storage and dissipating elements [5]. Behaviour of the system is accurately simulated by the derived mathematical model, whose physical meaning is kept clear. In a control objective, the structure of the chosen model is also of greatest importance: closed loop requirements may depend on groups of elements of the open loop model. Refining these parts of the model would enable to meet the control goals more efficiently, provided that these refinements also improve the model accuracy. In an input-output decoupling objective, the aim of this work is to identify, on the bond-graph model describing the system, the elements involved in major properties of the control solution.

Suitable tools for both structural analysis and synthesis of input-output decoupling control laws are defined by the geometric approach [10]. In particular, many contributions have been brought about input-output decoupling by regular static state feedback, in which the structure of the open loop model is of greatest interest [6]. This structure specially enables to know whether the model is decouplable [3]. If so, some poles of the decoupled model are also shown to be independent of the control law, so-called fixed modes [4]. An unstable one would lead to an unstable decoupled model, making this control strategy unrealistic.

In this paper, graphical methods are first developed to locate, on the bond-graph model, strategic elements involved in the decoupling problem. Straight from any square linear invertible bond-graph model, causal paths concepts are used to express whether the model is decouplable by regular static state feedback. If so, the expressions of the associated fixed modes are determined. Symbolic computation and bond-graph methodology are then combined to synthesize the corresponding decoupling state feedback. An example is finally presented to detail these analysis and computation methods, particularly emphasizing their interest in a modelling point of view.

2. Basic concepts

In this section, basic tools for input-output decoupling are recalled. They aim to characterize whether a model is decouplable. If so, well known methods to compute decoupling state feedbacks are presented. Fixed modes for the considered decoupling solution are finally introduced.

Consider the dynamical linear time invariant model (Σ) described by equation (1). $\mathbf{x}(.) \in X \approx R^n$ denotes the state vector; $y() \in Y \approx R^p$ denotes the output vector; $\mathbf{u}(.) \in U \approx R^p$ denotes the control input vector and $A:X \rightarrow X$, $B:U \rightarrow X$, $C:X \rightarrow Y$ are linear maps. Let $\mathbf{h}_i$ denote the ith row of a matrix $\mathbf{H}$ and $\mathbf{H}^i$ its ith column. Let $\mathbf{I}_n$ be the $n \times n$ identity matrix and $\mathbf{X}^i$ its ith column.

$$(\Sigma) \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \end{cases} \tag{1}$$

In the following, (Σ) is supposed to be square invertible, controllable and observable. Let us call regular static state feedback any control law $\mathbf{u}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}\mathbf{v}(t)$ with $\mathbf{G}$ invertible, noted rssf in the next [4]. Necessary and sufficient conditions are now recalled for (Σ) to be decouplable by rssf.

Decouplability of (Σ) is determined thanks to the orders of its infinite zeros. Two kinds of infinite zeros may describe the model (Σ) : the row ones and the global ones.

Definition 1: For each $j=1,\ldots,p$, the j^{th} row infinite zero order of (Σ) is the smallest integer n_j verifying $\mathbf{c}_j\mathbf{A}^k\mathbf{B}=\mathbf{0}$, $k<n_j-1$, and $\mathbf{c}_j\mathbf{A}^{n_j-1}\mathbf{B}\neq\mathbf{0}$.

Definition 2: Let G(s) be the transfer matrix of (Σ). Let $J_1(s)$ and $J_2(s)$ be bicausal matrices. The orders of the global infinite zeros of (Σ) are the increasingly ordered positive integers $\{n'_1,\ldots,n'_p\}$ such that:

$$\mathbf{G(s)}=\left\{\mathbf{J}_1\mathbf{(s)}\,.\,\mathbf{diag}\left(\mathbf{s}^{-n'_1},\ldots,\mathbf{s}^{-n'_p}\right).\,\mathbf{J}_2\mathbf{(s)}\right\} \tag{2}$$

The row infinite zero orders of (Σ) also enable to define the decoupling matrix Ω.

Definition 3: The decoupling matrix is the matrix Ω defined as in equation (3).

$$\Omega=\begin{bmatrix}\omega_1\\ \vdots\\ \omega_p\end{bmatrix},\quad \omega_j=\mathbf{c}_j\mathbf{A}^{n_j-1}\mathbf{B},\ \ j=1,\ldots,p \tag{3}$$

Property 1 then enables to express whether (Σ) is decouplable by rssf.

Property 1 [3]: The following assertions are equivalent:

(i) (Σ) is decouplable by rssf.
(ii) Ω is invertible.
(iii) $\{n_j\}=\{n'_j\}$.

Assume now that (Σ) is decouplable by rssf. Synthesis of decoupling rssf is suitably achieved thanks to the concept of (A,ImB)-invariant subspace introduced in the geometric approach [10]. Main results about such useful subspaces are now briefly recalled.

Definition 4 [10]: A subspace $\varepsilon\subset X$ is (A,ImB)-invariant iff there exists a map $F{:}X\rightarrow U$ satisfying $(A+BF)\varepsilon\subset\varepsilon$.

The supremal (A,ImB)-invariant subspace included in a subspace K is given as the limit of the Invariant Subspace Algorithm recalled in (4).

$$\begin{cases}\mathcal{V}^0=X\\ \mathcal{V}^\mu=K\cap A^{-1}\left(\operatorname{Im}B+\mathcal{V}^{\mu-1}\right),\mu\geq 1\end{cases} \tag{4}$$

It is called $\mathcal{V}^*$ - resp. $\mathcal{V}_j^*$ - when taking as subspace K the kernel of C - resp. the kernel of c_j. In the latter case, the row infinite zero orders enable to express $\mathcal{V}_j^*$, $j=1,\ldots,p$.

Property 2 [3]: For each $j=1,\ldots,p$, if n_j is the j^{th} row infinite zero order, then $\mathcal{V}_j^*=\bigcap_{k=0}^{k=n_j-1}\operatorname{Ker}\left(c_jA^k\right)$.

These invariant subspaces are finally shown to allow the computation of decoupling rssf.

Property 3 [2]: If the square model (Σ) is decouplable by rssf $\mathbf{u(t)}=\mathbf{F\,x(t)}+\mathbf{G\,v(t)}$, there exists, for each output $y_j(t)$, decoupling subspaces Q_j verifying $(A+BF)Q_j\subset Q_j$ such that:

$$\begin{cases}\mathbf{\Omega F}=\mathbf{[K]}\\ \mathbf{\Omega G}=\mathbf{diag}\left[\mathbf{g}_i\right]\end{cases} \tag{5}$$

$\mathbf{k}_j$ verifies $\mathbf{k}_j=\mathbf{h}_j-\mathbf{c}_j\mathbf{A}^{n_j}$ where $\mathbf{h}'_j$ is a linear combination of the spanning vectors of $Q_j^\perp$, $j=1,\ldots,p$. The coefficients of this combination are the degrees of freedom introduced in the control law through the j^{th} decoupling subspace. The choice of decoupling subspaces so determines the number of degrees of freedom available for pole assignment. For each choice, some poles of the decoupled model may be fixed, i.e. both independent of the control law and unobservable. Let us choose as decoupling subspaces the greatest ones : $Q_j=\mathcal{V}_j^*$, $j=1,\ldots,p$ [2]. In this case, property 4 allows to characterize the fixed modes straight from the open loop model.

Property 4 [2][4]: Taking as decoupling subspaces $\mathcal{V}_j^*$, $j = 1,\ldots,p$, leads to the design of a closed loop decoupled model whose fixed modes are the invariant zeros of the open loop model (Σ).

Zeros structure is thus of main interest in the input-output decoupling problem by rssf : the infinite zero orders enable to express whether a solution exists, whereas the invariant zeros characterize the fixed modes of the decoupled model. In the next section, causal paths concepts on the bond-graph model are shown to emphasize the symbolic expression of these zeros.

3. Zeros structure analysis: a bond-graph approach

Let us consider, in the following, bond-graph models with complete integral causality assignment. The state vector then deduced is $\mathbf{x}(t) = [\mathbf{p}_I(t) \quad \mathbf{q}_C(t)]^t$ whereas the state space equation is given by equation (1). Let DE_i be the i^{th} dynamical element with integral causality assignment, associated with the i^{th} state vector component $x_i(t)$. Let S be an input source and D an output detector - effort or flow. Some results about causal paths characterization are first recalled.

Property 5: The length of a causal path between DE_i and D or between S and D is the number of dynamical elements with integral causality assignment met on this path.

Property 6: When they contain at least one dynamical element, two causal paths are said to be different if they do not have any common dynamical element with integral causality assignment.

Both row and global infinite zero orders are now emphasized thanks to causal path concepts.

According to definition 1, for any right invertible model, the j^{th} infinite zero order n_j is the number of derivation of the j^{th} output variable $y_j(t)$ necessary to make appear explicitly at least one of the entries. In a bond graph manner, it may consequently be derived as stated by the following property.

Property 7 [8]: n_j is equal to the length of the shortest causal path between the j^{th} output detector D_j and all the input sources S_m, $m = 1,\ldots,p$.

As defined above, the row infinite zeros structure characterizes the relations between each separate output variable and all the control input variables. Conversely, the global infinite zeros structure describes independence properties between the global set of control input variables and the global set of output derivative variables. The j^{th} global infinite zero order is indeed the number of derivations of the j^{th} output variable $y_j(t)$ needed for the control input variables to appear explicitly in an independent way as they appear in the other output derivative variables. Different shortest input-output causal paths are now shown to emphasize the global infinite zero orders.

Property 8 [8]: The number of global infinite zeros is equal to the number of different input-output causal paths. Their orders $\{n'_1,\ldots,n'_p\}$ are computed as in equation (6), where L_k is the sum of the lengths of the k shortest input-output different causal paths.

$$\begin{aligned} n'_1 &= L_1, \\ n'_k &= L_k - L_{k-1}. \end{aligned} \tag{6}$$

One consequently derives in bond-graph terms the decouplability property expressed in property 1 (iii).

Property 9 [8]: The bond-graph model is decouplable by rssf iff the p different input-output causal paths are the shortest ones.

If there are several choices of p different input-output shortest causal paths, the gains of the shortest different causal paths from at least two output detectors to all the input sources may be proportional. For at least one output variable $y_j(t)$, the order of the global infinite zero order is then greater than the length of the shortest causal path from the associated output detector to the input sources. This allows to conclude that the bond-graph model is not decouplable by rssf.

The invariant zeros structure is now exhibited from the bond-graph model thanks to the same causal paths concepts.

Definition 5: The invariant zeros are the zeros of the system matrix $\mathbf{P}(s)$. They also are the roots of $\det[\mathbf{P}(s)]$ if the model is square.

Symbolic rules defined in the Grassmann algebra enable to compute $\det[\mathbf{P}(s)]$ in an easy symbolic manner [8]. The latter formalism also provides suitable tools for a graphical interpretation of the previous computation. The symbolic expressions of the invariant zeros of (Σ) are consequently determined as expressed by the following properties.

Property 10 [7][8]: Any decouplable bond-graph model with p inputs and p outputs contains at least one choice of p different input-output causal paths.

Property 11 [7]: The symbolic expression of $\det[\mathbf{P(s)}]$ may be derived straight from the bond-graph model of (Σ) as detailed by equation (7).

$$\det[P(s)] = \sum_q (-1)^{\sigma_q}.\left(\prod G_q\right).P_q \tag{7}$$

q is the number of possible choices for p different input-output causal paths. For each choice q: $\prod G_q$ is the product of the constant terms of the p different input-output causal paths, P_q is the characteristic polynomial of the bond-graph model obtained by removing from the initial one the p different input-output causal paths and σ_q is the number of permutations needed to express the outputs in the order of the initial output vector when the p different input-output causal paths are followed in the order of the initial input vector.

Consequently, considering the bond-graph model obtained by removing from the initial one each choice of p different input-output causal paths enables to determine the symbolic expressions of the invariant zeros of (Σ) [1]. As they also are the fixed modes for the considered decoupling strategy, the main interest of the previous methods is to identify, on the bond-graph model, the elements defining the fixed dynamics of the decoupled model.

By emphasizing infinite zero orders and invariant zeros, the study of particular input-output causal paths thus enables to locate, on the bond-graph model of (Σ), the elements involved in major properties related to the decoupling problem. Refinements of these parts of the bond-graph model may consequently be suggested, in order to achieve more efficiently the control objectives. The developments presented hereafter aim to show that such causal paths concepts also enable the symbolic computation of the decoupling rssf.

4. Bond-graph methodology for structural synthesis

As stated by property 3, two matrices are involved in the symbolic computation of the decoupling control law : the matrix **K** and the decoupling matrix Ω. The symbolic expressions of these matrices are now derived in a graphical manner.

According to property 4, the decoupling rssf we are interested in is associated with a set of decoupling subspaces composed of the subspaces $\mathcal{V}_j^*$, $j = 1,\dots,p$. Furthermore, property 3 states that the computation of this rssf requires the symbolic expressions of the subspaces $\mathcal{V}_j^{*\perp}$, $j = 1,\dots,p$. Thanks to property 2, the spanning vectors of these subspaces are those given in equation (8) [9].

$$\mathcal{V}_j^{*\perp} = \text{span}\left\{(c_j)^t,\cdots,\left(c_j A^{n_j-1}\right)^t\right\} \tag{8}$$

Consequently, for each $j = 1,\dots,p$, the row vectors needed to compute the jth row $\mathbf{k}_j$ of the matrix **K** are the row vectors $\mathbf{V}_k^j$ defined in equation (9), with $k = 0,\dots,n_j$.

$$\mathbf{V}_k^j = \left(\mathbf{c}_j\mathbf{A}^k\right) \tag{9}$$

State to output detector causal paths concepts are now shown to allow graphically the symbolic determination of these row vectors. Recall that DE_i is the ith dynamical element with integral causality assignment on the bond graph model of (Σ), associated with the ith state vector component $x_i(t)$. Also remind that $\mathbf{X}^i$ is the ith column of the identity matrix and D_j is the jth output detector. Let $G_k(DE_i,D_j)$ be the constant term of the gain of a causal path of length k between DE_i and D_j. Let $g(DE_i)$ be equal to $1/I$ for an I-element and $1/C$ for a C-element.

Property 12 [7]: $c_j A^k X^i = \sum G_k(DE_i,D_j) \times g(DE_i)$.

For each $j = 1,\dots,p$, the row vectors $\mathbf{V}_k^j$, $k = 0,\dots,n_j$ may thus be computed thanks to the gains of the causal paths of length k between the dynamical elements and the jth output detector D_j. The symbolic expression of the jth row $\mathbf{k}_j$ of the matrix **K** is then graphically deduced for each $j = 1,\dots,p$.

According to definition 3, the j^{th} row ω_j of the decoupling matrix Ω is defined as in equation (10).

$$\omega_j = \mathbf{c}_j \mathbf{A}^{n_j-1}\mathbf{B}\,,\ \ j=1,\ldots,p \tag{10}$$

The symbolic expression of each row matrix ω_j, $j=1,\ldots,p$, is now graphically emphasized thanks to input output causal paths concepts. Remind that S_m is the m^{th} input source on the bond-graph model of (Σ). Let $G_k(S_m,D_j)$ be the constant term of the gain of a causal path of length k between S_m and D_j.

Property 13 [7]: $c_j A^k B^m = \sum G_{k+1}(S_m,D_j)$.

For each $j=1,\ldots,p$, the row matrix ω_j may thus be computed thanks to the gains of the causal paths of length n_j between the input sources and the j^{th} output detector D_j.

In the next section, an example is finally presented that points out the interest of these analysis methods from a modelling point of view.

5. Example

Let us define, figure 1, a bond-graph model containing two input sources $\{E_1,E_2\}$, two output detectors $\{D_1,D_2\}$ and six dynamical elements, each with integral causality assignment. This model is invertible, controllable and observable [8].

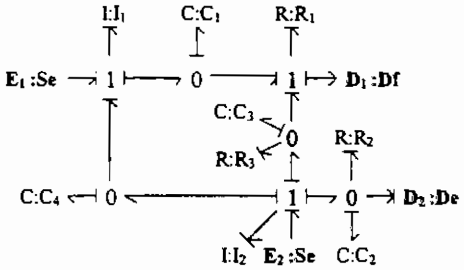


Figure 1. Bond-graph model.

First examine its infinite zeros structure. The shortest causal path from the output detector D_1 - resp. D_2 - to the input sources is $D_1 \rightarrow R_1 \rightarrow C_1 \rightarrow I_1 \rightarrow E_1$ - resp. $D_2 \rightarrow C_2 \rightarrow I_2 \rightarrow E_2$. According to properties 5 and 7, the row infinite zero orders are thus $n_1 = 2$ and $n_2 = 2$. These shortest input-output causal paths being different, their lengths also define the global infinite zero orders : $n'_1 = 2$ and $n'_2 = 2$. According to properties 1 and 9, the bond graph model is thus decouplable by rssf. As stated by property 4, the dynamic properties of the associated decoupled model rely on the open loop invariant zeros structure that is now investigated. Two couples of input output different causal paths are identified on the bond-graph model : $E_1 \rightarrow I_1 \rightarrow C_1 \rightarrow R_1 \rightarrow D_1$, $E_2 \rightarrow I_2 \rightarrow C_2 \rightarrow D_2$ and $E_1 \rightarrow I_1 \rightarrow C_1 \rightarrow R_1 \rightarrow C_3 \rightarrow R_1 \rightarrow D_1$, $E_2 \rightarrow I_2 \rightarrow C_2 \rightarrow D_2$. Remove from the bond-graph model the couple of shortest input-output causal paths. The remaining bond-graph model contains two dynamical elements, which means that the initial bond-graph model possesses two invariant zeros [8]. According to property 11, their symbolic expressions are derived from the study of the two couples of input-output different causal paths presented above. $s=0$ and $s=-1/R_3C_3$ are thus found to be the invariant zeros of the open loop model, also defining the fixed modes of the decoupling problem we are interested in.

The associated decoupling rssf may be computed as stated by properties 3 and 4. According to properties 12 and 13, causal paths gains between dynamical elements and output detectors or between input sources and output detectors enable to determine graphically the expressions of the needed matrices. Symbolic computations with MAPLE finally allow to determine the closed loop transfer matrix **T(s)** given in equation (11).

$$\mathbf{T(s)} = \begin{bmatrix} \mathbf{g_1/\left(s^2 - p_1^1 s - p_0^1\right)} & \mathbf{0} \\ \mathbf{0} & \mathbf{g_2/\left(s^2 - p_1^2 s - p_0^2\right)} \end{bmatrix} \tag{11}$$

This diagonal matrix is of fourth order : two closed loop modes have been made fixed, that are also the invariant zeros computed above. The four remaining modes may be assigned thanks to the degrees of freedom $p_0^1, p_1^1, p_0^2, p_1^2$, while the static gains requirements may be achieved by using the parameters g_1 and g_2.

As already determined, one of the fixed modes of this decoupled model is null. Focusing on the two couples of input-output different causal paths involved in the invariant zeros study enables to identify the physical origins of this null fixed mode. The bond-graph model obtained by removing alternately each of these couples is divided into two separated parts, one of those being always composed of the only C-element C_4. Property 11 consequently allows to conclude that the part of the open loop model responsible for closed loop unstability is composed of the only C-element C_4. Let us modify this part of the model by linking the C-element C_4 to some new R-element R_4. Structural properties such as controllability, observability and decouplability are not affected by this refinement [8]. Conversely, the invariant zeros structure has been modified. Property 11 indeed immediately enables to see that the previous null invariant zero becomes strictly stable, so leading to a set of invariant zeros composed of the two strictly stable elements $s = -1/R_4C_4$ and $s = -1/R_3C_3$. Introducing the R-element R_4 thus allows to design a stable decoupled model. Provided that the physical meaning of this element is made clear, this example illustrates how control objectives may be taken into account in a bond-graph modelling and design framework.

6. Conclusion

In this paper, the structure of linear square bond-graph models is investigated in a particular control goal : input output decoupling by regular static state feedback. The bond-graph methodology is first shown to enable a graphical characterization of the model decouplability. The symbolic expressions of the closed loop fixed modes, introduced by the non-interaction constraints, are also emphasized in a graphical manner. The associated decoupling control law is finally computed thanks to geometric concepts.

These structural analysis methods enable to identify, on the bond-graph model, the elements or groups of elements involved in major properties related to the decoupling problem. Such algorithms are implemented in a modelling and analysis software, so emphasizing bond-graph modelling efficiency in an integrated design framework.

In a next paper, further developments about input-output decoupling and bond-graph methodology will be presented. Causality concepts will be particularly investigated to characterize the decoupling solution from a stability point of view.

7. References

[1] Bertrand, J. M., Sueur, C. and Dauphin-Tanguy, G., Détermination Graphique des Modes Fixes de Systèmes Physiques Modélisés par Bond-Graph. In: Proc. AGI'96, Tours, 1996, 147 - 150.

[2] Claude, D., Automatique - Cours de Maîtrise. Service de Publications Paris Onze édition , Université Paris Sud, Orsay, 1992.

[3] Descusse, J. and Dion, J. M., On the Structure at Infinity of Linear Square Decoupled Systems. IEEE Transactions on Automatic Control, vol. AC-27, No. 4 (1982), 971 - 974.

[4] Icart, S., Lafay, J. F. and Malabre, M., A Unified Study of the Fixed Modes of Systems Decoupled via Regular Static State Feedback. Joint Conference on New Trends in System Theory, Genova, Birkhauser Boston, 1990, 425 - 432.

[5] Karnopp, D., Margolis, D. and Rosenberg, R., System Dynamics: A Unified Approach, second edition. John Wiley and Sons, Inc., New York, 1990.

[6] Martinez Garcia, J. C. and Malabre, M., The Row by Row Decoupling Problem with Stability: A Structural Approach. IEEE Transactions on Automatic Control, vol. 39, No. 12 (1994), 2457 - 2460.

[7] Rahmani, A., Etude Structurelle des Systèmes Linéaires par l'Approche Bond-Graph. Thèse de doctorat, Université Lille I, 1993.

[8] Sueur, C. and Dauphin-Tanguy, G., Poles and Zeros of Multivariable Linear Systems: a Bond-Graph Approach. In: Bond-graph for Engineers, (Eds.: Breedveld, P. C. and Dauphin-Tanguy, G.) Elsevier Science Publishers BV, IMACS, 1992, 211 - 228.

[9] Sueur, C. and Dauphin-Tanguy, G., Bond-Graph Modelling and Invariant Subspaces. In: Proc. IMACS Symposium on Mathematical Modelling MATHMOD, Vienna, 1994, 41 - 44.

[10] Wonham, W. M., Linear Multivariable Control: A Geometric Approach, third edition. Springer Verlag, New York, 1985.

STUDY OF CAUSAL LOOPS USING BOND GRAPHS AND THE SCATTERING FORMALISM

Abdelkader (EL) KAMEL and Geneviève DAUPHIN-TANGUY
Ecole Centrale de Lille, LAIL URA CNRS D1440
Cité Scientifique, BP 48, 59651 Villeneuve d'Ascq Cedex, France
Phone : +33. 3.20.33.54.11, Fax : +33. 3.20.33.54.18

Abstract. The purpose of this paper is to present a comparative study for a certain number of issues dealt with using the bond graph approach and reconsidered in the scattering formalism. Indeed, the problem of causal loops, solvable or algebraic, which may occur in a bond graph junction structure is considered in the scattering formalism and a structural analysis carried out in order to show how the wave-scattering approach remains as informative as the associated bond graph model.

Introduction

Bond graphs is a power tool for the modelling and especially the analysis of dynamical systems thanks to the structural properties which can be derived directly and graphically from the model [5, 10, 14]. Meanwhile, Paynter has advocated in [11, 12] the use of the scattering formalism as an alternative approach for physical systems modelling. Many results were then derived by a joint use of bond graphs and the scattering formalism.

In fact, in a previous work, a comparative study on the scattering formalism and bond graphs was devoted to system order investigation [7]. A particular interest was focused on the dynamical information derived either from the isolated system which behaves in free state just under initial conditions, or from the unbounded system which reacts with its environment at both the input and the output level. It has been shown then that the S-matrix gives a right indication on the eventual system order degeneracy since it goes through the different possible situations for the causality assignment on the associated bond graph model. Besides, the scattering matrix can be used in the design level since it intrinsically involves the organization of the different elements allowing to show up the appearance of equivalent dynamics which may simplify the whole model structure.

In the following paper, we propose to continue this comparative study and consider the classical problem of bond graph junction structures with loops. These loops may be solvable and have no effect on the model's behaviour, or algebraic and may cause simulation problems. A scattering point of view will be explored and a discussion undertaken to try to explain such phenomena.

The solvability of bond graph junction structures with loops was investigated by Rosenberg and Andry [13] and a computational criteria, based on the loop gain, was derived directly from the bond graph.

Background

Let us consider the following dynamical system represented in a special form where the process is a quadripole Q with different power waves, with respect to Q, at its two ends (Fig. 1).

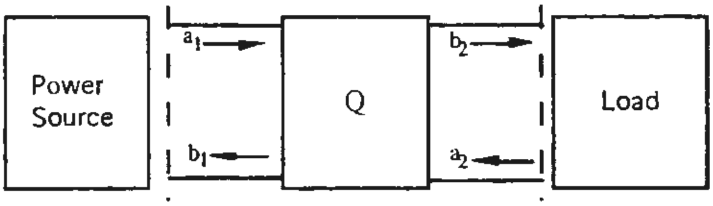


Fig. 1 : Physical system representation with the scattering variables

a_i $(i = 1,2)$ design the incident waves whereas b_i $(i = 1,2)$ design the reflection waves.

The incident and reflected powers, with respect to the quadripole Q, are directly linked to the incident and reflected power waves appearing on the flow diagram in Fig. 1, using relations (1). It is interesting to recall that the power delivered to the load by the source can be expressed as a difference between the incident power interring the load minus the reflected power going out from the load, $P_L = P_{r_2} - P_{i_2}$ [6].

$$P_{i_j} = \left|a_j\right|^2 \quad (j=1,2) \ \text{: Incident Power} \qquad (1)$$
$$P_{r_j} = \left|b_j\right|^2 \quad (j=1,2) \ \text{: Reflected Power}$$

The relation between the reflection power waves and the incident power waves is given by : $[b] = S[a]$, where S designs the scattering matrix (or S-matrix).

The general form of the S-matrix associated with a quadripole is [3, 6]

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} = \frac{\begin{bmatrix} b_n^{11} s^n + \cdots + b_0^{11} & b_n^{12} s^n + \cdots + b_0^{12} \\ b_n^{21} s^n + \cdots + b_0^{21} & b_n^{22} s^n + \cdots + b_0^{22} \end{bmatrix}}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_0} \qquad (2)$$

The S-matrix is not a transfer matrix but a special input-output representation described in the frequency domain thanks to Laplace operator s. Indeed, it is well known in microwave theory [9], where this matrix is commonly used, that the S-matrix of two quadripoles in cascade is not directly the product of the S-matrices of the two sub-systems.

In order to obtain the S-matrix, two methods, based on the associated bond graph model of the system, were introduced besides the classical approaches commonly used in microwave theory. The first one [2] is based on a causal bond graph and the determination of the gain of the causal paths and loops, whereas, the second [8] is based on an acausal bond graph and supposes a hierarchical organization of the model into an equivalent impedance and admittance.

The causal analysis of bond graph models especially with loops is the key point we propose to discuss in the following.

Causal loops between R-elements

Consider the acausal bond graph model in Fig. 2a. A structural analysis of this model is possible thanks to causality assignment. However, if we suppose that the kind of source, effort source or flow source, is not fixed a priori and if we notice that there are no physical restrictions which may lead to opt for one type of source to the detriment of the other, we find out three different possibilities for causal assignment on this bond graph (Fig. 2b, 2c, 2d).

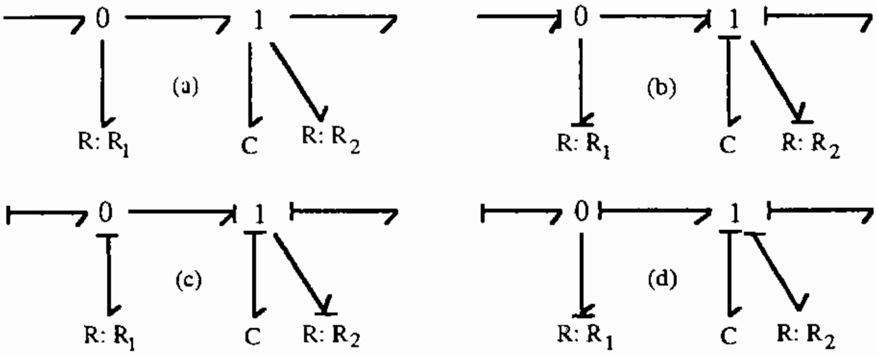


Fig. 2 : Acausal bond graph and different configurations for causality assignment

If we analyse the causal loops in the bond graph in Fig. 2b, we notice that there is only one causal loop between R-C elements. This loop relates the dynamic of the system equal to the gain of the loop, i.e., $-1/R_2Cs$. On the other hand, R_1 is not causally connected neither to R_2 nor to C. However, there is a causal path between the effort source and R_1 which relates the energy dissipation by the resistance. The global system behaves as if a new source including the previous dissipation is supplying an R-C circuit.

Now, consider the bond graph in Fig. 2c & 2d. In both cases, there is a causal loop between the R elements. In Fig. 2c, there is a causal loop between R-C elements besides the loop between the R elements; whereas in Fig. 2d, R_2 element is not directly connected to the C element but through the causal loop which links R_2 to R_1. Dynamically speaking, there is no relation between the system in Fig. 2b, whereas the dynamical behaviour of the bond graphs in Fig. 2c & 2d is the same.

It is interesting to note that the appearance of causality loops between R elements may induce simulation problems. That is why we try, at the design stage, to remove these loops by the introduction of an equivalent R element which represents this interaction. However, this procedure is not easy to carry out and may even appear not to be obvious.

Now, let us analyse this problem on the light of the scattering formalism. We bave recalled previously two methods to determine the S-matrix associated with a bond graph model. It is important to notice that the S-matrix associated with a system is unique, once the process has been defined as a quadripole for example, and whatever the applied method. Besides, we showed in [9] that the order of a system can be deduced from its S-matrix by observing the common denominator of S. That is wby in the following we will just consider this common denominator D(s).

When we determine the scattering matrix using the procedure developed in [8] we find :

$$D(s) = (1 + r_2 + r_2 r_1 + 2r_1)\tau s + (1 + r_1) \quad (3)$$

where r_2, r_1 and τ design the reduced variables, with respect to a normalization coefficient chosen depending on the context, respectively of R_2, R_1 and C. the r variables are homogenous to a constant while τ is bomogenous to a time constant associated with the C-element.

From expression (3), we can remark that the system behaves like a first order R-C filter. The interaction between the R elements is expressed in the coefficient in front of the time constant τ. We notice that there is not an easy expression which might characterize an eventual R equivalent. Indeed, it is interesting to recall a result established in [7] which shows that in the case we have a causal loop between two I elements or two C elements, that is one of the storage elements is in derivative causality, an equivalent element, with a time constant function of the two time constants, can be deduced. This equivalent element may replace the two storage elements with their causal loop by this unique equivalent element at a design stage. In this case, bowever, there is no evident indication for the existence of such an equivalent R-element.

If we consider the second method [2], we have to go through one of the causal bond graphs. However, the causal bond graphs including causal loops between R-elements lead to computational problems when determining the gains of the loops, which are the first step in order to obtain S, especially if the calculus is carried out by hand. Our purpose, is to include this methodology in the software ARCHER [1] which is able to determine such input-output matrices formally precisely thanks to the analysis of causal paths and loops.

Interpretation

The analysis of the scattering matrix allowed us to conclude that from a power propagation point of view, the system behaves like a first order R-C circuit without any order degeneracy or computational problem. In fact, the appearance of an algebraic loop between R-element is due to causality assignment in the bond graph model. However, even though the S-matrix intrinsically deals with causal relations and can explicitly include conservation laws, the cause to effect properties are not taken into consideration in the same way. Indeed, this is illustrated by the three bond graphs in Fig. 2b 2c & 2d where the dynamical behaviour of the system in Fig. 2b is completely different from that of the system in Fig. 2c or 2d which dynamical hehaviour is obviously the same. Nevertheless, when these systems are considered globally from a power propagation point of view, the behaviour is the same independently from the causality assignment. This discussion, brings to the fore a question concerning the relationship between the dynamics of power propagation compared with the proper dynamics of the system in automatic control sense. This study will be carried out in a future work.

Algebraic loops

Let us consider now the acausal bond graph model in Fig. 3(I).

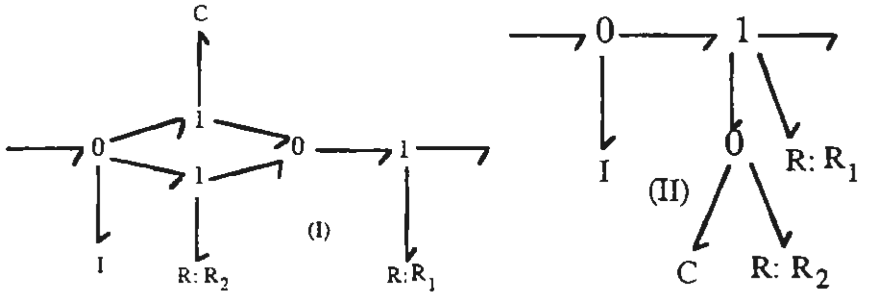


Fig. 3 : Acausal equivalent bond graphs

It is easy to show that the acausal bond graph models in Fig. 3(I) and 3(II) are equivalent [4].

In Fig. 4 we have represented different possibilities for causality assignment in the bond graph model of Fig. 3(I) given a causality specification at the input-output bonds.

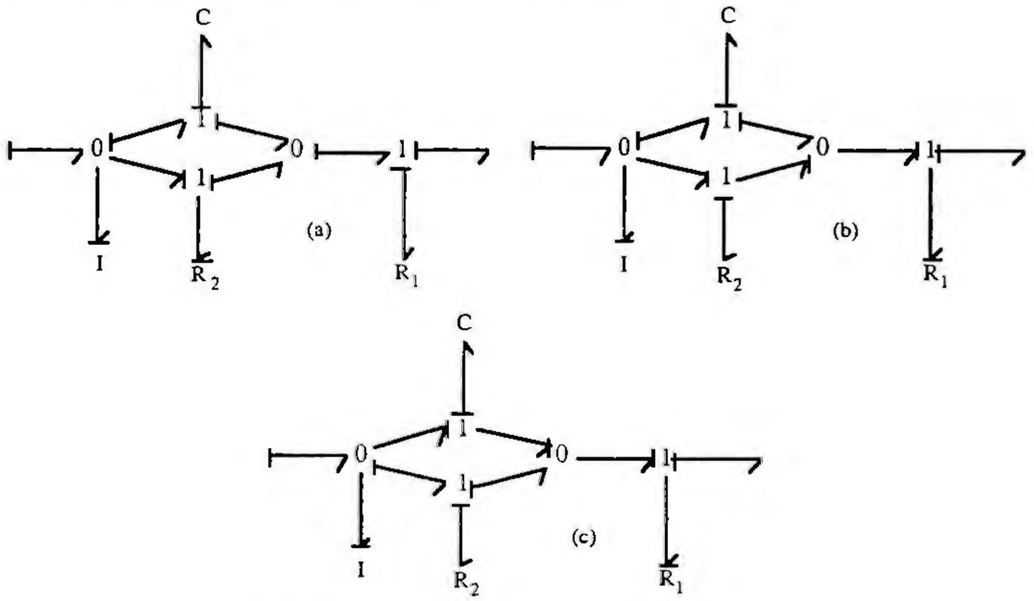


Fig. 4 : Different configurations for causality assignment

We notice that there are three configurations. The bond graph model in Fig. 4a has not any causality problem whereas those in Fig. 4b and 4c have a causality loop which gain is one. Rosenberg and Andry demonstrated in [13] how the appearance of such loops induce commutation problems since we have to get rid of them in order to determine, for example, the transfer function of the system.

On the other hand, the causality assignment in the equivalent bond graph model in Fig. 3(II) is as follows :

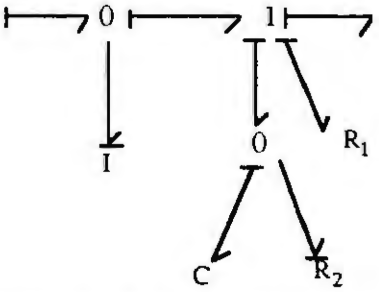


Fig. 5 : Causality assignment in the equivalent bond graph model

In this case, there is only one possibility for causality assignment, in view of the causalities at the input-output bonds. Besides, if we compare the causalities at the different elements, we notice that, in this case, they are the same than those of the bond graph in Fig. 4a. However, this property is not general since it happens that we obtain different configurations for causality assignment on the bond graph including a loop without being able to determine a priori their associated causally equivalent bond graph.

Let us analyse now this system's behaviour from a scattering point of view. If we use the method based on a causal bond graph model, we encounter the same problem described previously. However, the use of the acausal bond graph of Fig. 3(II) allows to obtain the S-matrix directly without any difficulty. We have then :

$$D(s) = (r_1 + 2)r_2\tau_I\tau_C s^2 + [(1 + r_1)r_2\tau_C + (r_1 + r_2 + 2)\tau_I]s + (r_1 + r_2 + 1) \quad (4)$$

We notice then that the system order is two and that the S-matrix is defined without the detection of any

DISTRIBUTED DEADLOCK AVOIDANCE CONTROL: PETRI NET APPROACH

R. Wójcik[1], K. Hasegawa[2], M. Sugisawa[2] and Z. Banaszak[3]
[1] Inst. of Engineering Cybernetics, Technical University of Wroclaw,
ul. Janiszewskiego 11/17, Wroclaw, POLAND,
[2] Dept. of Control Engineering, Toin University of Yokohama,
1614, Kurogane-cho, Aoba-ku, Yokohama, JAPAN,
[3] Dept. of Robotics and Software Engineering, Technical University
of Zielona Góra, ul. Podgórna 50, Zielona Góra, POLAND.

Abstract. This paper presents a new approach for distributed control of Flexible Manufacturing Systems (FMSs) that uses Petri nets formalism to model a system operation. The proposed method differs from previous works on modelling and control of such systems so that it constructs a feasible deadlock avoidance rules by exploiting the repetitive character of the material and data process flow. The proposed approach employs a Request/Allocation Graph (RAG) based concept of synchronization segments that provide the local rules for workflows control. Selection and allocation of segments in a RAG model of a FMS operation results in a real-time distributed deadlock avoidance procedure. The relevant, segment selection based conditions sufficient for deadlock avoidance are provided. A Petri net model of the high level control procedure is given. The Resource Allocation Procedure (RAP) for the low level controls is presented.

1. Introduction

In order to maximize the exploitation of a Flexible Manufacturing System (FMS) some resources are shared among different operations. Since a number of units of a resource is bounded, the processes being concurrently executed in the system, can compete for the resources that in turn can cause many problems related to the system control, e.g., deadlock occurrence.

A main objective of modern control systems is to increase a FMS productivity and reliability [1]. Such a requirement can be satisfied, however, by distributed control systems where decisions are undertaken by local, autonomous controllers linked each other through a computer network. This approach, in turn, imposes needs for a new distributed control oriented policies, in particular distributed deadlock avoidance procedures. So far developed procedures are aimed at computer/communication networks [5], [7], [8] and do not take into account a FMS specifics, e.g. data regarding resources allocation requirements. From another side, however, known methods provide only centralized solutions of the FMS deadlock avoidance problem [2], [3], [4], [9].

In this paper a distributed deadlock avoidance policy is presented. The policy requires determining of a set of synchronization segments in a RAG model of a FMS operation. It assumes that working processes require access to an unit of a specific resource to execute particular technological operation. The efficiency of the procedure is higher than the efficiency of preceding policies proposed in [2], [6] and can be as high as the efficiency of the methods presented in [3], [4], [9].

The following section gives basic definitions and formal model of a FMS operation. Section 3 presents the "process-in-segment suppression" deadlock avoidance policy and Section 4 includes a Petri net model of the communication scheme that let one to implement the policy in a distributed environment. Finally, Section 5 provides the efficiency estimation of the proposed policy and concluding remarks.

2. Basic notations and definitions

The resources

$R = \{r_i \mid i = 1,\dots,n\}$ - a set of resources, $c(r)$ - a number of units of a resource $r \in R$,

$BI = \{b_i \mid i = 1,\dots,L_b\}$ - a set of input buffers, $BO = \{e_i \mid i = 1,\dots,L_e\}$ - a set of output buffers.

We assume that $BI \cap BO = \varnothing$ and each buffer has unlimited number of instances.

Specification of the programs

$SPS = \{PS(i) \mid i = 1,\dots,v\}$ - a set of production programs to be executed in a FMS,

$PS(i)$ - the i-th program, $PS(i) = a_i(0), a_i(1),\dots,a_i(L_i), a_i(L_i+1)$

where $a_i(k)$, $k = 1,\dots,L_i$ - denotes the k-th operation in the program $PS(i)$,

$a_i(0)$, $a_i(L_i+1)$ - virtual operations that represent the input and output operations.

$OP = \{a_i(0), a_i(1),\dots,a_i(L_i), a_i(L_i+1) \mid i = 1,\dots,v\}$ - a set of the operations.

It is easy to note that $OP = \{o_{ik} \mid i = 1,\dots,v \quad \& \quad k = 0,\dots,L_i + 1\}$, where o_{ik} denotes the k-th operation in the program $PS(i)$.

A resource sequence corresponding to the program $PS(i)$

$RS(i) = r_i(0), r_i(1),\dots, r_i(L_i), r_i(L_i + 1) \quad \& \quad r_i(k) \neq r_i(k+1)$,

where $r_i(0) = b_i$ - the i-th input buffer, $r_i(L_i + 1) = e_i$ - the i-th output buffer,

$r_i(k)$ - a resource required by the operation $a_i(k)$, $k = 1,\dots,L_i$.

$SRS = \{RS(i) \mid i = 1,\dots,v\}$ - a set of the resource sequences corresponding to the given SPS.

The processes

$P = \{p_i \mid i = 1,\dots,m\}$ - a set of jobs (processes) to be executed in the system.

$P(i)$ - a set of the processes to be executed according to the program $PS(i)$.

A state of the system

Let $R(o_k)$ denotes a resource required by the operation $o_k \in OP$.

A state $S = [S(1),\dots,S(k),\dots,S(L)]$, where $S(k)$ is a number of processes that execute the operation $o_k \in OP \;\&\; R(o_k) \notin BI \cup BO$ and $L = |OP \setminus \bigcup_{i=1}^{v} a_i(0) \cup a_i(L_i + 1)|$ is a size of the state vector.

$S_0 = [0,\dots,0]$ - the initial state. $S_0[>$ - a space of all possible states.

For the given $SRS = \{RS(i) \mid i = 1,\dots,v\}$ we define **Request/Allocation Graph** (RAG).

Definition 1.

A graph $RAG = (V, E)$ where $V = R \cup BI \cup BO$ - a set of nodes and E - a set of edges, is said to be a Request/ Allocation Graph (RAG for short) if and only if the following condition holds

$(x,y) \in E \Leftrightarrow (\exists\, i \in \{1,\dots,v\})(\exists\, k \in \{0,\dots,L_i\})[\, x = r_i(k) \quad \& \quad y = r_i(k+1)]$,

where $r_i(k),\ r_i(k+1) \in RS(i)$. □

Let $P(r,S)$ be a set of processes that occupy units of a resource $r \in R$ in a state $S \in S_0[>$.

$n(r,S)$ - a number of processes that occupy units of a resource $r \in R$ in a state $S \in S_0[>$.

$f(r,S)$ - a number of a free instances of a resource $r \in R$ in a state $S \in S_0[>$.

$ZL = \{L_i \mid i = 1,\dots,w\}$ - a set of all cycles in the RAG corresponding to the SPS (cycle is defined as a sequence of nodes such that only the first and the last nodes are equal).

L_i - a set of nodes that define a cycle (e.g. $L_i = \{r_1, r_2\}$ defines the cycle (r_1, r_2, r_1)).

3. Process-in-segment suppression policy

Decomposition of the RAG model into segments

Let $RAG = (V, E)$ be a Request/Allocation Graph, where $V = R \cup BI \cup BO$ is a set of nodes. The set R can be subdivided into two disjoint subsets SR and UR such that

$r \in SR \Leftrightarrow (\exists a,b \in V)[(a,r) \in E \;\&\; (b,r) \in E]$,

$r \in UR \Leftrightarrow (\exists! a \in V)[(a,r) \in E]$,

$R = SR \cup UR \quad \& \quad SR \cap UR = \varnothing$.

The subset SR contains resources possessing at least two predecessors in a RAG model. Each resource belonging to the subset UR has only one predecessor (see Figure 1).

Consider the RAG model. Let $TH = (r, u_1,\dots,u_k, w)$ be a path in the RAG beginning from the resource $r \in SR$, where $u_1,\dots,u_k \in UR$. Of course, each path starting from $r \in SR$ either ends in $w \in SR$ or in $w \in BO$.

Figure 1. Resources: a) possessing at least two predecessors, b) possessing an unique a predecessor.

Let $TH = (r, u_1, \ldots, u_k, w)$ be a path. The sequence $SEG = (r, u_1, \ldots, u_k)$ is said to be a segment of the TH in the RAG model. Consider the segment $SEG = (r, u_1, \ldots, u_k)$. Its first element, i.e. r, and its last element, i.e. u_k, are said to be a head (HAP), and a tail (TAP), respectivelly. Thus, each segment can be seen as the following structure $SEG = (HAP, \ldots, TAP)$, in the particular case $SEG = (HAP)$.

In order to divide the RAG into segments one may use its own heuristics that uses the following rules.

Rule 1. Any two different segments may share some resources. One of the resources is their HAP.

Rule 2. For any two segments such that $SEG_i = (HAP_i)$ and $SEG_j = (HAP_j)$ the following holds $HAP_i \neq HAP_j$.

Rule 3. For any two segments such that $SEG_i = (HAP_i)$ and $SEG_j = (HAP_j, u_1, \ldots, u_k)$ the following holds $HAP_i \neq HAP_j$.

Rule 4. For any two segments $SEG_i \cap SEG_j \neq SEG_i$ and $SEG_i \cap SEG_j \neq SEG_j$.

Rule 5. Each $r \in SR$ must be a head of a segment.

Using the above introduced concept it can be easily noted that each circle of the RAG model contains at least one segment (see Figure 2), however, there may be the nodes not being related to any segment.

A capacity of a segment SEG is defined as a sum of all capacities regarding the resources included in the segment, i.e. $c(SEG) = \sum_{w \in SEG} c(w)$.

A number of processes that utilize resources of a segment SEG in a state $S \in S_0[>$ is defined as: $f(SEG, S) = \sum_{r \in SEG} n(r, S)$.

Therefore a number of free resource units of a segment SEG in a state $S \in S_0[>$ can be defined as: $f(SEG, S) = c(SEG) - n(SEG, S)$.

A length of a segment $SEG = (r_1, \ldots, r_k)$ is defined as a number of resources included in the segment, i.e., $|SEG| = k$.

Note, that the following propositions can be easily proven.

Proposition 1.

Let $SSEG$ be a set of segments in the RAG. The set $SSEG$ can be subdivided into the following disjoint subsets:

$SSEG_1 = \{SEG_k : |SEG_k| = 1\}$ - the subset of segments length of each one is equal to 1,

$SSEG_2 = \{SEG_k : |SEG_k| \geq 2\}$ - the subset of segments length of each one is equal or greater than 2,

such that $SSEG = SSEG_1 \cup SSEG_2$ & $SSEG_1 \cap SSEG_2 = \varnothing$.

Proposition 2.

Each circle in the RAG model contains either at least two segments belonging to the $SSEG_1$ or at least one segment belonging to the $SSEG_2$.

Conditions sufficient for deadlock avoidance

Let $f(SEG, S_n)$ be a number of free instances of the resources belonging to the segment $SEG \in SSEG$ in a state $S_n \in S_0[>$. Note, that the space $S_0[>$ can be seen as composed of SY_2, SY_1, SY_0 such that $S_0[>= SY_2 \cup SY_1 \cup SY_0$, $SY_2 \cap SY_1 \cap SY_0 = \varnothing$, and where:

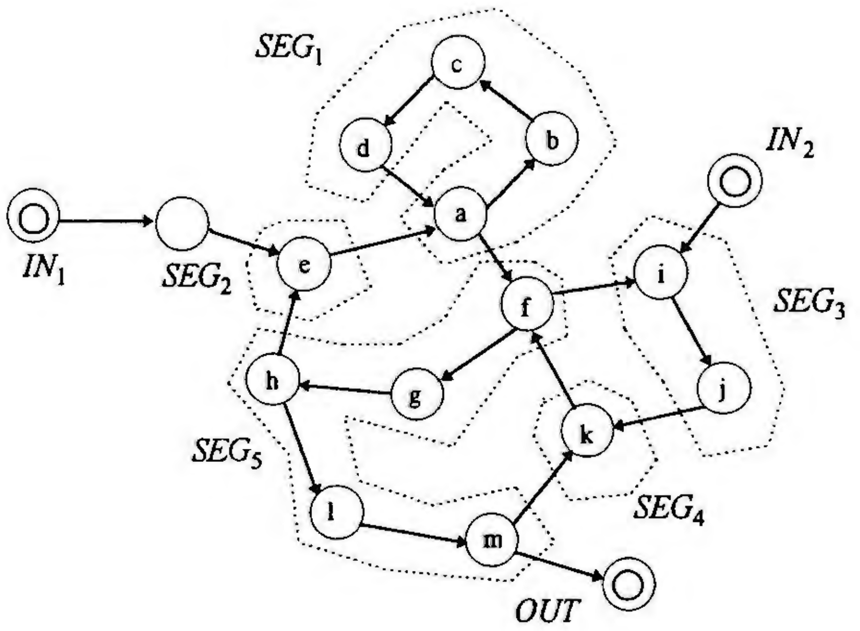


Figure 2. Possible segments: $SEG_1 = (a,b,c,d)$, $SEG_2 = (e)$, $SEG_3 = (i,j)$, $SEG_4 = (k)$, $SEG_5 = (f,g,h,l,m)$, IN_1, IN_2 - the input buffers, OUT - the output buffer.

- $S_n \in SY_2 \Leftrightarrow (\forall SEG \in SSEG_1: f(SEG,S_n) \geq 1) \quad \& \quad (\forall SEG \in SSEG_2: f(SEG,S_n) \geq 2)$

a set of states such that each cycle $L \in ZL$ has at least two free resource instances , e.g. $S_0 \in SY_2$,

- $S_n \in SY_1 \Leftrightarrow (\exists! SEG \in SSEG_1: f(SEG,S_n) = 0) \quad \& \quad (\forall SEG \in SSEG_2: f(SEG,S_n) \geq 2)$

OR

$(\forall SEG \in SSEG_1: f(SEG,S_n) \geq 1) \quad \& \quad (\exists SEG \in SSEG_2: f(SEG,S_n) = 1) \quad \&$

$\&$ $(\neg\exists SEG \in SSEG_2: f(SEG,S_n) = 0)$ a set of states such that there can exist a cycle $L \in ZL$ having only one free resource instance and there are no cycles having no free resource instances,

- $S_n \in SY_0 \Leftrightarrow (\exists SEG_i, SEG_j \in SSEG_1: f(SEG_i,S_n) = f(SEG_j,S_n) = 0)$

OR

$(\exists SEG \in SSEG_2: f(SEG,S_n) = 0)$ a set of states such that there can exist a cycle $L \in ZL$ having no free resource instances.

Let $SD_0[>$ denotes a set of deadlock states. Note, that the *necessary condition* for a state to be a deadlock is an existence of a cycle in the RAG having no free resource instances. Hence, $SD_0[> \subseteq SY_0$.

Each state $S_n \in SY_2 \cup SY_1$ is not a deadlock because every cycle has at least one free resource instance. Therefore, *sufficient condition* for a deadlock avoidance is to allocate resources to the processes in such that a state $S_{n+k} \in SY_2 \cup SY_1$, where $S_0 \in SY_2 \;\&\; k > 0$. In the following a suitable resource allocation procedure will be presented.

Let $St(SEG,S_n)$ be a variable that describes a status of a segment $SEG \in SSEG$ in a state $S_n \in S_0[>$.

Definition 2.

A status of a segment $SEG \in SSEG_1$ in a state $S_n \in S_0[>$ is defined as follows:

$St(SEG,S_n) = 2 \Leftrightarrow f(SEG,S_n) \geq 1 \quad \& \quad St(SEG,S_n) = 1 \Leftrightarrow f(SEG,S_n) = 0$. □

Definition 3.

A status of a segment $SEG \in SSEG_2$ in a state $S_n \in S_0[>$ is defined as follows:

$St(SEG,S_n) = 2 \Leftrightarrow f(SEG,S_n) \geq 2$ & $St(SEG,S_n) = 1 \Leftrightarrow f(SEG,S_n) = 1$. □

Let us consider a state $S_n \in SY_2$. There is $\forall SEG \in SSEG$: $St(SEG,S_n) = 2$. Assume, that an unit of a resource has been allocated to a process $p \in P$ and a state $S_{n+1} \in S_0[>$ has been reached. The following cases are possible:

1) $S_{n+1} \in SY_2$, i.e. the status of each segment is equal to 2,
2) $S_{n+1} \in SY_1$ and there exists the unique segment $SEG_i \in SSEG_1$ (i.e. $SEG_i = (HAP_i)$), such that $St(SEG_i,S_{n+1}) = 1$, and $\forall SEG \in SSEG \setminus \{SEG_i\}$: $St(SEG,S_{n+1}) = 2$,
3) $S_{n+1} \in SY_1$ and there exists the unique segment $SEG_i \in SSEG_2$, such that $St(SEG_i,S_{n+1}) = 1$, and $\forall SEG \in SSEG \setminus \{SEG_i\}$: $St(SEG,S_{n+1}) = 2$,
4) $S_{n+1} \in SY_1$ and there exist segments $SEG_i, SEG_j \in SSEG_2$, such that $SEG_i \cap SEG_j \neq \varnothing$ & & $HAP_i = HAP_j$ & $St(SEG_i,S_{n+1}) = St(SEG_j,S_{n+1}) = 1$, and $\forall SEG \in SSEG \setminus \{SEG_i, SEG_j\}$: $St(SEG,S_{n+1}) = 2$.

In the state $S_{n+1} \in SY_1$ there exists a segment $SEG_i = (HAP_i,\ldots,r,\ldots,TAP_i)$ having status $St(SEG_i,S_{n+1}) = 1$. Starting from the head HAP_i we can always find a resource $r \in R \cap SEG_i$, such that $n(r,S_{n+1}) > 0$ (i.e. there exists a process $p_1 \in P$ which is using the resource r). Thus, there exists a sequence $PRS = \langle p_1, p_2, \ldots, p_{w-1}, p_w \rangle$ such that (see Figure 3):

- a process $p_{k+1} \in PRS$ is using a unit of the resource which is required by a process $p_k \in PRS$,
- resources requested by the processes $p_1,\ldots,p_{w-1}$ have no free units,
- a unit of the resource requested by the process p_w is available.

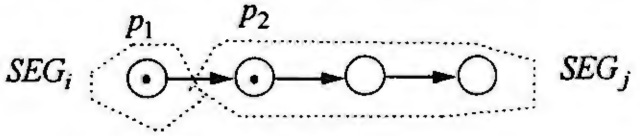


Figure 3. A sequence $PRS = \langle p_1, p_2 \rangle$.

It can be seen that allocating resources according to the $PRS^* = \langle p_w,\ldots,p_1 \rangle$ a state $S_{n+2} \in SY_2 \cup SY_1$ is reached. Thus, the cases 1) 2) 3) 4) can be considered, again.

Resource Allocation Procedure (RAP)

1. If $S_{n+1} \in SY_2$ then any resource request can be done.
2. If $S_{n+1} \in SY_1$ then find HAP_i such that $St(SEG_i,S_{n+1}) = 1$.
3. Starting from the head HAP_i find the first resource $r \in SEG_i$ which is used by a process $p \in P$.
4. Construct a sequence $PRS = \langle p_1,\ldots,p_w \rangle$, where $p_1 = p$.
5. Allocate resources according to the $PRS^* = \langle p_w,\ldots,p_1 \rangle$.
6. Let $S_{n+2} \in SY_2 \cup SY_1$ be a state reached after the allocation. Set $S_{n+1} := S_{n+2}$ and GOTO 1.

4. The communication scheme

We assume that each resource is managed by the relevant resource server. The processes send requests to the servers and wait for the resource donation. The allocations are realized according to the RAP which distributes the control between resource servers of the segment heads. The server of the head manages the resource allocations in the whole segment. Therefore, the server is responsible for creation of the PRS sequence in case when a status of the segment changes into 1 (note, that the status can be verified by the server, locally).

The servers can exchange messages. If a server receives a message called TOKEN it may allocate resources to the processes. In order to assure that only one head server will be making decisions at a time the servers exchange TOKEN according to a logical ring protocol. Thus, the server of the HAP_i sends TOKEN to the server of the HAP_{i+1}. However, if the allocation of the resources to the processes belonging to the PRS^* leads to a state S such that $St(SEG_k, S) = 1$, then TOKEN is send to the server of the HAP_k. A Petri net model of the communication scheme is shown in Figure 4.

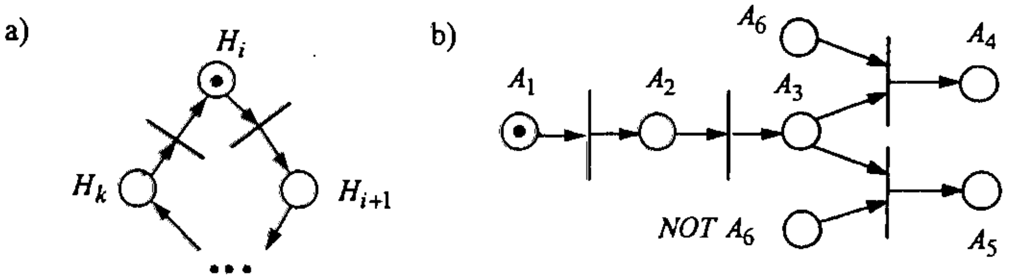


Figure 4. The communication scheme: a) the segments have always status equal to 2 (H_i - token is accessible by the server of the HAP_i); b) a segment can have status equal to 1 (A_1 - token is accessible by the server of the HAP_i, A_2 - resource allocating, A_3 - send token, A_4 - sending token to the server of the HAP_k, A_5 - sending token to the server of the HAP_{i+1}, A_6 - $\exists SEG_k : St(SEG_k, S) = 1$).

5. Conclusions

If each cycle $L \in ZL$ in the RAG consists at least two segments then the policy accepts more states than the method of zones [2]. However, if any of the RAG's cycles consists of only one segment, then it is possible to reach a state which is accepted by the method of zones and is not allowed by the RAP (e.g. consider RAG which contains two subgraphs such that each one consists of two cycles having only one common resource; the method of zones accepts states such that in each subgraph only the common resource of the cycles has one free instance; however, the RAP does not accept such states).

The presented policy is distributed oriented (the control is distributed between segments). However, it does not contain communication protocols (message exchange rules) that allow one to implement the RAP procedure in a network of autonomous controllers. This is still an open problem and should further be investigated.

References

1. Banaszak Z. (ed.), Modelling and control of FMS: Petri net approach, Wroclaw Technical University Press, Wroclaw, 1991.
2. Banaszak Z., Krogh B., Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows, IEEE Trans. on Robotics and Automation, 1990, Vol.6, No.6, 724-734.
3. Cho H., Kumaran T. K., Wysk R. A., Graph-theoretic deadlock detection and resolution for flexible manufacturing systems, IEEE Trans. on Robotics and Automation, 1995, Vol.11, No.3, 413-421.
4. Ezpelet J., Colom J. M., Martinex J., A Petri net based deadlock prevention policy for flexible manufacturing systems, IEEE Trans. on Robotics and Automation, 1995, Vol.11, No.2, 173-184.
5. Günther K.D., Prevention of deadlocks in packet-switched data transport systems, IEEE Trans. on Commun., 1981, COM-29, No.4, 512-524.
6. Hasegawa K., Sugisawa M., Banaszak Z. A., Ma Liqun, Graphical analysis and synthesis of deadlock avoidance in flexible manufacturing systems, In: Proc. of the 1st Int. Workshop on Manufacturing and Petri Nets, M. Silva, R. Valette, K. Takahashi (Eds.), June 25, 1996, Osaka, Japan, 161-176.
7. Peterson L., Silberschatz A., Operating systems concepts, Addison-Wesley, Amsterdam, 1983.
8. Raynal M., Helary J. M., Synchronization and control of distributed systems and programs, Wiley, 1990.
9. Wójcik R., Banaszak Z., Roszkowska E., Automation of self-recovery resource allocation procedures synthesis in FMS, In: Proc. of IFAC Workshop on CIM in Process and Manufacturing Industries - Espoo, Finland, Nov. 23-25, 1992, ed. by K. Leiviska, Pergamon Press Ltd., Oxford, 1993, 127-132.

ACTION LOGICAL CORRECTNESS PROVING

Kurt Lautenbach
University Koblenz-Landau
Institute for Software Technology
laut@informatik.uni-koblenz.de

Abstract. A logic is introduced whose basic formulae are elementary actions. The interpretations and models of this kind of logic are processes. The counterpart of the logical conclusion concept is a concept of fulfilling a formula (an action). A formula R (realization) fulfils a formula S (specification) iff every process of R is also a process of S. Direct and indirect proving correctness in the sense of a formula being carried out by another formula is based on this definition. Petri nets are used as a medium between this logic of actions and technical applications.

Introduction

The key idea of this paper is to develop a proving technique for a simple *logic of actions* (LA) that is based on Petri net theory. The expressions of LA, which will be called *modules*, denote actions. LA is comparable to propositional logic since it has no action variables. To every module a special Petri net (place/transition net) is assigned whose processes represent the module's semantics. This is comparable to the set of its models that is assigned to an expression in propositional logic.

The special form of the Petri nets allows to calculate processes with linear algebraic means which is important for two reasons. Firstly, because usually the performance of linear algebraic algorithms is acceptable. Secondly, because a "higher" logic of actions would demand a "higher" class of Petri nets. But then the difficulty to find suitable algorithms increases drastically. So, it seems hopeless to look for more than the algorithms for calculating linear invariants etc. In principle, however, the idea of direct and indirect proofs in LA is liftable to higher logics.

Syntax and Semantics of LA

In this section we will introduce the syntax and the semantics of LA. LA is only one of a lot of possible logics of actions. So, we will not show too many details according to the aim to only present the idea of Petri net based proving in LA.

The *alphabet* of LA consists of a finite set $A = \{a, b, c, \ldots, \bot\}$ of *atomic actions (atoms)* where $\bot$ represents the *impossible action* (falsum), the *operators* $^-$ (negation), $<$ (before), $>$ (after), $\wedge$ (and), $+$ (exclusive or), $\vee$ (non-exclusive or), $=$ (coincident), ^{n} (n-fold iteration), $^+$ (n-fold iteration for some $n \geq 1$), * (n-fold iteration for some $n \geq 0$), and the brackets (,).

The set $\mathcal{L}$ of *literals* consists of the set A of atoms, and the set of negated atoms.

The set $\mathcal{M}$ of *modules* (expressions of LA) is the smallest set with atoms are modules, for $M \in \mathcal{M}$ $\overline{M} \in \mathcal{M}$ and $\overline{\overline{M}} = M$ hold, for $M_1, M_2 \in \mathcal{M}$ $M_1 \circ M_2 \in \mathcal{M}$ holds where $\circ \in \{<, >, \wedge, +, \vee, =\}$, and for $M \in \mathcal{M}$ $\{M^n, M^+, M^*\} \subseteq \mathcal{M}$ holds.

For the operators we assume some rules: $<$ and $>$ are associative, but not commutative. $=, \wedge, \vee, +$ are associative and commutative. $\wedge, <, >, =$ are distributive over $+$.

To any module $M \in \mathcal{M}$ we assign a *Petri net (place/transition net)* $\mathcal{N}(M) = \{S, T, F, W, K\}$ where S and T denote the sets of *places* and *transitions*, respectively, F denotes the *flow relation*, $W : F \longrightarrow \{1\}$, and $K : S \longrightarrow \{\infty\}$ denote the restrictions on *arc weights* and *place capacities*.

Moreover, $\mathcal{N}(M)$ has exactly one *start transition* (without input places) and exactly one *goal transition* (without output places).

The set $\mathcal{P}(M)$ of *processes* of M is the set of reproductions of the empty marking in $\mathcal{N}(M)$ where the start transition and the goal transition both *occur exactly once*.

Figure 1 shows the net representations of some simple modules. Figure 2 shows the net representation of some non-atomic modules (The arc weight n in $\mathcal{N}(a^n)$ should be considered as an abbreviation).

The meaning of processes in LA resembles the meaning of models in propositional logic. Both are the key notion of the respective semantics.

So, a module $M \in \mathcal{M}$ is *contradictory* iff $\mathcal{P}(M) = \varnothing$, and is *satisfiable* iff $\mathcal{P}(M) \neq \varnothing$.

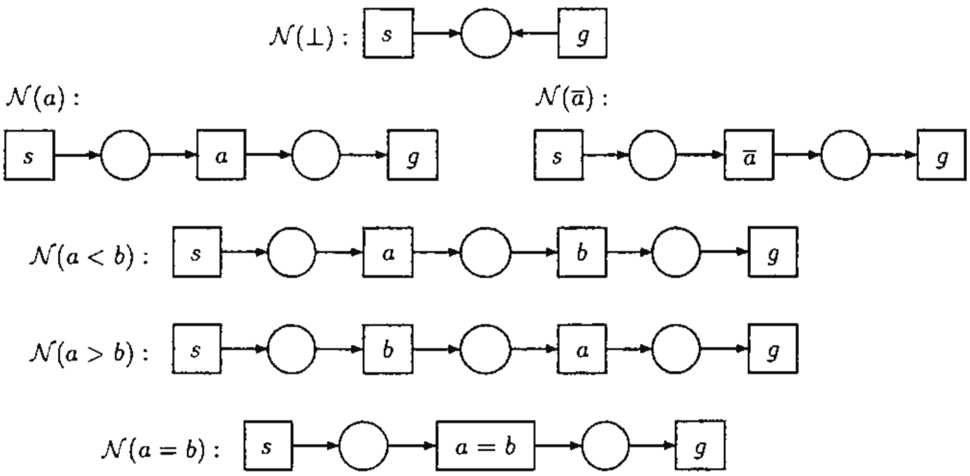


Figure 1

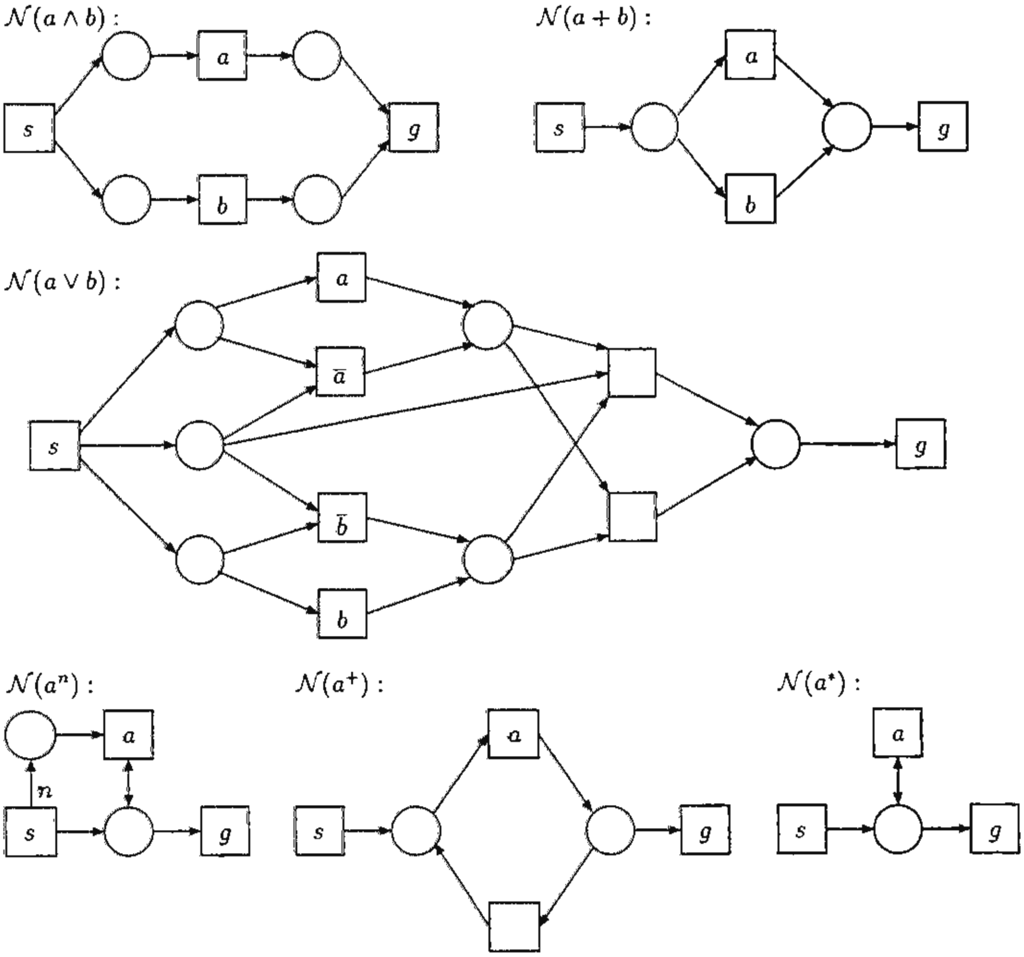


Figure 2

A module M is contradictory iff its net representation deadlocks. So, deadlocking is the Petri net paradigm of contradictions in LA. For example $\mathcal{N}(\perp)$ in figure 1 deadlocks in the sense that after exactly one occurrence of both transitions the empty marking has not been reproduced.

Proving in LA

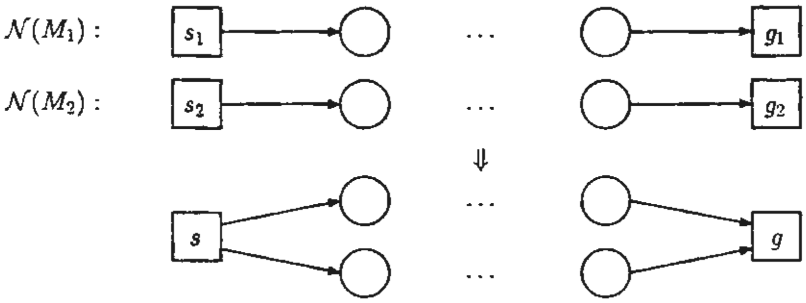
In this section we introduce the concept of a *logical consequence* in LA. We start with an algorithm for constructing $\mathcal{N}(M_1 \wedge M_2)$ from $\mathcal{N}(M_1)$ and $\mathcal{N}(M_2)$:

Input: $\mathcal{N}(M_1), \mathcal{N}(M_2)$

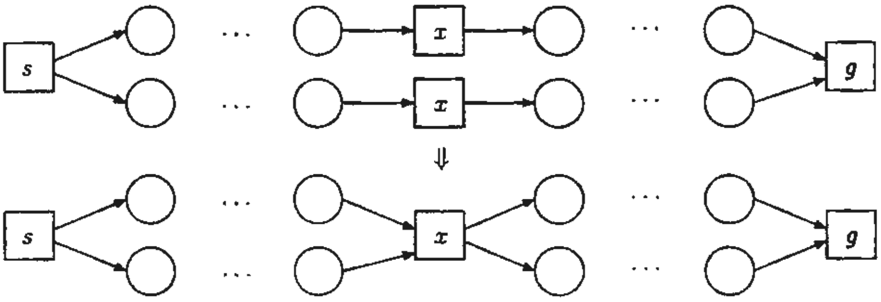
Output: $\mathcal{N}(M_1 \wedge M_2)$

Method:

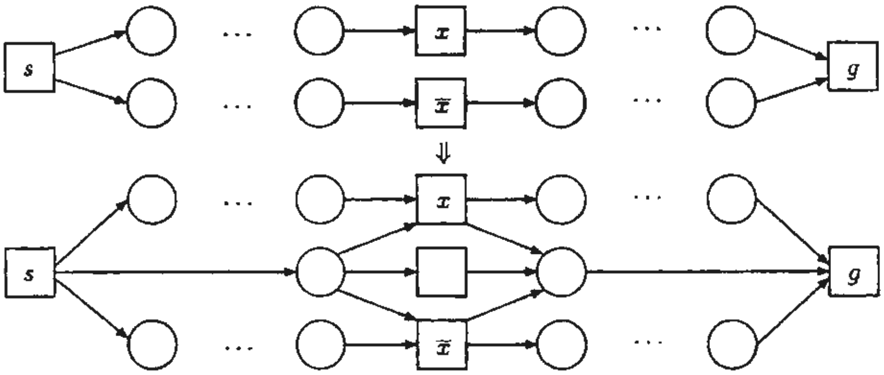
(1) Identification of both start transitions and identification of both goal transitions:



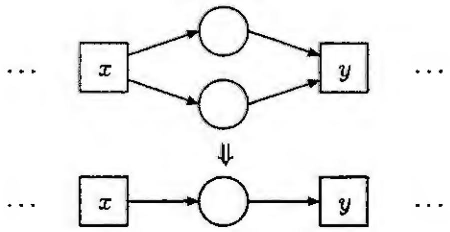
(2) Let be $x \in \mathcal{L}$ and let (without loss of generality) in $\mathcal{N}(M_1)$ and in $\mathcal{N}(M_2)$ exist only one x–labeled transition; then both x-labeled transitions are identified:



(3) Let be $x \in \mathcal{L}$ and let (without loss of generality) in $\mathcal{N}(M_1)$ exist only one x–labeled transition and in $\mathcal{N}(M_2)$ only one $\overline{x}$–labeled transition; then a mutual exclusion of x and $\overline{x}$ has to be guaranteed:



(4) Deletion of redundant places:



Two examples are shown in figures 3 and 4.

$\mathcal{N}((a+b) \wedge (b+c))$:

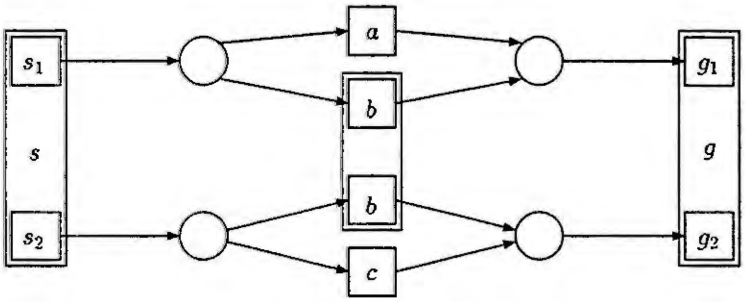


Figure 3

$\mathcal{N}((a+b) \wedge (\overline{b}+c))$:

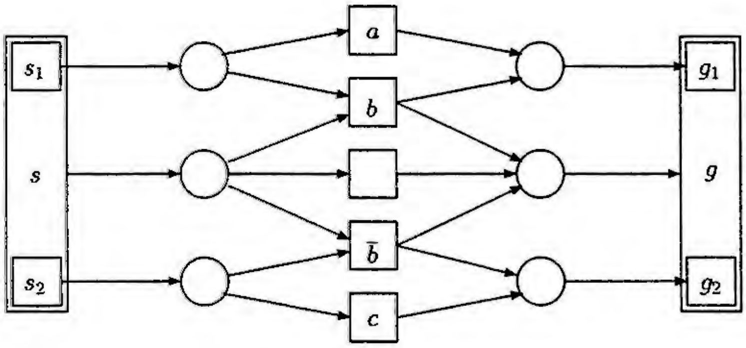


Figure 4

In propositional logic B is a logical consequence of A iff all models of A are models of B, too.

In LA, we try a similar approach. The aim is to express that a module R *fulfils* a module S if all processes of R are also processes of S. But in contrast to models, where a truth value is assigned to every atom, it is not usual to overload processes with mentioning the non–occurring actions. In other words, models are easily comparable – processes are not. Instead, we try to manage with numbers of processes.

Definition:
Let R and S be modules; R *fulfils* S ($R \twoheadrightarrow S$) iff $|\, \mathcal{P}(R \wedge S) \,| = |\, \mathcal{P}(R) \,|$.

If $\overline{S}$ is given instead of S the following Lemma is important for indirect proving.

Lemma:
Let R and S be modules; $R \twoheadrightarrow S$ iff $|\, \mathcal{P}(R \wedge \overline{S}) \,| = 0$.

Proof:
See [Fide93]. □

So, we have the possibility of direct and indirect proving.

(1) Direct Proofs:
$R \twoheadrightarrow S$ iff $|\mathcal{P}(R \wedge S)| = |\mathcal{P}(R)|$
$R \not\twoheadrightarrow S$ iff $|\mathcal{P}(R \wedge S)| < |\mathcal{P}(R)|$

(2) Indirect Proofs:
$R \twoheadrightarrow S$ iff $|\mathcal{P}(R \wedge \overline{S})| = 0$
$R \not\twoheadrightarrow S$ iff $|\mathcal{P}(R \wedge \overline{S})| > 0$

Example:
$M_1 = ((a < (b + c)) \wedge d)$, $M_2 = (\overline{a} \vee c)$, and $\overline{M}_2 = (a \wedge \overline{c})$ are modules. A direct and an indirect proof is given for $M_1 \not\twoheadrightarrow M_2$. Figure 5 shows the net representations of M_1, M_2, and $\overline{M}_2$.

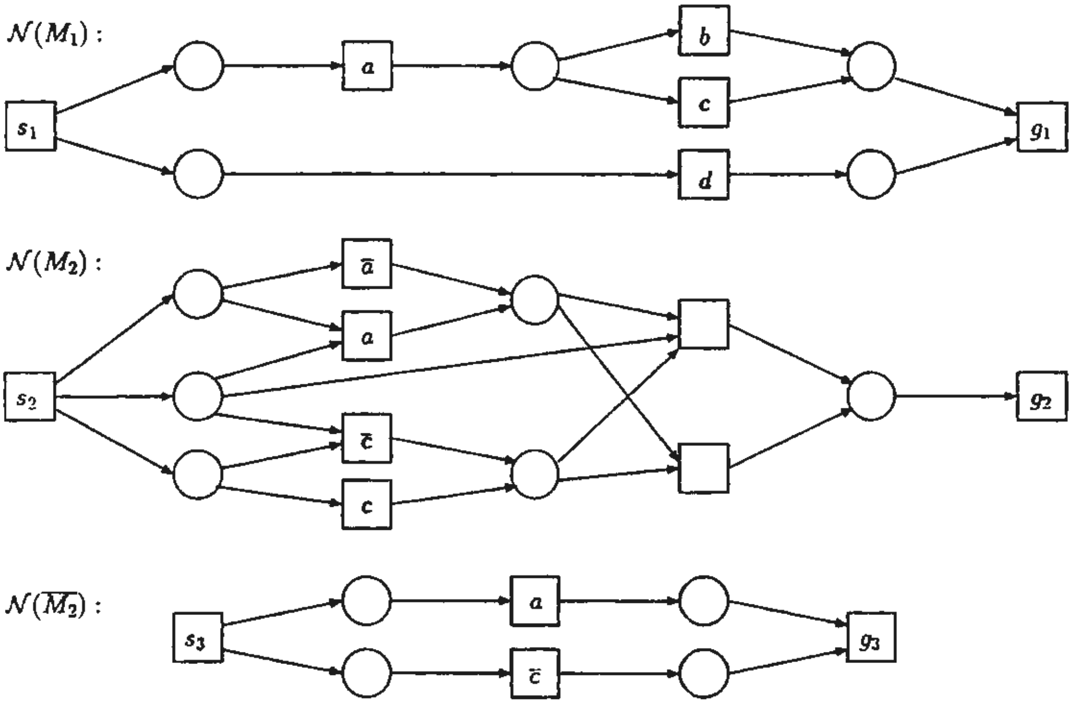


Figure 5

(1) Direct Proof:
We have to compare $\mathcal{P}(M_1)$ and $\mathcal{P}(M_1 \wedge M_2)$.

In an easily understandable process notation we write $\mathcal{P}(M_1) = \{((a < b) \wedge d), ((a < c) \wedge d)\}$.

This corresponds exactly to the two possibilities to reproduce the empty marking in $\mathcal{N}(M_1)$.

The net of figure 6 is $\mathcal{N}(M_1 \wedge M_2)$ with only one possibility to reproduce the empty marking. Precisely $\mathcal{P}(M_1 \wedge M_2) = \{((a < c) \wedge d)\}$.

The result $|\mathcal{P}(M_1 \wedge M_2)| = 1 < |\mathcal{P}(M_1)| = 2$ implies that $M_1 \not\twoheadrightarrow M_2$ holds.

The process $((a < b) \wedge d)$ of M_1 is no process of $M_1 \wedge M_2$ since it does not belong to M_2 whose set of processes is $\mathcal{P}(M_2) = \{(\overline{a} \wedge \overline{c}), (\overline{a} \wedge c), (a \wedge c)\}$. □

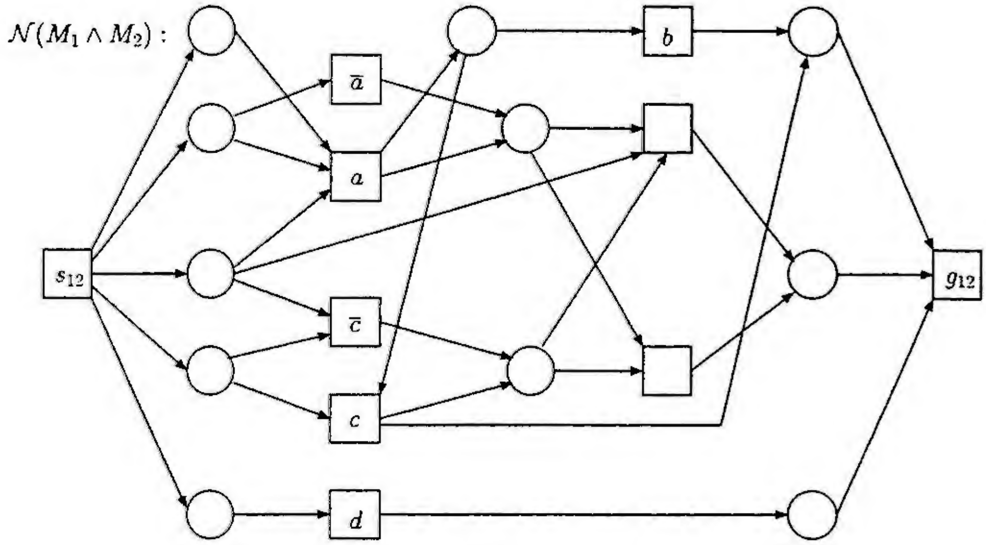


Figure 6

(2) Indirect Proof:

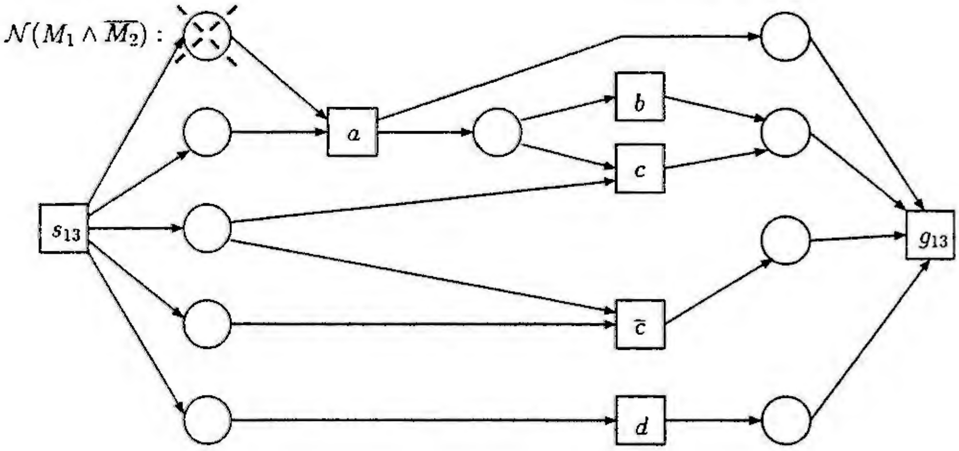


Figure 7

In the net $\mathcal{N}(M_1 \wedge \overline{M}_2)$ of figure 7 exactly one reproduction of the empty marking is possible, i.e. $\mathcal{P}(M_1 \wedge \overline{M}_2) = \{((a < b) \wedge \overline{c} \wedge d)\}$.

Consequently, $M_1 \not\twoheadrightarrow M_2$ holds.

The process $((a < b) \wedge \overline{c} \wedge d)$ of $M_1 \wedge \overline{M}_2$, which is responsible for that, "is" the process $((a < b) \wedge d)$ of M_1 in a version that is completed with respect to $\overline{M}_2$.

This shows the diffculty to apply set operations to process sets.

Needless to say, $((a < b) \wedge d)$ is the same process we missed in $\mathcal{P}(M_1 \wedge M_2)$. □

Aplicational hints. Example

In this section a few hints for calculating processes are given. After that our approach is demonstrated by means of a traffic light example.

According to [Laut92] the empty marking is reproducible in a place/transition net iff there exists a T-invariant whose net representation does neither contain a (structural) deadlock nor a trap (cf. [Reis91]).

T–invariants, deadlocks, and traps can be determined by solving linear homogeneous equation systems (cf. [LautRidd94]).

Example
The bottleneck in a street should be controlled by two traffic lights (see figure 8). The demand for the traffic lights is formulated by the module $S : \overline{dir_1 = dir_2}$ or $\overline{S} : dir_1 = dir_2$.

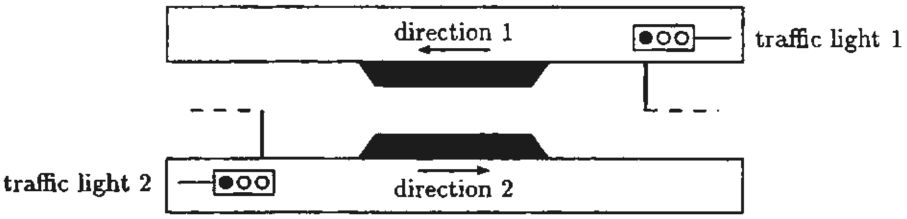


Figure 8

S says that the action "a car passes through the bottleneck in direction 1" must not coincide with the action "a car passes through the bottleneck in (opposite) direction 2 ".

Every traffic light (in Germany) runs through the following cycle: $\underline{red} \rightarrow red$ and $yellow$, red and $yellow \rightarrow green$, $green \rightarrow yellow$, $yellow \rightarrow red$, $red \rightarrow \underline{red}$.

$\underline{red}$ is the red phase in which the other light runs through its cycle. red is the red phase belonging to the cycle.

The following modules describe the course of events of the traffic lights and their synchronization:

$$\begin{array}{lll}
((\underline{red_1} \rightarrow red_1 \text{ and } yellow_1) & < & (red_1 \text{ and } yellow_1 \rightarrow green_1) & < \\
(green_1 \rightarrow yellow_1) & < & (yellow_1 \rightarrow red_1) & < \\
(red_1 \rightarrow \underline{red_1}))^*
\end{array}$$

$$\begin{array}{lll}
((red_2 \rightarrow \underline{red_2}) & < & (\underline{red_2} \rightarrow red_2 \text{ and } yellow_2) & < \\
(red_2 \text{ and } yellow_2 \rightarrow green_2) & < & (green_2 \rightarrow yellow_2) & < \\
(yellow_2 \rightarrow red_2))^*
\end{array}$$

$$((\underline{red_1} \rightarrow red_1 \text{ and } yellow_1) = (red_2 \rightarrow \underline{red_2}))^*$$

$$((\underline{red_2} \rightarrow red_2 \text{ and } yellow_2) = (red_1 \rightarrow \underline{red_1}))^*$$

The car drivers behaviour is expected to be:

$$((red_1 \text{ and } yellow_1 \rightarrow green_1) < dir_1 < (green_1 \rightarrow yellow_1))^*$$

$$((red_2 \text{ and } yellow_2 \rightarrow yellow_2) < dir_2 < (green_2 \rightarrow yellow_2))^*$$

All that results in the module

$$\begin{array}{lllll}
R = & (((\underline{red_1} \rightarrow red_1 \text{ and } yellow_1) = (red_2 \rightarrow \underline{red_2})) & < & (red_1 \text{ and } yellow_1 \rightarrow green_1) & < \\
& (dir_1)^* & < & (green_1 \rightarrow yellow_1) & < \\
& (yellow_1 \rightarrow red_1) & < & & \\
& ((\underline{red_2} \rightarrow red_2 \text{ and } yellow_2) = (red_1 \rightarrow \underline{red_1})) & < & (red_1 \text{ and } yellow_2 \rightarrow green_2) & < \\
& (dir_2)^* & < & (green_2 \rightarrow yellow_2) & < \\
& (yellow_2 \rightarrow red_2))^* & & &
\end{array}$$

The net $\mathcal{N}(R \wedge \overline{S})$ is shown in figure 9.

The easiest way to see that $| \mathcal{P}(R \wedge \overline{S}) | = 0$ holds is to simulate the net. It is impossible to reproduce the empty marking by exactly one occurrance of s and g since the transition $dir_1 = dir_2$ cannot be enabled. So we have an indirect proof that $R \twoheadrightarrow S$ holds.

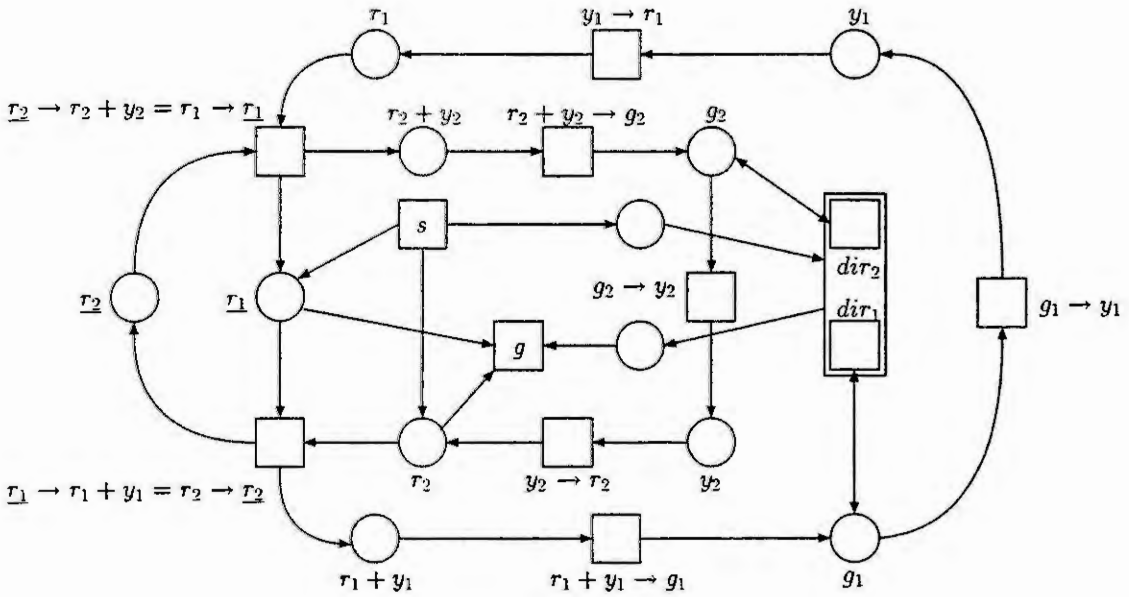


Figure 9

Conclusion

In this paper a simple logic of action (LA) with a process semantics is introduced. The processes are very special in that they are processes in Petri nets that reproduce the empty marking. The reason for doing so is that these processes can be determined by linear algebraic means.

The expressions of LA are called modules. A fulfilment relation based on processes indicates whether a module R fulfils a module S or not. So, the fulfilment relation is a formalization of fulfilling plans, rules, specifications etc.

Even thought LA is extremly simple the idea of direct or indirect proving whether modules fulfil each other can be lifted to higher logics by using correspondingly lifted Petri nets.

References

[Fide93] **Manfred Fidelak**. *Integritätsbedingungen in Petri-Netzen.* Dissertation, Universität Koblenz-Landau, 1993.

[Genr78] **H. J. Genrich**. *Ein Kalkül des Planens und Handelns. Ansätze zur Organisationstheorie rechnergestützter Informationssysteme, GMD-Bericht 111*, S. 77–92, Oldenbourg, 1978.

[Genr80] **H. J. Genrich**. *Ein systemtheoretischer Beitrag zur Handlungslogik, in German.* Lenk (Ed.), *Handlungstheorien interdisziplinär I. Handlungslogik, formale und sprachwissenschaftliche Handlungstheorien*, S. 107–136, Fink, 1980.

[LautRidd94] **Kurt Lautenbach and Hanno Ridder**. *Liveness in bounded Petri nets which are covered by T-invariants. Application and Theory of Petri nets*, Lecture Notes in Computer Science 815, S. 358–375, 1994.

[Laut92] **Kurt Lautenbach**. *The Reproducibility of the Empty Marking in Place/Transition Nets.* Fachberichte Informatik 16/92, Universität Koblenz-Landau, 1992.

[Linz92] **R. Linz**. *Grundbegriffe für eine Logik über Ereignisse und Normen: Ein semantischer Ansatz auf der Basis von Petrinetzen. GMD-Bericht 200.* Oldenbourg, 1992.

[Reis91] **W. Reisig**. *Petri Nets, 2nd edition.* Springer, 1991.

QUANTITIES OF PETRI SYSTEMS

Hartmann J. Genrich
GMD – Forschungszentrum Informationstechnik GmbH
Schloß Birlinghoven, D–53754 Sankt Augustin, Germany

Abstract: In the context of using Petri nets in the analysis of industrial or business processes, two kinds of *quantities* are identified: *S-quantities*, connected with the places of the net, and *T-quantities* connected with its transitions. T-quantities show some similarity to fields. They can be derived as the gradient of a S-quantitiy as potential if and only if their dot product ('line integral') along any *cyclic* process equals zero.

Introduction

This note addresses certain aspects of using Petri nets in the analysis of computerized systems: engineering systems, manufacturing systems, workflow systems, communication systems – whatever Petri net models are used for in practice. Any such system when adequately presented by a Petri net is called a *Petri system* in the sequel. And throughout this note *Petri nets* means *higher-level* ones as represented by *PrT-nets* [4].

Technically, the note contains little new for those familiar with Petri nets and with the notions of *S-invariants* and *T-invariants* in particular, the solutions of homogeneous linear equation systems based on the incidence matrix of a Petri net. The mathematics of S- and T-invariants is quite developed [1, 7, 8]. The *pragmatics* of invariants, however, the way to use them in real applications is still unsatisfactory.

The reasons seem to be twofold, at least. One, if a S-invariant turns out to be useful in verifying some relevant system property, it always seems to be a matter of good luck. Two, it is not so easy to formulate and verify invariants even in tools so powerful as *Design/CPN*[10] or *INA*[9]. This note gives a twist to the understanding and usage of invariants. It shifts our focus from S-invariants to those *quantities* that may be, or not be, constant, and from T-invariants to those *changes* that may be *cyclic* or not.

Mechanical systems, for example, are characterized in terms of quantities like mass, displacement, momentum, force, work, energy, etc. and the relationships between such quantities. Workflow systems, however, require very different quantities to be associated, like work-load, resources, performance, throughput, costs, and many more. While PrT-nets introduced the notion of an *individual* or *value* into the theory of Petri nets [3], this note goes beyond that. It distinguishes between *S-quantities* that are connected to the markings of a Petri net and *T-quantities* that are connected to the changes. T-quantities turn out to be like *fields*. They can be derived as the *gradient* of a S-quantitiy – *a potential* – if and only if their dot product – *line integral* – along any *cyclic* process equals zero.

An Example

The note employs a simple Petri system, shown in figure 1, as the running toy example. It presents a simple production schema consisting of several nested loops. It is denoted by Λ in the sequel. It has been extracted from a more complex Petri system in the field of chemical engineering – a recipe for batch process control [6] – but it could as well have been derived from the transaction processing in a database system or the processing of work objects in a workflow system or, just some piece of software.

The example has been built and run by means of Design/CPN [10] and hence it follows its syntactical conventions. The declarations of its datatypes (color sets) and other non-logical symbols are shown in table 1. In the symbolic analysis of Petri nets that we favour in this note, however, they play a minor role only.

For the time being treat the shaded transitions that indicate how the system may interact with other systems or its environment as comment. Then table 2 presents the Petri system Λ in an equivalent form, namely by its *incidence matrix*. The incidence matrix, C, of a Petri net has a row C_s for each place s and a column C^t for each transition t. Each entry C_s^t is the combination of arc expressions between s and t, the inputs to t taken negative.

The incidence matrix of a Petri net is the object to be studied in merely symbolical terms. A little tool kit has been written in Standard ML using Design/CPN's SML interface [5]. It is going to help us with all the manipulations of the incidence matrix discussed in the sequel.

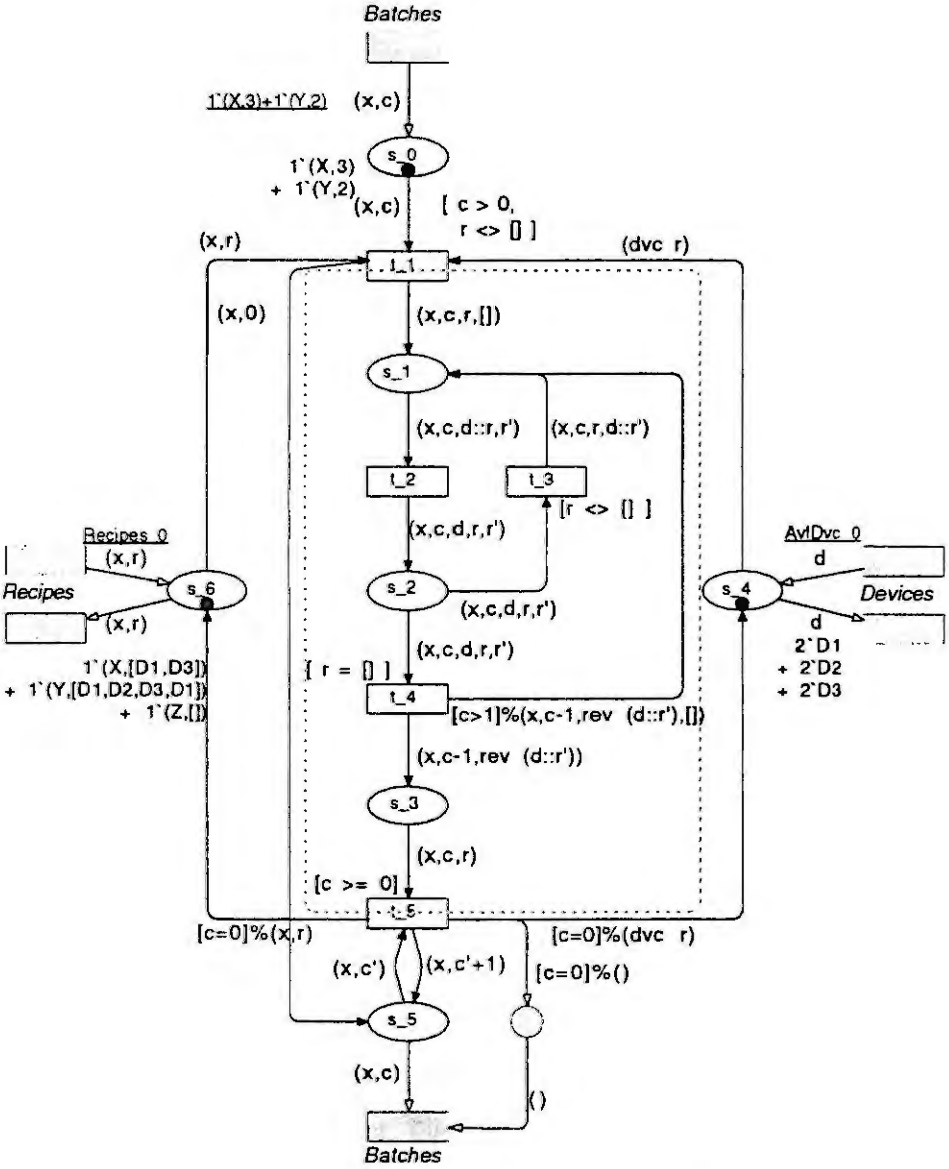


Figure 1: Petri System: Production Schema

To get acquainted with the example, let us consider what happens with one of the tokens of the initial marking of place s_0. It is the batch $(X, 3)$, *'produce 3 units of substance* X*'*, that on place s_0 appears as an incoming order.

The processing of the batch begins with an occurrence of transition t_1 for $\{c, r, x := 3, [D1, D3], X\}$ which takes the batch $(X, 3)$ from the *incoming orders*, s_0, reads and locks the recipe for X, $r = [D1, D3]$, from the *recipe database*, s_6, removes the multi-set of devices listed in r, (dvc r) $= 1`D1 + 1`D3$, from the *pool of available devices*, s_4, puts an empty container for substance X on *outgoing results*, s_5, and puts the status vector for the batch on place s_1 where the production cycle begins.

All this happens, at the chosen level of abstraction, as a single indivisible *event*. However, it may occur *only if* the binding for c and r satisfies the list of constraints stated in the *guard* of t_1, $[\ c > 0, r <> [\,]\ ]$, that is if the count c and the recipe r are meaningful and non-trivial.

The result of applying $\{c, r, x := 3, [D1, D3], X\}$ to t_1 is shown in table 3(a). Note that in our symbolic approach no evaluation takes place. Rather, individual variables are replaced by terms and the resulting expressions may be transformed by some simple equivalence transformations as reduction rules.

Through transition t_1 the kernel of the processing of the batch has been entered. It consists of two loops. The inner one, determined by transitions t_2 and t_3, follows the recipe step by step until one unit

```
color Sbs = union X + Y + Z; var x,x' : Sbs;
color Cnt = int; var c,c' : Cnt;
color Bch  = product Sbs * Cnt;

color Dvc = with D1 | D2 | D3 declare ms; var d,d' : Dvc;
color Stp = Dvc;
color Rcp = list Stp; var r,r' : Rcp;

val AvlDvc_0   = 2 * Dvc;
val Recipes_0  = 1'(X,[D1,D3]) + 1'(Y,[D1,D2,D3,D1]) + 1'(Z,[]);

fun dvc r     = list_to_ms r;
```

Table 1: Global Declarations for the Production Schema

	t_1	t_2	t_3	t_4	t_5	M^0
	[c>0, r<>[]]		[r<>[]]	[r=[]]	[c>=0]	
s_0	– (x,c)					$1`(X,3)$ $+1`(Y,2)$
s_1	(x,c,r,[])	– (x,c,d::r,r')	(x,c,r,d::r')	[c>1]%(x,c-1, rev(d::r'),[])		
s_2		(x,c,d,r,r')	– (x,c,d,r,r')	– (x,c,d,r,r')		
s_3				(x,c-1,rev(d::r'))	– (x,c,r)	
s_4	– (dvc r)				[c=0]%(dvc r)	*AvlDvc*^{0}
s_5	(x,0)				– (x,c') + (x,c'+1)	
s_6	– (x,r)				[c=0]%(x,r)	*Recipes*^{0}

Table 2: Incidence Matrix of the Production Schema

of substance x is done. A single production step is identified, for the sake of simplicity, with utilizing a particular device (resource). In the outer loop, when one unit is done, t_4 puts it on place s_3 for being accumulated at s_5. And, as long as more units have to be produced yet, t_4 also restores the recipe and decreases the count in the status vector, and enters another round.

Transition t_5 accumulates the single units of a substance in the corresponding container that was put on place s_5 by transition t_1. If the whole batch is done, t_5 also returns the set of devices that were reserved for the batch in the beginning and unlocks the recipe used for substance X thus completing the processing of a single batch.

Let us now summarize the whole processing of batch $(X, 3)$ by adding up, for each transition, all the bindings for which this transition occurs. In doing so we deliberately disregard any order in which the transitions occur for the listed bindings, and whether the transition occurrences are enabled or not. The result is shown in table 3(b). It is a *T-vector* whose entries are combinations of bindings. We call such a T-vector a *change* in the sequel.

The rows C_s of the incidence matrix C (see table 2) are T-vectors, too, whose entries are combinations of arc expressions. If we apply, for each transition t, the combinations of bindings of the change in table 3(b) to the corresponding arc expressions, we get the transformed incidence matrix shown in table 4. If we add-up all columns, we get the net effect that the change has on the markings of each place.

In order to add the columns, the entries in each row must be *uniform* in the following sense. They not only must be of the same datatype but all identifiers must have the same scope, too. All expressions in one row are of the same datatype assocated as a *colour set* with the corresponding place. The scope of the indvidual variables, however, varies from column to column; the *binding types* of the expressions

	t_1
	[3>0, $[D1, D3]$<>[]]
s_0	$-\ (X, 3)$
s_1	$(X, 3, [D1, D3], [])$
s_2	
s_3	
s_4	$-$ (dvc $[D1, D3]$)
s_5	$(X, 0)$
s_6	$-\ (X, [D1, D3])$

(a) $\Delta t = t_1{:}\{c, r, x := 3, [D1, D3], X\}$

t_1		$1\text{`}\{x, c, r := X, 3, [D1, D3]\}$
t_2		$1\text{`}\{x, c, d, r, r' := X, 3, D1, [D3], []\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 3, D3, [], [D1]\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 2, D1, [D3], []\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 2, D3, [], [D1]\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 1, D1, [D3], []\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 1, D3, [], [D1]\}$
t_3		$1\text{`}\{x, c, d, r, r' := X, 3, D1, [D3], []\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 2, D1, [D3], []\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 1, D1, [D3], []\}$
t_4		$1\text{`}\{x, c, d, r, r' := X, 3, D3, [], [D1]\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 2, D3, [], [D1]\}$
	$+$	$1\text{`}\{x, c, d, r, r' := X, 1, D3, [], [D1]\}$
t_5		$1\text{`}\{x, c, c', r := X, 2, 0, [D1, D3]\}$
	$+$	$1\text{`}\{x, c, c', r := X, 1, 1, [D1, D3]\}$
	$+$	$1\text{`}\{x, c, c', r := X, 0, 2, [D1, D3]\}$

(b) Summary of Processing of Batch $(X, 3)$

Table 3: Changes as Combinations of Substitutions

differ. Conversely, all expressions of one column have the same binding type but the datatypes vary from row to row. Hence, in order to add columns, individual variables must be replaced by closed terms or by global parameters (individual identifiers of a global scope). In order to add rows – as we do in the next section – the expressions must be transformed such that the datatypes are the same in all entries of a column.

Formally, the operation we have applied to the incidence matrix in order to get the *effect* of a change shows all the ingredients of an (asymmetrical) *matrix product* between the incidence matrix C and the one-column matrix (T-vector) denoting a change. The result is exactly what we expect. The batch $(X, 3)$ is transformed from an unprocessed order into a container with 3 units of substance X. All internal places are unaffected. Their markings are restored to the initial ones. Restricted to the inner part of the system, the change is *cyclic*.

Every binding for a transititon t that satisfies the guard of t denotes an *elementary change* of the Petri system at hand. Our generic notation for such an elementary change will be Δt. Denoting the corresponding effect on the places as Δs and viewing Δt as a unit T-vector, we get as the *fundamental equation* of Petri systems

$$\Delta s = C \bullet \Delta t \tag{1}$$

S-Quantities

Quantities of physical systems, like displacement, force, energy, charge are real-valued functions of time, continuous and differentiable sufficiently often. For Petri systems we do not expect that much structure. However, a minimum of numbership should be found with the quantities of Petri systems, too. That is, we call some observable that we associate with a Petri system, a *quantity* only if we can *add* its values to each other and *multiply* them by *scalar* numbers. These scalars may be integer, rational or real numbers.

Structures whose elements can be added to each other and multiplied by numbers are called *modules*. Every marking of a place, for example, is an element of a particular module; it is a combination of structured tokens, a so-called *multi-set* of values of the datatype (color set) associated with that place.

Let X be a set of *pieces*, usually just some datatype. A *(finite, integer, linear) combination* of X is a

	t_1	t_2	t_3	t_4	t_5	Δs
s_0	- (X,3)					- (X,3)
s_1	(X,3,[D1,D3],[])	- (X,3,D1::[D3],[]) - (X,3,D3::[],[D1]) - (X,2,D1::[D3],[]) - (X,2,D3::[],[D1]) - (X,1,D1::[D3],[]) - (X,1,D3::[],[D1])	(X,3,[D3],D1::[]) + (X,2,[D3],D1::[]) + (X,1,[D3],D1::[])	[3>1]%(X,3-1, rev(D3::[D1]),[]) + [2>1]%(X,2-1, rev(D3::[D1]),[]) + [1>1]%(X,1-1, rev(D3::[D1]),[])		0
s_2		(X,3,D1,[D3],[]) + (X,3,D3,[],[D1]) + (X,2,D1,[D3],[]) + (X,2,D3,[],[D1]) + (X,1,D1,[D3],[]) + (X,1,D3,[],[D1])	- (X,3,D1,[D3],[]) - (X,2,D1,[D3],[]) - (X,1,D1,[D3],[])	- (X,3,D3,[],[D1]) - (X,2,D3,[],[D1]) - (X,1,D3,[],[D1])		0
s_3				(X,3-1,rev(D3::[D1])) + (X,2-1,rev(D3::[D1])) + (X,1-1,rev(D3::[D1]))	- (X,2,[D1,D3]) - (X,1,[D1,D3]) - (X,0,[D1,D3])	0
s_4	- (dvc [D1,D3])				[2=0]% (dvc [D1,D3]) + [1=0]% (dvc [D1,D3]) + [0=0]% (dvc [D1,D3])	0
s_5	(X,0)				- (X,0) + (X,0+1) - (X,1) + (X,1+1) - (X,2) + (X,2+1)	(X,3)
s_6	- (X,[D1,D3])				[2=0]% (X,[D1,D3]) + [1=0]% (X,[D1,D3]) + [0=0]% (X,[D1,D3])	0

Table 4: Incidence Matrix Transformed by a Change

mapping $l : X \to \mathbf{Z}$ such that for finitely many elements x of X only, the *coefficient* $l(x)$ is different from 0. The set of all combinations of X is denoted by $\mathcal{L}(X)$.

Let Σ be a Petri system. A linear operator that for some module Y, assigns to every conceivable marking of Σ an element of Y, is called a *S-quantity* of Σ.

Table 5 shows three S-vectors of anonymous functions that commute with substitution. They denote S-quantities of our production schema Λ. Every such *distribution*, as we call them, assigns to every token of the respective place s, hence by linear extension to every marking of s, a combination of pieces. It denotes the combination of pieces that each token on s contributes to the respective S-quantity.

	WorkLoad			*Recipes*			*SignOfCount*		
s_0	(x, c)	$\mapsto$	$c\grave{}x$	$\_$	$\mapsto$	0	$(\_, c)$	$\mapsto$	$[c > 0]\cdot c\ \grave{}\ ()$
s_1	$(x, c, \_, \_)$	$\mapsto$	$c\grave{}x$	$(x, \_, r, r')$	$\mapsto$	$(x, (\text{rev } r')\hat{}\hat{}r)$	$(\_, c, \_, \_)$	$\mapsto$	$[c > 0]\cdot c\ \grave{}\ ()$
s_2	$(x, c, \_, \_, \_)$	$\mapsto$	$c\grave{}x$	$(x, \_, d, r, r')$	$\mapsto$	$(x, (\text{rev } d{::}r')\hat{}\hat{}r)$	$(\_, c, \_, \_, \_)$	$\mapsto$	$[c > 0]\cdot c\ \grave{}\ ()$
s_3	$(x, \_, \_)$	$\mapsto$	$1\grave{}x$	(x, c, r)	$\mapsto$	$[c = 0]\grave{}(x, r)$	$(\_, c, \_)$	$\mapsto$	$[c \geq 0]\cdot 1\ \grave{}\ ()$
s_4	$\_$	$\mapsto$	0	$\_$	$\mapsto$	0	$\_$	$\mapsto$	0
s_5	(x, c)	$\mapsto$	$c\grave{}x$	$\_$	$\mapsto$	0	$(\_, c')$	$\mapsto$	$c'\ \grave{}\ ()$
s_6	$\_$	$\mapsto$	0	(x, r)	$\mapsto$	(x, r)	$\_$	$\mapsto$	0

Table 5: Some S-Quantities of the Production Schema

The S-quantity *WorkLoad*, for example, gives the work-load of Λ at an arbitrary marking in terms of the number of units of each substance. It shows how the work objects (batches) are spread over the places at the various stages of the production process. Each entry of the vector specifies the pieces that

	t_1	t_2	t_3	t_4	t_5
	[c>0, r<>[]]		[r<>[]]	[r=[]]	[c>=0]
s_0	$-c`x$				
s_1	$c`x$	$-c`x$	$c`x$	$[c{>}1]\cdot(c-1)`x$	
s_2		$c`x$	$-c`x$	$-c`x$	
s_3				$1`x$	$-1`x$
s_4	0				0
s_5	$0`x$				$-c'`x + (c'+1)`x$
s_6	0				0
∇	0	0	0	$[c{>}1]\cdot(c-1)`x$ $-\ (c-1)`x$	0

Table 6: Incidence Matrix Transformed by *WorkLoad*

the tokens of the corresponding place contribute to the current work-load. For the initial marking shown in figure 2, the value of *WorkLoad* is

$$\begin{array}{lrll} & (1`(X,3)+1`(Y,2)) : & \{(x,c)\mapsto c`x\} & \\ + & 0 : & \{(x,c,\_,\_)\mapsto c`x\} & \\ + & 0 : & \{(x,c,\_,\_,\_)\mapsto c`x\} & \\ + & 0 : & \{(x,\_,\_)\mapsto 1`x\} & \\ + & \mathit{AvlDvc}^0 : & \{\_\mapsto 0\} & \\ + & 0 : & \{(x,c)\mapsto c`x\} & \\ + & \mathit{Recipes}^0 : & \{\_\mapsto 0\} & = \quad 3`X + 2`Y \end{array} \tag{2}$$

Table 6 shows the result of applying every entry of the S-vector *WorkLoad* to the entries of the corresponding row of the incidence matrix (table 2). The bottom row, ∇, is the formal sum of the rows for places $s_0 \dots s_6$. It gives for each transition t the effect the corresponding elementary change Δt has to the S-quantity *WorkLoad*. We call it the *defect* of *WorkLoad* with respect to transition t.

The defect of a S-quantity q with respect to a transition t shows a strong similarity to the partial derivative. Denoting it by $\frac{\partial q}{\partial t}$, we get in analogy to equation 1,

$$\Delta q = \frac{\partial q}{\partial t}\Delta t \tag{3}$$

and, pushing the analogy a little further, the bottom row ∇ of table 6 denotes indeed the *gradient* of the S-quantity *WorkLoad*.

All entries of ∇ *WorkLoad* equal zero except for transition t_4. The batch count c, however, remains greater than zero through t_4 for all meaningful processes of the production schema (see below). And for $c > 0$, the defect of *Workload* at t_4 reduces to 0 as well. Hence ∇Workload = 0 for all meaningful processes; the S-quantity *Workload* is constant.

If you follow the same procedure concerning the other two S-quantities of table 5, you will see after some obvious reductions of the gradients that also *Recipes* and *SignOfCount* are constant or, as the terminology is in net theory, are *S-invariants*. *Recipes* shows how a recipe is decomposed during the production process, and *SignOfCount* is an auxiliary S-quantity whose invariance can be used to prove that a non-negative batch count may never become negative.

Exchange of S-Quantities

S-quantities of a Petri system Σ that are constant satisfy a conservation law. Within Σ, they may be distributed over the various parts, or subsystems, but their total remains unchanged. This leads directly to the notion of *exchange* of S-quantities between (sub-)systems, or between a system and its environment.

Let Σ_1 and Σ_2 be two Petri systems with S-quantities q_1 and q_2, respectively. Σ_1 and Σ_2 are said to be able to *exchange* q_1 and q_2 if they can be coupled in such a way that

$$q_1 + q_2 = \text{const.} \tag{4}$$

	t_1	t_2	t_3	t_4	t_5	w_1		
s_1	(x,c,r,[])	– (x,c,d::r,r')	(x,c,r,d::r')	[c>1]%(x,c-1, rev(d::r'),[])		$(x, c, \_, \_)$	↦	c`x
s_2		(x,c,d,r,r')	– (x,c,d,r,r')	– (x,c,d,r,r')		$(x, c, \_, \_, \_)$	↦	c`x
s_3				(x,c-1,rev(d::r'))	– (x,c,r)	$(x, \_, \_)$	↦	1`x
∇w_1	c`x	0	0	[c>1]·(c−1)`x − (c−1)`x	−1`x			
						w_2		
s_0	– (x,c)					(x, c)	↦	c`x
s_4	– (dvc r)				[c=0]%(dvc r)	_	↦	0
s_5	(x,0)				– (x,c') + (x,c'+1)	(x, c)	↦	c`x
s_6	– (x,r)				[c=0]%(x,r)	_	↦	0
∇w_2	−c`x	0	0	0	1`x			

Table 7: Exchange of *WorkLoad* Between Production Kernel and Embedding

or, equivalently,

$$\Delta q_1 + \Delta q_2 = 0 \tag{5}$$

It is obvious from equations (4) and (5) that exchangeable quantities must be comparable; their value domains must be the same module. If their values cannot be added, the equations make no sense. Also note that we talk here about 'pure' exchange. In the coupled system, $\Sigma = \Sigma_1 \& \Sigma_2$, while q_1 and q_2 will vary in general, the quantity $q = q_1 + q_2$ is constant. There is no loss or gain, no destruction or creation of q but mere exchange between Σ_1 and Σ_2. If a S-quantity that represents an *exchange quantity* like energy, momentum, or charge is not constant, its defect respectively gradient indicates where there is exchange with the environment.

There are two ways how Petri systems may be coupled in order to exchange one or more quantities according to equation (4). One way is the *fusion* of places. It is provided by Design/CPN. The dual one, fusion of transitions, is not yet offered by Design/CPN but it is known how to do it (see [2]). One may also think of connecting the two systems by some kind of bridges but then, how are the bridges connected to the systems?

The re-formulation of equation (4) by (5) suggests coupling at transitions since any Δq is the effect of an elementary transition occurrence, Δt. Consequently let us divide our production schema Λ into Λ_1 and Λ_2 by splitting transitions t_1 and t_5 (as indicated in figure 1). The result is shown in table 7. The defects of the two separate S-quantities w_1 and w_2 clearly show how *WorkLoad* is exchanged between the two subsystems through transitions t_1 and t_4.

Now that we know a little about the exchange of quantities between systems, we can unravel the secret about the shaded 'dangling' transitions in figure 1. The one feeding into place s_0 indicates how the environment may increase the work-load by entering a new batch. It may also be viewed as putting the initial marking on s_0. And the shaded transition out of place s_5 decreases the work-load by taking delivery of a finished batch.

Hence this particular pair of transitions represents the *exchange type* of our production schema Λ with respect to quantity *WorkLoad*. If they were considered part of the system (they had to show up in the incidence matrix, table 2), *WorkLoad* would no longer be invariant. Coupling of system Λ with another system or its environment in order to exchange *WorkLoad* requires fusing of the dangling transitions with their corresponding opposites. As a special case, they can be fused with each other to close Λ with respect to *WorkLoad*. Λ would then process the same set of batches over and over again.

In the same way, the pair of shaded transitions at place s_6 indicate that the recipe for some substance x may be changed when the current one is not in use. And at place s_4, equipment that is currently idle may be removed or added. Without the dangling transitions, available equipment and recipes would be considered invariable.

	t_1	t_2	t_3	t_4	t_5
type	$\{x, c, r\}$	$\{x, c, d, r, r'\}$	$\{x, c, d, r, r'\}$	$\{x, c, d, r, r'\}$	$\{x, c, c', r\}$
guard	[c>0, r<>[]]		[r<>[]]	[r=[]]	[c>=0]
Costs	**0**	$K_2(x, d)$	**0**	**0**	$K_5(x)$
∇*WorkLoad*	0	0	0	[c>1]·(c−1)\`$x$ − (c−1)\`x	0

$$K_2 = \begin{array}{c|ccc} & X & Y & Z \\ \hline D1 & 4 & 3 & 0 \\ D2 & 2 & 2 & 0 \\ D3 & 3 & 5 & 0 \\ D4 & 1 & 1 & 0 \end{array} , \quad K_5 = \begin{array}{|ccc} X & Y & Z \\ \hline 1 & 1 & 0 \end{array}$$

Table 8: Production Costs and Other T-Vectors

T-Quantities

Not all interesting quantities that we associate with a Petri system Σ may be functions of markings. There are also quantities that are functions of changes. For example, the production costs for a batch in a production schema, the total time during which a resource is idle respectively occupied, or the *work* performed by a component or subsystem during a particular process – all those quantities cannot be expressed, in general, as a function of markings but rather as functions of changes.

A linear operator that for some module Y assigns an element of Y to every change in a Petri system Σ is called a *T-quantity* of Σ.

Assume we want to express the production costs for batches of our production schema A, in terms of entire units of some currency. We assign to transition t_2 the costs involved using device d during the production of a single unit of substance x, and to transition t_5 the costs for the transport and storage of a single unit of x. Other costs are ignored. The corresponding cost functions are K_2 : *Sbs*×*Dvc*→*int* and K_5 : *Sbs*→*int*. The whole *Costs* function is represented by a T-vector as shown in table 8.

The entries of *Costs* for each transition t_i are integer expressions whose free variables belong to the (binding) type of t_i. For every binding of t_i they denote an integer value. If K_2 and K_5 are given as in table 8, the costs for processing batch $(X, 3)$ are $3\,(K_2^X(D1) + K_2^X(D3) + K_5^X) = 24$. Note that with the given K_2 and K_5, the expressions $K_2(x, d)$ and $K_5(x)$ could be equivalently written as $4[x{=}X, d{=}D1] + \ldots + 0[x{=}Z, d{=}D4]$ respectively $[x{=}X] + [x{=}Y]$. Then it will be even more obvious that the T-vector ∇ *WorkLoad* of table 8 also denotes a T-quantity. In general:

The T-vector of defects (the gradient) ∇q of a S-quantity q is a T-quantity. In our analogy between Petri systems and engineering systems we may say that ∇q is a *field* with *potential* q.

The value of a T-quantity, k, for some change u is independent of the order in which the Δt of u may occur in an actual process. It is just the dot product between k and u (in our analogy: the line integral of field k along change u).

Now assume there are two different changes, u_1 and u_2, both leading from the same marking m^1 to the same marking m^2. Then we may ask whether the values of k along u_1 and u_2 are the same or not. If it is the case that for any pair of markings (m^1, m^2) the value of k is the same along all changes that connect m^1 to m^2, we call k *path-indifferent* ('path' in the reachabilty graph, not in physical space). In physical systems, work is a 'path-indifferent' quantity iff the forces are conservative – no friction, for example.

A T-quantity k is path-indifferent iff its value along any cyclic change is **0**. And it is a linear-algebraical fact that

Theorem For a T-quantity k there exists a S-quantity $\widehat{k}$ such that $k = \nabla\widehat{k}$ – a *potential* for k – iff k is path-indifferent.

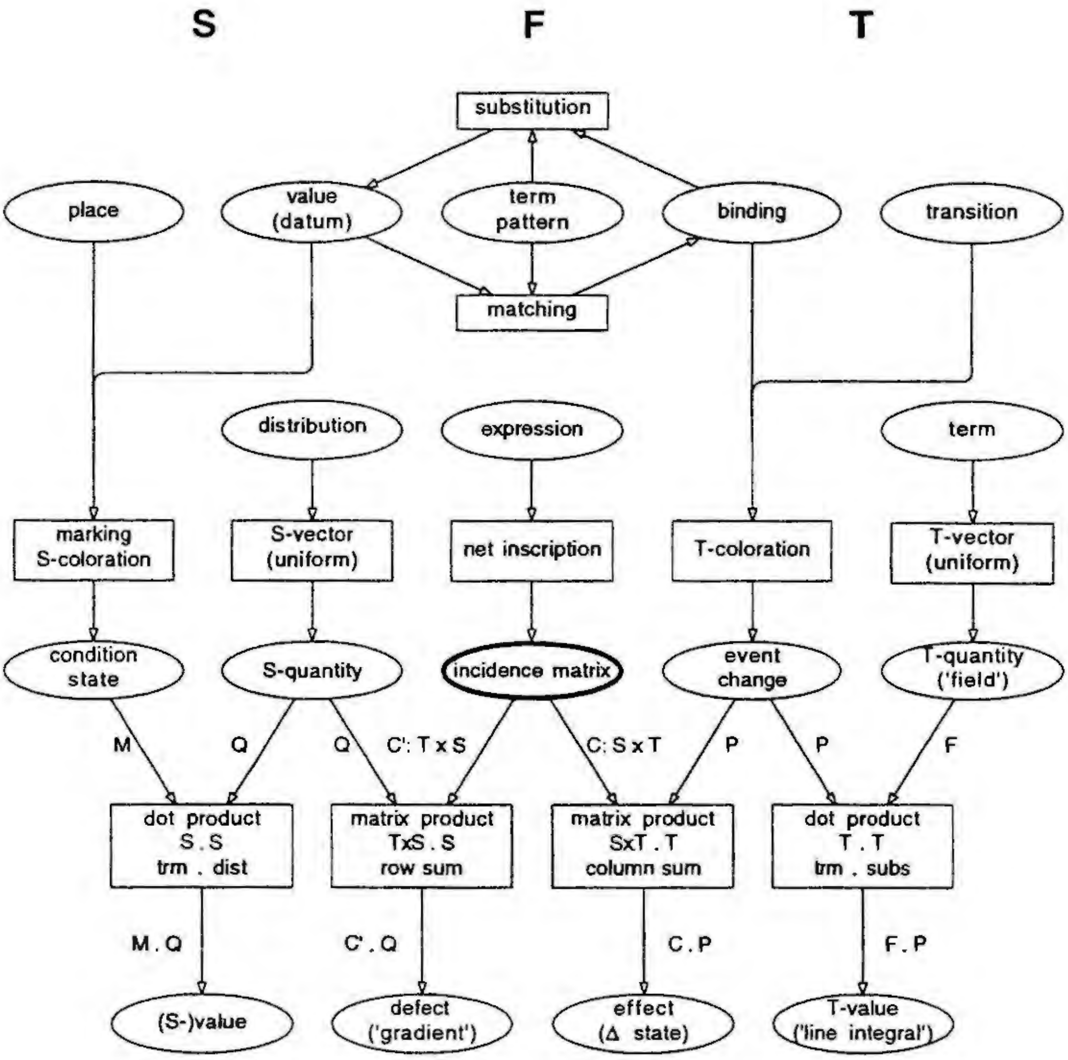


Figure 2: A Map of Notions

The *Costs* for the processing of any set of batches as given in table 8 are independent of the various ways in which these processes may be intertwined. And since there are no real cyclic processes in the production schema not completed by an environment (cf. previous section on exchange), *Costs* happens to be path-independent. In general however, whenever there are cyclic changes with all Δt being non-negative, *Costs* or the time consumed are typical T-quantities that have no potential.

Conclusion

The contents of this note are summarized in figure 2. It shows how the general S-T symmetry of Petri systems, the fact that *state* and *change* are treated on equal footing in net theory, translates into the linear-algebraic *row/column* symmetry. It seems that the S-T symmetry is in turn reflecting a fundamental aspect of information processing in general.

In information processing we deal with two ways of representing information as structured collections of values. One form allows to transport and store information, the other to process it. The first way is to use kind of hard boxes where each value has its fixed position or is contained in a labeled extra box. It's like the arrangement of atoms in a molecule, or of christmas gifts in a parcel for the whole family. A collection of values in molecule form is called a *datum*. As data, value collections can be transmitted over time and space whithout losing their internal arrangement.

In order to access the values in a datum and to process and combine them with other parts of other data, however, we have to unwrap the parcels, destroy their arrangements and put their contents all on one table. To keep the information about the purpose or meaning that is contained in the arrangement we assign them, temporarily and locally for some basic processing step, some names (descriptors). The table with names assigned to values is an *environment* or *binding*. The links between the two forms, datum and binding, are the *patterns*. Matching a pattern to a datum gives a binding, evaluating a pattern under a binding gives a datum.

The state elements of a Petri system, the places, are closely related to data, the molecule kind of representing collections of values. The tokens on each place are data. Their type ('color set') is associated with that place. In contrast, the process elements of a Petri system, the occurrences of transitions, are given as (local and temporary) bindings of individual variables. Their type (the set of free identifiers with their datatype) is associated with each transition.

In the symbolic manipulation of the incidence matrix of a Petri system we have used two kinds of transformations of the entries (the arc expressions), *substitutions* to present changes and *distributions* to present S-quantities. Substitutions and distributions operate on the arc expressions of a Petri system in opposite directions. Given an expression X, a substitution replaces place holders in X by expressions; X serves as a *context* for expressions to be inserted. A distribution, in contrast, decomposes X into subexpressions in order to fill the wholes in its body; X serves as an *'intext'* rather than context. Substitutions affect the binding type of an expression and distributions affect the datatype. Substitutions are applied in a per-transition manner, to all entries of some column. Distributions are applied in a per-place manner, to all entries of some row.

Formal addition of expressions makes sense only if the expressions are *uniform*, i.e. if they have the same datatype *and* the same binding type. In the incidence matrix, the entries of a column (transition) have the same binding type (scope of identifiers) and the rows (places) have the same datatype (color set). Consequently, addition of columns requires adjustment of the binding types by means of substitutions, addition of rows requires adjustment of the datatypes by means of distributions.

This note just pointed at some obvious consequences of these considerations.

References

[1] Best, E.; Thielke, Th.: Coloured nets with Curry. Petri Net Newsletter 50, Bonn : Gesellschaft für Informatik (1996)

[2] Christensen, S.; Hansen, N.D.: Coloured Petri nets extended with channels for synchronous communication. In: Application and Theory of Petri Nets 1994 (R. Valette, Ed.), Lecture Notes in Computer Science Vol. 815. Berlin : Springer (1994)

[3] Genrich, H.J.; Lautenbach, K.: System modelling with high-level Petri nets. Theoretical Computer Science 13 (1981) 109–136

[4] Genrich, H.J.: Predicate/transition nets. In: Petri Nets: Central Models and their Properties. Advances in Petri Nets 1986, Part I (W. Brauer, W. Reisig, G. Rozenberg Eds.), LNCS 254. Berlin : Springer (1987) 207–247

[5] Genrich, H.J.: An experimental package for the symbolic analysis of Petri systems. [Under Construction]

[6] Genrich, H.J.; Hanisch, H.-M.: Modelling and analysis of recipes. Workshop on Analysis and Design of Event-Driven Operations in Process Systems, Imperial College, April 1995.

[7] Jensen, K.: Coloured Petri nets and the invariant method. Theoretical Computer Science 14 (1981) 317–336

[8] Schmidt, K.: Symbolische Analysemethoden für algebraische Petri-Netze. Berlin : Humboldt-University (1996) [PhD Thesis, in German]

[9] Starke, P.: INA - Integrated Net Analyzer Version 1.5. Berlin : Humboldt-University (1995)

[10] Design/CPN reference manual. Version 3.00. Aarhus : DAIMI, Aarhus University (1996)

A COMPARISON OF PETRI NET SUPERVISORY APPROACHES FOR STATE SPECIFICATIONS

Alessandro Giua
DIEE: Dip. di Ingegneria Elettrica ed Elettronica — Università di Cagliari
P.zza d'Armi - 09123 CAGLIARI, Italy — giua@diee.unica.it — Fax: +39 (70) 675-5900

Abstract. The paper discusses a class of state specifications for Petri net models called Generalized Mutual Exclusion Constraints. Several authors have presented solutions to this problem within the framework of Supervisory Control. Here some of these approaches are reviewed presenting them with a consistent notation.

1. Introduction

Generalized mutual exclusion constraints (GMEC) [4] are a natural way of expressing the concurrent use of a finite number of resources, shared among different processes. In traditional Petri net (PN) modeling all transitions are assumed to be controllable, i.e., may be prevented from firing by a control agent. A single GMEC may be easily implemented by a monitor, i.e., a place whose initial marking represents the available units of a resource and whose outgoing and incoming transitions represent, respectively, the acquisition and release of units of the resource.

In the framework of Ramadge and Wonham's supervisory control [16] the complexity of enforcing a GMEC is enhanced by the presence of uncontrollable transitions. It is possible to prove [4] that in presence of uncontrollable transitions, a problem of mutual exclusion is transformed into a more general forbidden marking problem, which is a qualitatively different problem, in the sense that it may not always be solved with the same techniques used when all transitions are controllable.

Two ways have been explored for reducing the computational complexity involved in solving a GMEC problem for nets with uncontrollable transitions.

On one hand, one may consider special PN structures for which the maximally permissive control policy can be easily computed and implemented. There have been several interesting approaches in this sense. Holloway and Krogh [6, 8] presented an approach in which the problem of controlling the marking of a place can be decomposed into the control of paths of uncontrollable transitions and used these techniques to enforce GMEC's on safe marked graphs. Li and Wonham [9, 10] showed how closed-form solutions for GMEC problems may be computed for restricted classes of nets. By closed-form solution the authors mean that the controller may be represented as a net. Giua et al. [5] have discussed several control structures (including monitors) capable of enforcing GMEC's on marked graphs with control safe places.

On another hand, one may give up the requirement that the control policy be maximally permissive and may be willing to accept a more restrictive control policy provided it can be easily computed. This approach has been followed by Moody et al. [12, 13, 17]. In their approach the idea is that of always using very simple controllers in the form of monitor places that only constrain controllable transitions. An algorithm is given to compute such a monitor to ensure that a given GMEC will never be violated.

These approaches are reviewed and compared in this paper.

2. Generalized Mutual Exclusion Constraints

We recall the notion on Petri nets used here. A *Place/Transition net* (P/T net) [14] is a structure $N = (P, T, Pre, Post)$, where P is a set of n *places*; T is a set of m *transitions*; $Pre : P \times T \rightarrow \mathbb{N}$ and $Post : P \times T \rightarrow \mathbb{N}$ are the *pre-* and *post-incidence functions* that specify the arcs. The *incidence matrix* of the net is defined as $C(p,t) = Post(p,t) - Pre(p,t)$. A *P/T system* or *net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0. One writes $M\ [\sigma\rangle\ M'$ to denote that the enabled sequence of transitions σ may fire at M yielding M'. A marking M is *reachable* in $\langle N, M_0 \rangle$ iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. If marking M is reachable in $\langle N, M_0 \rangle$ by firing a sequence σ, then the following *state equation* is satisfied: $M = M_0 + C \cdot \vec{\sigma}$, where $\vec{\sigma} : T \rightarrow \mathbb{N}$ is the *firing count vector*. The set of nonnegative integer vectors M such that there exists a vector $\vec{\sigma}$ satisfying the previous state equation is called *potentially reachable set* and is denoted $PR(N, M_0)$. Note that $PR(N, M_0) \supseteq R(N, M_0)$. However, for acyclic nets — i.e., nets where no direct path forms a cycle — $PR(N, M_0) = R(N, M_0)$ [14]. A net system $\langle N, M_0 \rangle$ is *safe* if $M(p) \leq 1$ for every place p and for every marking $M \in R(N, M_0)$.

Let us define a *generalized mutual exclusion constraint* (GMEC) as a condition that limits a weighted sum of tokens contained in a subset of places [4, 5].

Definition 1. *Let $\langle N, M_0 \rangle$ be a net system with set of places P, $\vec{w} : P \rightarrow \mathbb{Z}$ a weight vector of integers, and $k \in \mathbb{Z}$ a constant. The* support *of $\vec{w}$ is the set $Q_w = \{p \in P \mid w(p) \neq 0\}$. A single* generalized mutual exclusion constraint *$(\vec{w}, k)$ defines a set of legal markings on $\langle N, M_0 \rangle$ $\mathcal{M}(\vec{w}, k) = \{M \in R(N, M_0) \mid \vec{w}^T \cdot M \leq k\}$. A set of GMEC's $(W, \vec{k})$, with $W = [\vec{w}_1 \ldots \vec{w}_r]$ and $\vec{k} = (k_1 \ldots k_r)^T$, defines a set of legal markings $\mathcal{M}(W, \vec{k}) = \{M \in R(N, M_0) \mid W^T \cdot M \leq \vec{k}\}$.*

Markings in $R(N, M_0)$ that are not legal will be denoted *forbidden* markings.

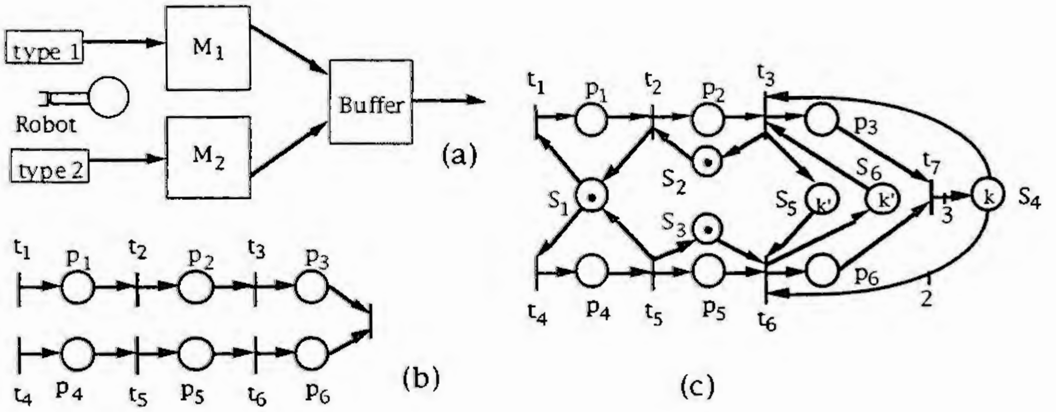


Figure 1: A manufacturing process: (a) layout; (b) Petri net model; (c) Petri net controlled with monitors.

Example 2. As an example of the modeling power of GMEC's, consider the simple manufacturing process with two machines, a robot and a buffer shown in Figure 1.(a). There exists an infinite supply of parts of type 1 (type 2) that are loaded by the robot on machine 1 (machine 2). After machining the parts are directly deposited into the buffer. Machined parts are taken in pairs from the buffer to be assembled.

The process can be represented by the net in Figure 1.(b), where: t_1 and t_2 (t_4 and t_5) represent the start and the end of the loading operation of machine 1 (machine 2); t_3 (t_6) represents the storing of a part of type 1 (type 2) in the buffer; t_7 represents the withdrawal of two parts of different type from the buffer. The places have the following interpretation: p_1 (p_4) represents the parts being loaded on machine 1 (machine 2); p_2 (p_5) represents the parts being machined on machine 1 (machine 2); p_3 (p_6) represents the parts of type 1 (type 2) in the buffer.

The following constraints should be imposed on the system's behavior. a) Only one robot is available, hence only one loading operation may be executed at a given time, i.e., $M(p_1) + M(p_4) \leq 1$. b) Only one part can be machined at a given time on each machine, i.e., $M(p_2) \leq 1$, and $M(p_5) \leq 1$. c) The buffer has k slots, and each part of type 1 takes two slots, while each part of type 2 takes one slot. To avoid overflow one wants $2M(p_3)+M(p_6) \leq k$. d) In the buffer, the number of parts of one type should not exceed the number of parts of the other type by more that k' units, i.e., $M(p_3) - M(p_6) \leq k'$, and $M(p_6) - M(p_3) \leq k'$.

As this very simple example shows, GMEC's are a natural way of expressing the concurrent use of a finite number of resources, shared among different processes. The use of weights permits to assign different units of resources to the various processes. The use of negative weights permits to express fairness constraints in the allocation of the shared resources. Note also that constraints to prevent underflow (i.e., of the form $M(p_1) + M(p_2) \geq k \geq 0$) may also be expressed, using negative weights, as $-M(p_1) - M(p_2) \leq -k$.

3. Monitors

In traditional Petri net modeling all transitions are assumed to be *controllable*, i.e., may be prevented from firing by a control agent. A single GMEC may be easily implemented by a *monitor*, i.e., a place whose initial marking represents the available units of a resource and whose outgoing and incoming transitions represent, respectively, the acquisition and release of units of the resource.

Definition 3. *Given a system (N, M_0), with $N = (P, T, Pre, Post)$, and a GMEC $(\vec{w}, k)$, the* monitor *that enforces this constraint is a new place S to be added to N. The resulting system is denoted (N^S, M_0^S), with $N^S = (P \cup \{S\}, T, Pre^S, Post^S)$. Let C be the incidence matrix of N. Then N^S will have incidence matrix C^S and initial marking M_0^S as shown here. Note that there are no selfloops containing S in N^S, hence Pre^S and $Post^S$*

$$C^S = \left[\begin{array}{c} C \\ -\vec{w}^T \cdot C \end{array} \right]; \quad M_0^S = \left(\begin{array}{c} M_0 \\ k - \vec{w}^T \cdot M_0 \end{array} \right)$$

may be uniquely determined by C^S. The initial marking M_0 of the system is assumed to satisfy the constraint $(\vec{w}, k)$.

According to the definition, the monitor that enforces a constraint $(\vec{w}, k)$ will have arcs going to (coming from) all input (output) transitions of a place $p \in |Q_w|$ and such that $w(p) > 0$. If the place p is such that $w(p) < 0$, then the directions of the arcs is reversed. The weight of these arcs depends on the coefficient of $\vec{w}$ and on the coefficients of the incidence matrix C.

As an example, in Figure 1.(c) monitors have been added to the net in Figure 1.(b) to enforce the constraints discussed in Example 2. Monitor S_1 enforces the constraint a); monitors S_2 and S_3 constraints b); monitor S_4 constraint c); and monitors S_5 and S_6 constraints d).

A set of GMEC's can be enforced adding a monitor for each constraint in the set.

In [4] it was proven that the addition of a monitor to the net structure modifies the behavior of a system, to avoid reaching markings that do not satisfy the corresponding GMEC. The monitor is also *maximally permissive* [7], i.e., it only disables transitions whose firing would yield a marking that violates the corresponding GMEC.

Let us compare GMEC's with the most general kind of constraint that can be defined on the marking set of a system, the *forbidden markings* constraint [6, 8]. A forbidden marking constraint consists of an *explicit list* of markings $\mathcal{F}$ that one wants to forbid.

Let $\mathcal{F}$ be any set of forbidden markings on a net system $\langle N, M_0\rangle$. Is it possible to find a set of GMEC's $(W, \vec{k})$ equivalent to $\mathcal{F}$, i.e., such that $R(N, M_0) \setminus \mathcal{F} = \mathcal{M}(W, \vec{k})$? In general the answer is no [4]. However, for safe nets it was shown that there exists a set of GMEC's equivalent to any forbidden marking constraint (see [4]).

4. Nets with uncontrollable transitions

In the framework of supervisory control the complexity of enforcing a GMEC is enhanced by the presence of a set $T_u \subseteq T$ of *uncontrollable* transitions which cannot be disabled by a supervisor. Thus arcs from monitors to uncontrollable transitions are not allowed, since the effect of such an arc would be that of preventing the firing of the transition when the monitor place is not marked.

When the net has uncontrollable transitions, to enforce a given GMEC it is necessary to prevent the system from reaching a superset of the forbidden markings, containing all those markings from which a forbidden one may be reached by firing a sequence of uncontrollable transitions.

Given a system $\langle N, M_0\rangle$ and a set of GMEC's $(W, \vec{k})$, in the presence of uncontrollable transitions the set of legal markings is given as: $\mathcal{M}_c(W, \vec{k}) = \mathcal{M}(W, \vec{k}) \setminus \{M \in R(N, M_0) \mid \exists M' \notin \mathcal{M}(W, \vec{k}), M[\sigma\rangle M' \wedge \sigma \in T_u^*\}$, i.e., one does not consider legal the markings that satisfy $(W, \vec{k})$ but from which a forbidden marking may be reached by firing only uncontrollable transitions. It is necessary to introduce this restriction because a firing sequence $\sigma \in T_u^*$ may not be prevented by a controlling agent.

When all transitions are controllable, it was shown that a *monitor* is capable of enforcing a given GMEC $(\vec{w}, k)$, with a maximally permissive control. In the case of uncontrollable transitions, the maximally permissive control policy should ensure that: *a*) only markings in $\mathcal{M}_c(\vec{w}, k)$ will be reached by the system under control; *b*) all transition firings that yield a marking in $\mathcal{M}_c(\vec{w}, k)$ should be allowed.

It is possible to prove [4] that there may not exist a GMEC $(W, \vec{k})$ such that $\mathcal{M}(W, \vec{k}) = \mathcal{M}_c(\vec{w}, k)$. Thus in presence of *uncontrollable* transitions, a problem of mutual exclusion is transformed into a more general *forbidden marking problem*, which is a qualitatively different problem, in the sense that it may not always be solved with the same techniques used when all transitions are controllable.

5. Petri net techniques for GMEC's

Holloway and Krogh's approach

The PN model considered by Holloway and Krogh is called *controlled PN*. A controlled PN is a P/T net with 2 sets of places, the state places P represented by circles and the input control places C represented as boxes (see Figure 2.(a)). Hence the marking has two components, M (related to P) and U (related to C). U is assumed to be binary.

The marking of places in C is computed by an external agent as a function of the marking of the state places, i.e., $U = f(M)$. The firing of a transition modifies M in the usual way. The control input U is computed again each time a new marking is reached. The main difference wrt the firing policy of P/T nets is that *two enabled transitions may fire simultaneously*.

The basic restriction Holloway and Krogh consider is that the net be a *safe marked graph*, i.e., a safe P/T net such that each place has exactly one input arc and one output arc. In the examples given here, only GMEC $(\vec{w}, k)$ where $w(p) \in \{0, 1\}$ will be considered.

Given a GMEC $(\vec{w}, k)$ the control law is computed in two steps.

Off-line computation. For each place in Q_w compute backwards the paths until controlled transitions are found.

On-line computation. Given a marking M define for each $p \in Q_w$: a) $\Lambda_p(M) = 1$ if all its path are marked (i.e., p may be marked uncontrollably) else $\Lambda_p(M) = 0$; b) $\Delta_p(M) = 1$ if, unless some control input is disabled, at the next marking all paths of p may be marked (M is a boundary marking for p) else $\Delta_p(M) = 0$.

Let: L be the number of places in Q_w such that $\Lambda_p(M) = 1$; B be the number of places in Q_w such that $\Delta_p(M) = 1$; $D(U)$ be the number of places in Q_w such that $\Delta_p(M) = 1$ hut such that U disables some transition whose firing will increase Λ_p. Then a control input is admissible if: $D(U) \geq L + B - k$.

Example 4. Let us consider the GMEC $M(p_1) + M(p_2) \leq 1$. In Figure 2.(a), we have represented the subnet computed during the off-line computations for places p_1 and p_2.

Given the marking in the figure, $\Lambda_{p1}(M) = \Lambda_{p2}(M) = 0$, and $\Delta_{p1}(M) = \Delta_{p2}(M) = 1$, hence $L = 0$, $B = 2$. The possible markings for the control places and the corresponding values of D are: $D(0\ 0\ 0) = D(0\ 1\ 0) = 1$ (p_1 and p_2 cannot be marked); $D(0\ 0\ 1) = D(0\ 1\ 1) = 1$ (p_1 cannot be marked); $D(1\ 0\ 0) = D(1\ 1\ 0) = 1$ (p_2 cannot be marked); $D(1\ 0\ 1) = D(1\ 1\ 1) = 1$ (both p_1 and p_2 can be marked). One needs $D(U) \geq L + B - k = 1$, hence one can choose either $U = (0\ 1\ 1)^T$ or $U' = (1\ 1\ 0)^T$, i.e., there are two maximally permissive controls.

The approach of Holloway and Krogh is extremely efficient, since it requires very simple computations, both in the off-line and on-line steps.

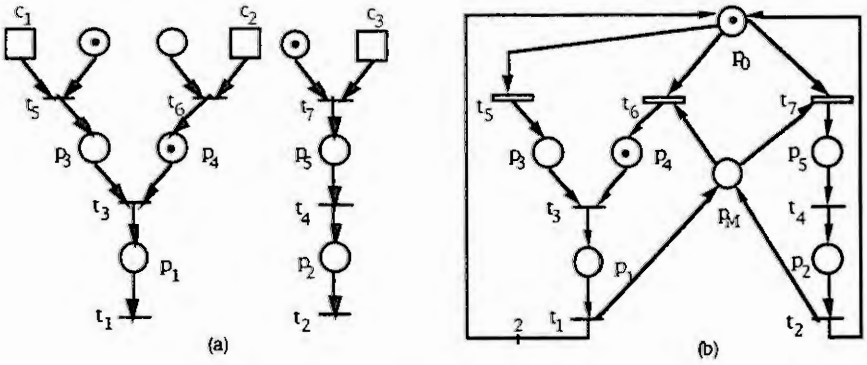


Figure 2: Nets in Examples 4-7.

However, because the controller is given as a feedback law it is not possible to built a net model of the closed-loop system. This approach has received a lot of attention in the literature and has also been extended to classes of nets other than marked graphs: controlled state machines [2], forward and backward conflict-free nets [3], colored nets [1, 11].

Li and Wonham's approach

Li and Wonham have considered Vector Discrete Event Systems, a model that is known to be equivalent to Petri nets [15], and used incidence matrix analysis to compute the control law that enforces GMEC's (that they call Linear Predicates).

Let $\langle N, M_0 \rangle$ be a net system, and let N_u be the uncontrollable subnet, i.e., the subnet obtained from N by removing all controllable transitions. Let $(\vec{w}, k)$ be a GMEC. Then one can write the set of legal markings defined in Section 4 as: $\mathcal{M}_c(w, k) = \{M \mid (\forall M' \in R(N_u, M))\vec{w}^T \cdot M' \leq k\}$. The state equation of a net may be used to decide reachability if the net is *acyclic*, i.e., if no direct path in the net forms a cycle [14]. Thus if N_u is acyclic, a given marking M is in $\mathcal{M}_c(w, k)$ if and only if the following integer programming problem IPP (where C_u is the incidence matrix of N_u) has solution $x^* \leq k$: . $x = \max \vec{w}^T \cdot M'$ such that $M' = M + C_u \cdot \vec{\sigma}$, and $M', \vec{\sigma} \geq \vec{0}$. .

The following algorithm can be used to enforce a GMEC $(\vec{w}, k)$ on nets whose uncontrollable subnet is acyclic.

1. Let the initial marking M_0 be in $\mathcal{M}_c(w, k)$. (Solve IPP with $M = M_0$.)
2. Let $\overline{M}$ be the present marking. For any controllable t enabled by $\overline{M}$:
   a) compute M_t such that $\overline{M}[t\rangle M_t$;
   b) solve IPP with $M = M_t$;
   c) if $x^* \leq k$ then t should be enabled else it should be disabled.
3. As soon as a transition fires go back to step 2.

The condition that the uncontrollable subnet be acyclic is not too restrictive, in the sense that in many practical applications it holds. However, the problem is that the controller has to solve on-line at each step several IPP's. The complexity of IPP's is an open problem. It is doubtful that these problems have polynomial complexity in the size of the constraint set. Thus, the approach is unfeasible in practical applications.

This motivated Li and Wonham to study other classes of nets for which the controller may be represented as a net. The method can be applied to ordinary nets whose uncontrollable transitions form either tree structures of type TS1 (each transition has a single output arc) or tree structures of type TS2 (each transition has a single input arc) as defined in [10]. Another mild restriction is that the uncontrollable subnet be composed by non connected components and two or more places in Q_w cannot belong to the same component (mutual independence).

In the case of TS2 nets they show that given a GMEC $(\vec{w}, k)$, there exists a new constraint $(\vec{w}_c, k_c)$ such that the set of legal markings can be written as: $\mathcal{M}_c(w, k) = \{M \mid \vec{w}_c^T \cdot M \leq k_c\}$ and the monitor that enforces $(\vec{w}_c, k_c)$ does not have arcs going to uncontrollable transitions. Thus $(\vec{w}, k)$ can be enforced using the monitor for $(\vec{w}_c, k_c)$.

In the case of TS1 nets they show that given a GMEC $(\vec{w}, k)$, there exists a new set of constraints $(\vec{w}_i, k_i)$ $(i = 1, \ldots, r)$ such that: $\mathcal{M}_c(w, k) = \{M \mid \bigvee_{i=1}^{r} \vec{w}_i^T \cdot M \leq k_i\}$ ($\bigvee$ is the disjunction operator, i.e., the logical OR) and the monitor that enforces each $(\vec{w}_i, k_i)$ does not have arcs going to uncontrollable transitions. Note, however, that in this second case the net structure corresponding to the disjunction operator cannot be represented as a net. In fact the addition of r monitors, one for each $(\vec{w}_i, k_i)$, would enforce the conjunction (logical AND) of these constraints. In [10] is defined a new structure, called *generalized vector addition system*, that can enforce a disjunction of constraints.

Example 5. Let us consider the GMEC $(\vec{w}, k)$ requiring $M(p_1) + M(p_2) \leq 1$ and the net in Figure 2.(a). The uncontrollable subnets for p_1 and p_2 (controllable transitions t_5, t_6 and t_7 should be removed), are TS1 nets.

Let $(\vec{w}_1, k_1)$ and $(\vec{w}_2, k_2)$ be the constraints requiring, respectively: $M(p_1)+M(p_2)+M(p_3)+M(p_5) \leq 1$ and $M(p_1) + M(p_2) + M(p_4) + M(p_5) \leq 1$. Clearly, $\mathcal{M}_c(w, k) = \{M \mid \vec{w}_1^T \cdot M \leq k_1 \vee \vec{w}_2^T \cdot M \leq k_2\}$.

Giua, DiCesare, and Silva's approach

The authors consider marked graphs with *control safe places* and GMEC's with $\vec{w} \in \mathbb{N}^m$.

A transition t belongs to the set of *control transitions* A_p of a place p iff: a) t is controllable; b) there exists a path from t to p that does not contain controllable transitions except t. A place p is *control safe* if on at least one path from each $t \in A_p$ to p the number of tokens cannot exceed one. A transition $t \in A_p$ is said to be *constraining* at a marking M if there exists a path from t to p that is unmarked at M.

For this classes of nets and constraints, the authors showed that given a GMEC $(\vec{w}, k)$, there exists a set of GMEC's $(W_c, \vec{k}_c)$ such that $\{M \mid W_c^T \cdot M \leq \vec{k}_c\} = \mathcal{M}_c(\vec{w}, k)$. Hence a maximally permissive control law for $(\vec{w}, k)$ may always be implemented by a set of monitors.

In particular, if $(\vec{w}, k)$ is such that $w(p) \in \{0, 1\}$ and $|Q_w| = k + 1$, then the set $(W, \vec{k})$ reduces to a single constraint. This constraint can be enforced by a monitor place p_0 with (for all $p \in Q_w$): a) one arc going from p_0 to each transitions in A_p ; b) $|A_p|$ arcs going from the output transition of p to p_0. The initial marking of p_0 is equal to $d - 1$, where d is the number of constraining transitions in $A = \cup_{p \in Q_w} A_p$ at the initial marking M_0.

Example 6. Let us consider the GMEC $(\vec{w}, k)$ requiring $M(p_1) + M(p_2) \leq 1$ and the net in Figure 2.(b) without places p_0 and p_M and their input/output arcs. Here the controllable transitions (t_5, t_6 and t_7) are shown as boxes. Let us assume that places p_1 and p_2 are control safe. The set control transitions for place p_1 is $A_{p1} = \{t_5, t_6\}$. The set of control transitions for place p_2 is $A_{p2} = \{t_7\}$. Transitions t_5 and t_7 are constraining. Since the constraint $(\vec{w}, k)$ is such that $|Q_w| = |\{p_1, p_2\}| = 2 = k + 1$ and $w(p_1) = 1$, $w(p_2) = 1$, this constraint can be enforced by a single monitor, place p_0 in Figure 2.(b).

Assume now $(\vec{w}, k)$ is such that (for some p) $w(p) > 1$, or $|Q_w| > k + 1$. The previous construction may not be used. However it was shown in [5] that the original constraint may be rewritten as a set of at least r constraints $(\vec{w}_j, k)$ where $|Q_{w_j}| = k + 1$ and each of these constraints may be enforced by a monitor. However the problem is that $r = \begin{pmatrix} |Q_w| \\ k+1 \end{pmatrix}$, thus in the worst case the number of monitors is exponential with respect to the cardinality of Q_w. In these cases, a different supervisory based control structure — which grows linearly with the number of places in the support of the weight vector — can be used [5]

Moody, Antsaklis, and Lemmon's approach

These authors use monitors as control structure to be added to the net structure for enforcing GMEC's (called *place invariants*). When there are uncontrollable transitions, they still use monitor based solutions but in this case the solution may not be maximally permissive.

Let us consider a set of GMEC's $(W, \vec{k})$ to be enforced on a net system (N, M_0). Let W be a $(m \times r)$ matrix, where m is the number of places of the net and r the number of constraints in the set. Let C be the incidence matrix of the net and C_u the incidence matrix of the uncontrollable subnet (obtained removing all controllable transitions). The set of monitors corresponding to $(W, \vec{k})$ can be added to net without disabling any uncontrollable transition if all elements in $W^T \cdot C_u$ are less than zero, because in this case there will be no arcs going from the monitors to uncontrollable transitions.

If such is not the case, one can try to find a new constraint $(W_c, \vec{k_c})$ (where W_c is a matrix with the same dimension of W) such that: a) $W_c^T \cdot C_u$ has all elements less than zero; b) $\mathcal{M}(W_c, \vec{k}_c) \subseteq \mathcal{M}_c(W, \vec{k})$, i.e., such that all markings that are legal for $(W_c, \vec{k_c})$ are also legal for $(W, \vec{k})$. Note that the set of legal markings for the new constraint may be a strict subset of the set $\mathcal{M}_c(W, \vec{k})$ and thus the new constraint may prevent the net from reaching markings that are legal.

In [13] it was shown how one may try to find such a new set of constraints by performing row operations on the matrix $\begin{bmatrix} C_u \\ W^T \cdot C_u \end{bmatrix}$. The main idea is to add to the support of each constraint in W new places as will be shown in the next example.

Example 7. Let us consider the GMEC $(\vec{w}, k)$ requiring $M(p_1) + M(p_2) \leq 1$ and the net in Figure 2.(b) without places p_0 and p_M and their input/output arcs.

The uncontrollable incidence matrix in this case is $C_u = [C(\cdot, t_1)\ C(\cdot, t_2)\ C(\cdot, t_3)\ C(\cdot, t_4)]$. Since $\vec{w}^T \cdot C_u = (-1\ -1\ 1\ 1)$, the constraint cannot be enforced. Let us add places p_4 and p_5 to the constraint to obtain the new constraint $(\vec{w}_c, k_c)$ with $\vec{w}_c^T = (1\ 1\ 0\ 1\ 1)$ and $k_c = 1$.

$$C_u = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Now $\vec{w}_c^T \cdot C_u = (-1\ -1\ 0\ 0)$, hence this constraint can be enforced by a monitor. Note that all markings that satisfy the new constraint also satisfy the original one, since $M(p_1) + M(p_2) \leq M(p_1) + M(p_2) + M(p_4) + M(p_5) \leq 1$. The monitor corresponding to $(\vec{w}_c, k_c)$ is place p_M in Figure 2.(b). The constraint $(\vec{w}_c, k_c)$ from the marking in Figure 2.(a) prevents the firing of t_7 even if the marking $(0\ 0\ 0\ 1\ 1)^T$ reachable by firing t_7 is in $\mathcal{M}_c(\vec{w}, k)$.

It's worth comparing the solution obtained following Giua's approach (monitor p_0) and that obtained following Moody's approach (monitor p_M). Monitor p_0 is the best solution (it implements the maximally permissive control policy) if places $p_1, \cdots, p_5$ are control safe. If these places are not control safe (this property depends on the structure and marking of the rest of the net) this solution cannot be applied. Monitor p_M, instead, does not implement necessarily the maximally permissive control policy but can be applied regardless of the structure of the net.

6. Conclusions

The paper has present an example of the efficient use of Petri net structural techniques in supervisory control. The design of control structures for enforcing linear constraints on the reachability set of a net has been discussed and different approaches have been compared.

References

[1] R.K. Boel, L. Ben-Naoum, and V. Van Breusegem, *On forbidden state problems for colored closed controlled state machines*, Preprints of the 12th IFAC World Congress (Sidney, Australia), vol. 4, July 1993, pp. 161–164.

[2] ______, *On forbidden state problems for a class of controlled Petri nets*, IEEE Trans. on Automatic Control **40** (1995), no. 10, 1717–1731.

[3] H. Chen, *Synthesis of feedback control logic for controlled Petri nets with forward and backward conflict-free uncontrolled subnet*, Proc. 33rd IEEE Trans. on Decision and Control (Lake Buena Vista, FL), Dec 1994, pp. 3098–3103.

[4] A. Giua, F. DiCesare, and M. Silva, *Generalized mutual exclusion constraints on nets with uncontrollable transitions*, Proc. 1992 IEEE Int. Conf. on Systems, Man, and Cybernetics (Chicago, Illinois), October 1992, pp. 974–979.

[5] ______, *Petri net supervisors for generalized mutual exclusion constraints*, Proc. 12th IFAC World Congress (Sidney, Australia), July 1993, pp. I:267–270.

[6] L. E. Holloway and B. H. Krogh, *Synthesis of feedback control logic for a class of controlled Petri nets*, IEEE Trans. on Automatic Control **35** (1990), no. 5, 514–523.

[7] B. H. Krogh, *Controlled Petri nets and maximally permissive feedback logic*, Proc. 25th Annual Allerton Conference (1987), 317–326, University of Illinois, Urbana.

[8] B. H. Krogh and L. E. Holloway, *Synthesis of feedback control logic for discrete manufacturing systems*, Automatica **27** (1991), no. 4, 641–651.

[9] Y. Li and W.M. Wonham, *Control of vector discrete-event systems I – the base model*, IEEE Trans. on Automatic Control **38** (1993), no. 8, 1214–1227.

[10] ______, *Control of vector discrete-event systems II – controller synthesis*, IEEE Trans. on Automatic Control **39** (1994), no. 3, 512–531.

[11] M. Makungu, M. Barbeau, and R. St-Denis, *Synthesis of controllers with colored Petri nets*, Proc. 32nd Annual Allerton Conference (1994), 709–718, University of Illinois, Urbana.

[12] J.O. Moody and P.J. Antsaklis, *Petri net supervisors for DES in the presence of uncontrollable and unobservable transitions*, Proc. 33rd Annual Allerton Conference (Monticello, IL., USA.), October 1995.

[13] J.O. Moody, P.J. Antsaklis, and M.D. Lemmon, *Automated design of a Petri net feedback controller for a robotic assembly cell*, Proc. INRIA/IEEE Sym. on Emerging Technologies and Factory Automation (Paris, France), vol. 2, October 1995, pp. 117–128.

[14] T. Murata, *Petri nets: Properties, analysis and applications*, Proc. of the IEEE **77** (1989), no. 4, 541–580.

[15] J. L. Peterson, *Petri net theory and the modeling of systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[16] P. J. Ramadge and W. M. Wonham, *The control of discrete event systems*, Proc. of the IEEE **77** (1989), no. 1, 81–98.

[17] K. Yamalidou, J.O. Moody, M.D. Lemmon, and P.J. Antsaklis, *Feedback control of Petri nets based on place invariants*, Automatica **32** (1996), no. 1.

Controller Synthesis by Means of Symbolic Backward Search

H.-M. Hanisch, A. Lüder, J. Thieme

Otto-von-Guericke University Magdeburg   Department of Electrical Engineering

PO-Box 4120   39016 Magdeburg   Germany

☎ +49 391 67 127 47   email: {hami|jan|arndt} @hamlet.et.uni-magdeburg.de

Abstract: *We present an idea for controller synthesis for a represention of Net Condition/Event systems which model the plant and which are based on bounded Petri nets. The specification is given by means of forbidden states. Our synthesis algorithm uses a backward search in the net graph of the plant model, the complete enumeration of all states is required only in the worst case. The method is illustrated by means of a resource allocation problem.*

Keywords: controller synthesis, forbidden states, Petri nets, Condition/Event systems

1 Introduction

The Supervisory Control Theory initiated by Ramadge and Wonham [8] is currently the most popular method for synthesis of controllers for Discrete Event Dynamic Systems (abbr.: DEDS). It provides numerous and very elegant theoretical results, and by the extension to Vector Discrete Event Systems it is possible to synthesize non binary systems [7]. The application of this theory to control problems of realistic scale, however, is rather sparse. We think that this is due to the inherent complexity problem and the lack of a clear concept of input and output signals.

The complexity problem can be overcome by means of Petri nets which provide an implicite description of large state spaces. Unfortunately, the classical concept of Petri nets does not provide an explicite notation of input and output signals in the way it is in general needed when dealing with control systems. Although there is a very large amount of different concepts to provide input and output signals for Petri nets (we cite here only a very fundamental work [6]), such concepts are mostly only descriptive techniques, and analysis or synthesis of controllers based on these concepts is often impossible. An exception are controlled Petri nets [5], which are, however, neither modular nor hierarchical.

Hence, in the last two years we have been developing a modular modelling technique which combines the very clear input/output concept of Condition/Event systems [10] with the implicite description of state space by means of Petri nets. As a result, a class of discrete event models has been defined which we call Net Condition/Event systems (abbr.: NCES) [4, 9]. Since we were mainly interested in binary systems, almost all of our work is based on the assumption that our underlying Petri net is safe. The only exception is [2]. Here we tried to develop a model which uses bounded Petri nets and (condition) signals with multiple value. Synthesis procedures, however, are not provided.

We want to come back to this model in our paper and want to provide a method for controller synthesis which is an extention of our methods for controller synthesis for forbidden states [3].

For this purpose, our paper is organized as follows. We will briefly introduce the definition of NCES using a set/function notation and our running example in Section 2. Section 3 sketches the idea of the algorithm and demonstrates it by means of our running example. Finally we draw some conclusions and give some directions for further work which seem to be necessary from our point of view.

2 Net Condition/Event Models with Multiple Condition Signals

2.1 Basic Notations

We define a Net Condition/Event system with multiple condition signals as a tuple

$$NCES_m = \{P, T, F, CN, EN, C^{in}, E^{in}, C^{out}, E^{out}, Bc, Be, Cs, Dt, W, m_0\} \text{ where} \qquad (1)$$

$P = \{p_1, \ldots, p_n\}$ is the ordered set of places

$T = \{t_1, \ldots, t_m\}$ is the ordered set of transitions t

$F \subseteq (P \times T) \cup (T \times P)$ are the arcs

$CN \subseteq P \times T$ are the condition signal arcs

$EN \subseteq T \times T$ are the event signal arcs
$C^{in} = \{c_1^{in}, \ldots, c_r^{in}\}$ is the ordered set of condition input signals
$E^{in} = \{e_1^{in}, \ldots, e_s^{in}\}$ is the ordered set of event input signals
$C^{out} = \{c_1^{out}, \ldots, c_k^{out}\}$ is the ordered set of condition output signals
$E^{out} = \{e_1^{out}, \ldots, e_l^{out}\}$ is the ordered set of event output signals
$Bc \subseteq C^{in} \times T$ are the condition input arcs
$Be \subseteq E^{in} \times T$ are the event input arcs
$Cs \subseteq P \times C^{out}$ are the condition output arcs
$Dt \subseteq T \times E^{out}$ are the event output arcs
$W : F \cup CN \cup Bc \cup Cs \rightarrow \mathcal{N}$ is the weight function of arcs and condition signals
$m_0 : P \rightarrow \mathcal{N},$ is the initial marking.

Conditon and event signals between places and transitions or transitions and transitions, respectively, result from composition of ordinary Petri nets which are extended by incoming and outgoing signals and are denoted by special arc symbols (see Figure 1). Signals have no effect on the behaviour of the sending part but only on the receiving part. This assumption is usual in control systems, and we want to deal with control systems here. Both types of signals affect the firing of receiving transitions in the following way. A transition t can fire ***spontaneously*** if it has no incoming event signal, if all places p with $(p,t) \in F \cup CN$ contain no less than $W((p,t))$ tokens and all condition inputs c^{in} with $(c^{in}, t) \in Bc$ have values no less than $W((c^{in}, t))$. A transition is *forced to fire* if the above property for all arcs and condition signals from places and inputs hold and if a transition fires, which sends an event signal, that is received by t. For sake of simplicity, we restrict ourselfes in this paper on at most one incoming event signal per transition. We see that event signals establish an asymmetric synchronisation relation between transitions which goes beyond the definition of classical Petri nets, but it is needed in our modular modelling concept and justified in [9]. The interaction of the model with its environment is described by incoming and outgoing condition and event signals too, which are denoted by squares for condition signals and diamonds for event signals at the end of the arc which represents the environment. A particular part of the environment of a NCES model of a plant is a controller, which receives signals from the plant and sends signals to the plant. By means of introduction of signals, we get a setting which is usually used in control systems. That means that there is ***no*** flow of tokens between the plant and the controller.

2.2 Example

We want to introduce a running example to illustrate our method. The example is described in detail in [2]. Let us assume that we have 3 chemical reactors which need cooling fluid to perform a chemical reaction and that the amount of cooling capacity is limited to 7 "units". Let us further assume that for sake of simplicity each reactor can either be in the state "waiting" or "reaction" (we ignore feeding and discharging the reactor since this would not impose additional restrictions to our example). We have two different types of reactors which need a different amount of cooling capacity. If more cooling capacity is required by the reactors than can be provided, the system breaks down, and wasted product is the result. Figure 1 shows the NCES model of the uncontrolled behaviour of the plant. The undesired (forbidden) states are denoted by facts.

Obviously, the plant should never reach a state in which any of the facts is enabled. It is obvious too, that this problem could be solved very easily by adding a "monitor place" to the system. This makes sense when a computer program is to be designed that must have a correct behaviour. But, as mentioned before, we deal with control problems where we have plants that process material or energy, and it is not possible to install a physical instance ***in the plant*** that acts like a monitor. This function must be performed by a controller which uses the ***signals*** that are provided by measuring devices in the plant as inputs and which gives control signals to the actuators in the plant which can start or stop reaction in the reactors. We will show in the following section how such a controller can be synthesized by means of a backward search in the graph of the plant model without the need to generate the complete state space.

3 Controller Synthesis

3.1 Preparations

At first we generate expressions describing the pre- and postrange of transitions. With respect to the prerange, we have to divide the prerange in such a range where tokens flow and in a range which provides

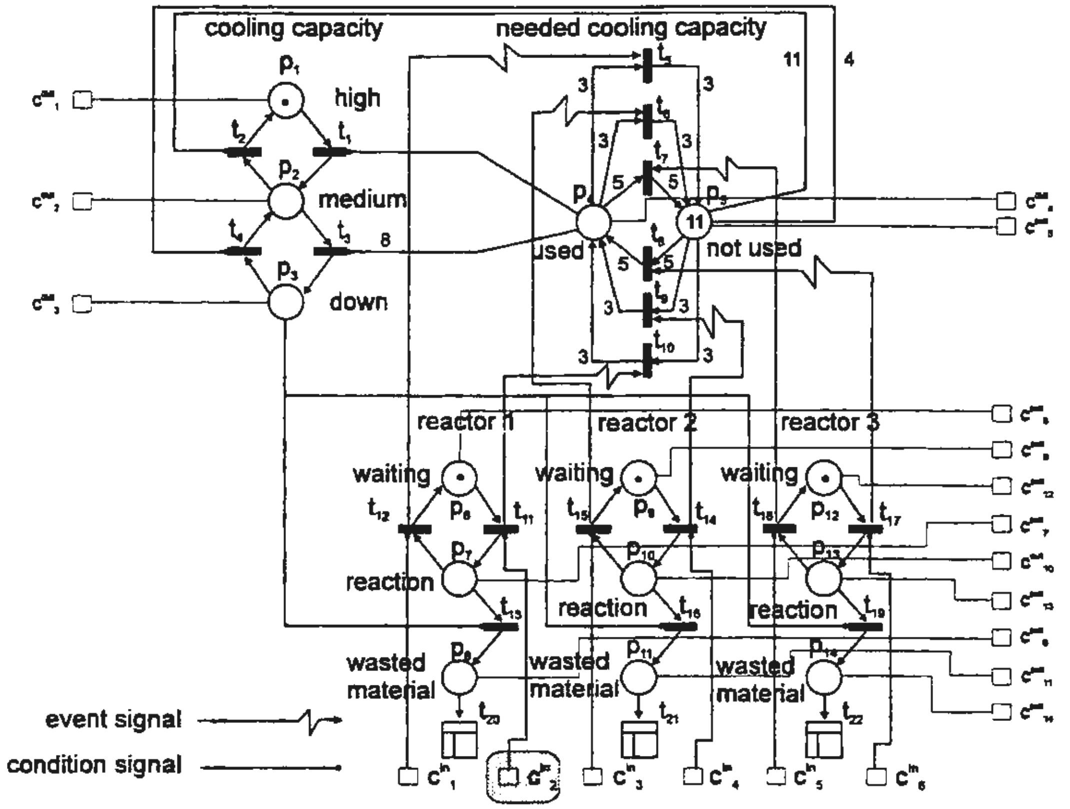


Figure 1: Plant model

necessary condition signals. In both cases we have to consider the influence of event signals.

$PA^-_M(p,t)$ is a set expression over the marking of places and activity of inputs called token flow prerange. We define $PA^-_M(p,t)$ recursively:

$$\forall p \in P, \forall t \in T: \ (t,p) \in F \quad PA^-_M(p,t) = \bigcup_{\substack{W((p_i,t)) = k > 0\\ (p_i,t) \in F}} p_i^k \cup \bigcup_{(t_i,t) \in EN} PA^-_M(\bullet, t_i) \tag{2}$$

$PA^-_C(p,t)$ is a set expression over the marking of places and activity of inputs called condition prerange. We define $PA^-_C(p,t)$ recursively:

$$\forall p \in P, \forall t \in T: \ (t,p) \in F \quad PA^-_C(p,t) = \bigcup_{\substack{W((p_i,t)) = k > 0\\ (p_i,t) \in CN}} p_j^k \cup \bigcup_{\substack{W((c_i^{in},t)) = k > 0\\ (c_i^{in},t) \in Bc}} {c_i^{in}}^k \cup \bigcup_{(e_i^{in},t) \in Be} e_i^{in} \cup \bigcup_{(t_i,t) \in EN} PA^-_C(\bullet, t_i) \tag{3}$$

$PA^+(p,t)$ is a set expression over the marking of places called postrange. We define $PA^+(p,t)$ recursively:

$$\forall p \in P, \forall t \in T: \ (t,p) \in F \quad PA^+(t,p) = \bigcup_{\substack{W((t,p_i)) = k > 0\\ (t,p_i) \in F}} p_i^k \cup \bigcup_{(t_i,t) \in EN} PA^+(t_i, \bullet) \tag{4}$$

The recursion depth is always finite since cycles of event signals are forbidden for all three set expressions. During our backward search algorithm we will use and generate parts of marking vectors. It is convenient for our purpose to represent this information by multisets since we do not need to process places whose markings are not relevant. Hence, in the following, we will use p_i^n which denotes the element p_i is n-times element of a multiset as an abbrevation for $m(p_i) = n$. To process multisets in the way we intend, three operators must be defined.

DEFINITION 1 *The union operator* $\otimes$ *is defined as follows: Let* $A = \{a_1, a_2, \dots, a_n\}$, $B = \{b_1, b_2, \dots, b_m\}$ *be two sets where each element is a set itself. Then:*
$A \otimes B = \{a_1 \cup b_1, a_1 \cup b_2, \dots, a_1 \cup b_m, a_2 \cup b_1, a_2 \cup b_2, \dots, a_2 \cup b_m, \dots, a_n \cup b_1, a_n \cup b_2, \dots, a_n \cup b_m\}$

This operator corresponds to the AND-connection of BOOLEan functions representing ordinary sets. If we had for example two BOOLEan functions $f_A = p_i \vee p_j$ und $f_B = p_k \vee p_l$ which are characteristic functions of two ordinary sets A and B, $A \otimes B$ would be given by $f_A \wedge f_B = p_ip_k \vee p_ip_l \vee p_jp_k \vee p_jp_l$. For our multiset representation with $A = \{\{p_i\}, \{p_j\}\}$ and $B = \{\{p_k\}, \{p_l\}\}$ we get $A \otimes B = \{\{p_ip_k\}, \{p_ip_l\}, \{p_jp_k\}, \{p_jp_l\}\}$.

DEFINITION 2 *The set reduction operator* $\ominus_1$ *is defined as follows: Let* $A = \{a_1, a_2, \dots, a_n\}$ *be a set where each element is a multiset, and let* a_{n+1} *be a multiset. Then:* $A \ominus_1 a_{n+1} = \{a_{i'} | a_{i'} = a_i \backslash (a_i \cap a_{n+1}),\ i \leq n\}$

In a BOOLEan representation of characteristic functions this corresponds to assigning the constant value one to all elements of a_{n+1} in f_A. For example if $f_A = p_ip_j \vee p_kp_l$ and $a_{n+1} = p_k$, the resulting characteristic function f'_A would be $f'_A = p_ip_j \vee p_l$. With multisets $A = \{\{p_ip_j\}, \{p_kp_l\}\}$ and $a_{n+1} = \{p_k\}$ we get $A \ominus_1 a_{n+1} = \{\{p_ip_j\}, \{p_l\}\}$.

DEFINITION 3 *The set reduction operator* $\ominus_2$ *is defined as follows: Let* $A = \{a_1, a_2, \dots, a_n\}$ *be a set where each element is a multiset, and let* a_{n+1} *be a set. Then:* $A \ominus_2 a_{n+1} = \{a_i | a_i \cap a_{n+1} = \emptyset,\ i \leq n\}$

In a BOOLEan representation of characteristic functions this corresponds to assigning the constant value zero to all elements of a_{n+1} in f_A. For example if $f_A = p_ip_j \vee p_kp_l$ and $a_{n+1} = p_k$, the resulting characteristic function f'_A would be $f'_A = p_ip_j$. With multisets $A = \{\{p_ip_j\}, \{p_kp_l\}\}$ and $a_{n+1} = \{p_k\}$ we get $A \ominus_1 a_{n+1} = \{\{p_ip_j\}\}$.

3.2 Synthesis algorithm

Our synthesis method is very similar to the method proposed in [3] for binary NCES. The extension is that we have to handle multisets instead of BOOLEan functions which describe ordinary sets.
1. We compute the enabling terms $\Sigma(P)$ of the facts. We set this to zero to describe that the facts must never become enabled.

$$\Sigma(P) = \bigcup_{f \in T} \bigotimes_{(p,f) \in F} \{p\} \quad = 0 \tag{5}$$

Each element of $\Sigma(P)$ is called a forbidden marking. In fact, each element denotes a whole set of markings since the elements of $\Sigma(P)$ contain only the relevant places. The markings of the other places are "don't care".
2. For each place p and each transition t of the net we compute the expressions $PA^-_M(p,t)$, $PA^-_C(p,t)$ and $PA^+(p,t)$.
3. To ensure that a fact never becomes enabled we perform a backward search algorithm as follows:

```
ALGORITHM 1
Input:  $\Sigma(P)$
        $PA^-_M(p,t)$, $PA^-_C(p,t)$ and $PA^+(p,t)$      $\forall p$
        decision function $D : \sigma \subseteq (P \cup C^{in} \cup E^{in})^\lambda \to \{0,1\}$
S.0: $\Sigma'(P) = \Sigma(P)$, $\Sigma^*(P, C^{in}, E^{in}) = \emptyset$
S.1: CHOOSE $\sigma \in \Sigma'(P)$
     $\Sigma'(P) = \Sigma'(P) \backslash \{\sigma\}$, $\Sigma^*(P, C^{in}, E^{in}) = \Sigma^*(P, C^{in}, E^{in}) \cup \{\sigma\}$
     IF $\sigma \cap (C^{in} \cup E^{in}) \neq \emptyset$ THEN GOTO S.1
S.2: FOR $p \in \sigma$ DO
        FOR $t$ : $(t,p) \in F$ DO
           $\sigma' = \sigma$
(*1*)      FOR $\hat{p}^k \in PA^+(t,p)$ with $k = \max_{\hat{p}^\mu \in PA^+(t,p)} \{\mu\}$      (* Start Backward Firing *)
              IF $l < k$ with $l = \max_{\hat{p}^\mu \in \sigma} \{\mu\}$ THEN $\sigma' = \sigma' \backslash \{\hat{p}^l\}$
              ELSE $\sigma' = \sigma' \backslash \{\hat{p}^k\}$
           END_FOR
(*2*)      $\sigma' = \sigma' \cup PA^-_M(t,p)$
```

```
(*3*)         FOR $\hat{p}^k \in PA_C^-(t,p)$ with $k = \max_{\hat{p}^\mu \in PA_C^-(t,p)} \{\mu\}$
                  IF $l < k$ with $l = \max_{\hat{p}^\mu \in \sigma'} \{\mu\}$ THEN $\sigma' = \sigma' \cup \{\hat{p}^{k-l}\}$
              END_FOR                                        (* End Backward Firing *)
(*4*)         IF $D(\sigma') = 1$ THEN $\Sigma'(P) = \Sigma'(P) \cup \{\sigma'\}$
          END_FOR
     END_FOR
S.3: IF $\Sigma'(P) \neq \emptyset$ THEN GOTO S.1
END
Output:       $\Sigma^*(P, C^{in}, E^{in})$
```

Step 0 initializes sets we create during our backward search. Step 1 up to step 3 denotes the following. We replace each element of $\Sigma'(P)$ by all predecessor markings if we can not guarantee by controllable inputs that we can prevent the system from reaching this particular forbidden marking. This main part of the algorithm realizes a "backward" firing of transitions. We have to distinguish different arcs, namely the arcs denoting flow of tokens and the arcs expressing condition signals which additionally enable transitions.
If a transition is fired backward, it obviously adds the tokens which it would remove by firing forward. Tbis is denoted by PM_M^- (*2*). Additionally, it sets all its incoming condition signals denoted by PA_C^- to the value demanded by the condition arc weights (*3*). That means that the number of tokens at places which are connected with the transition by condition signals is unchanged if there are at least as much tokens on it as the arc weight denotes, or set to the weight of the condition arc from the place to the transition, otherwise. On the other hand, the backward firing of a transition removes all tokens it would add by forward firing. This is denoted by PA^+ (*1*).
By this backward search algorithm we generate systematically all partial markings which could eventually cause a marking in which a fact is enabled. The backward search terminates when we find a partial marking which contains also an input signal from the controller since we can prevent by setting this input signal to zero that a transition would be enabled which could actually lead to a forbidden state.
4. The resulting set $\Sigma^*(P, C^{in}, E^{in})$ contains two different types of multisets.

$$\Sigma^*(P, C^{in}, E^{in}) = \widehat{\Sigma}(P) \cup \widehat{\Sigma}(P, C^{in}, E^{in}) \tag{6}$$

All multisets of the first type are part of $\widehat{\Sigma}(P)$. These multisets do not contain controllable inputs and represent partial states from which we can not guarantee by a controller that no forbidden state is ever reached. Hence, we have to check the expression:

$$\emptyset \notin \widehat{\Sigma}(P) \ominus_1 m_0 \text{ where } \widehat{\Sigma}(P) = \Sigma^*(P, C^{in}, E^{in}) \ominus_2 (C^{in} \cup E^{in}) \tag{7}$$

If equation 7 holds, no partial state denoted by an element of $\widehat{\Sigma}(P)$ is part of the initial marking.
5. The second type of multisets in $\Sigma^*(P, C^{in}, E^{in})$ is contained in $\widehat{\Sigma}(P, C^{in}, E^{in})$. These multisets contain at least one controllable input and describe the controller. The controller has to react at a moment when the net is marked with m_1 and the condition

$$\emptyset \quad \in \quad \left[\widehat{\Sigma}(P, C^{in}, E^{in}) \ominus_1 (C_C^{in} \cup E_C^{in})\right] \ominus_1 m_1 \tag{8}$$

holds. In this case, the controller has to set an input belonging to the multiset to zero. This leads to the controller:

$$c_i^{in} = \begin{cases} 0 & \text{if } \emptyset \in \left[\left(\widehat{\Sigma}(P, C^{in}, E^{in}) \ominus_2 \left((C^{in} \cup E^{in}) \setminus \{c_i^{in}\}\right)\right) \ominus_1 \{c_i^{in}\}\right] \ominus_1 m_1 \\ - & \text{otherwise} \end{cases}$$

$$e_i^{in} = \begin{cases} 0 & \text{if } \emptyset \in \left[\left(\widehat{\Sigma}(P, C^{in}, E^{in}) \ominus_2 \left((C^{in} \cup E^{in}) \setminus \{e_i^{in}\}\right)\right) \ominus_1 \{e_i^{in}\}\right] \ominus_1 m_1 \\ - & \text{otherwise} \end{cases}$$

We use a decision function D in our algorithm (*4*). This function is used to determine whether a multiset generated in step 2 of the algorithm has to be considered further or not. For example, the decision function checks whether σ' has been computed before or not and cbecks if a multiset contradicts structural properties of the net. At the moment we use only place invariants. We assume, that all invariants are calculated and are given in the form as defined in [1]. We use the following lemma:

LEMMA 1 *If x is a place invariant and m_0 is the initial marking of the net, than no marking m_1 can be reached with $\sigma' \subseteq m_1$ if: $\underline{m_0} \cdot x < \underline{\sigma'} \cdot x$ with $\underline{m_0}_j = k$ if $k = \max_{p_j^\mu \in m_0} \mu$ and $\underline{\sigma'}_j = k$ if $k = \max_{p_j^\mu \in \sigma'} \mu$.*

The prove follows directly from Theorem 6.48 in [1].
In a first variant we define D as follows:

$$D(\sigma') = \begin{cases} 0 & \text{if the property of lemma 1 holds or } \sigma' \text{ has been computed before} \\ 1 & \text{otherwise} \end{cases}$$

EXAMPLE 1 In our example we get for controllable input c_2^{in} the following:

$$c_2^{in} = \begin{cases} 0 & \text{if } \emptyset \in \left\{ \begin{array}{c} \{p_6 p_5^3 p_4^5 p_2\}, \{p_{10} p_6 p_5^3 p_4^5 p_2\}, \{p_{13} p_6 p_5^3 p_4^5 p_2\}, \{p_6 p_4^8 p_2\}, \{p_6 p_3\}, \\ \{p_6 p_5^3 p_4^5 p_1\}, \{p_{10} p_6 p_5^3 p_4^5 p_1\}, \{p_{13} p_6 p_5^3 p_4^5 p_1\} \{p_6 p_4^8 p_1\}, \end{array} \right\} \ominus_1 m_1 \\ - & \text{otherwise} \end{cases}$$

We see in our example that the multisets which contain $p_5^3 p_4^5$ disable the state transition from waiting to reaction of reactor 1. Otherwise, 8 units of cooling capacity could be required which would eventually cause a wasted batch. We also see that there are multisets in our controller function which can not be reached in the controlled system (for example $\{p_6 p_4^8 p_1\}$). Hence, these multisets will never become active in the controller function, and the controller is overspecified. This follows from our backward search and could in general only be avoided if we would know all reachable states. We do not want to assume, however, the knowledge of the complete state space, and the price we have to pay is an overspecified and sometimes not maximally permissive controller. By identifying the places and their output signals (see [3] for details), we obtain an interconnection of plant and controller by signals only. The controller functions of the other controllable inputs are similar.

4 Conclusions and Further Work

We have introduced an algorithm for controller synthesis based on bounded Petri net representation of Condition/Event systems modelling the uncontrolled behaviour of the plant and the specification. The class of problems we can solve is larger than in our priour work. In contrast to our previous work we are able to deal with allocation problems for non exclusive use of resources.

Further work must deal with proving formally that every controller synthesized by the algorithm ensures safe behaviour of the plant. Another inportant issue is to include more structural knowledge into our algorithm to reduce the gap between the set of states generated by backward search and the set of reachable states.

References

[1] B. Baumgarten. Petri-Netze, Grundlagen und Anwendungen. B.I. Wissenschaftsverlag, Mannheim, Wien, Zürich, 1990

[2] M. Rausch and H.-M. Hanisch. Net Condition/Event Systems with Multiple Condition Outputs. *Symposium on Emerging Technologies and Factory Automation* (ETFA'95), Paris, France, October 1995, INRIA/IEEE, 1995, vol. 1, pp. 592–600.

[3] H.-M. Hanisch, A. Lüder, and M. Rausch. Controller Synthesis for Net Condition/Event Systems with Incomplete State Observation. *Computer Integrated Manufactoring and Automation Technologie* CIMAT'96, Grenoble, France, May 1996.

[4] H.-M. Hanisch, S. Kölbel and M. Rausch. A Modular Modelling, Controller Synthesis and Control Code Generation Framework. 13th IFAC World Congress, San Francisco, July 1996, Preprints, Volume J, pp. 495-500.

[5] L. E. Holloway and B. H. Krogh. *Synthesis of feedback control logic for class of controlled Petri nets representing discrete-event systems; application to automated guided vehicle coordination.* IEEE Transactions on Automatic Control, 35 (1990) 5, pp. 514–523.

[6] R. König and L. Quäck. Petri-Netze in der Steuerungstechnik. Verlag Technik, 1.Edition, Berlin, 1988

[7] Y. Li, W.M. Wonham. Control of Vector Discrete Event Systems 1 - The Base Model. IEEE Transactions on Automatic Control, Vol. 38, No. 8, pp.1214-1227, 1993.

[8] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.

[9] M. Rausch. Modulare Modellbildung, Synthese und Codegenerierung Ereignisdiskreter Steuerungssysteme. PhD Thesis, University of Magdeburg, Department of Electrical Engineering, 1996.

[10] R. S. Sreenivas and B. H. Krogh. On Condition/Event Systems with Discrete State Realizations. *Discrete Event Dynamic Systems: Theory and Applications*, 2(1):209–236, 1991.

ON EXPLOITING THE ANALYSIS POWER OF PETRI NETS FOR THE VALIDATION OF DISCRETE EVENT SYSTEMS

Monika Heiner
Brandenburg University of Technology at Cottbus, Computer Science Institute
Postbox 101344, D-03013 Cottbus, Germany
mh@informatik.tu-cottbus.de, http://www.informatik.tu-cottbus.de

Abstract: The development of provably error-free concurrent systems is still a challenge of practical system engineering. Modelling and analysis of concurrent systems by means of Petri nets is one of the well-known approaches using formal methods. Among those Petri net analysis techniques suitable for strong verification purposes there is an increasing amount of promising methods avoiding the construction of the complete interleaving state space, and by this way the well-known state explosion problem. This paper claims to demonstrate that the available methods and tools are actually applicable successfully to at least medium-sized systems. For that purpose, the step-wise validation of various system properties (consistency, safety, progress) of the concurrent control software of a reactive system is performed. If possible, different analysis techniques are applied and compared with each other concerning its efforts.

keywords: programmable logic controller, hierarchical place/transition nets, verification, temporal logics, model checking, interval nets;

1 Introduction

Petri nets enjoy several advantages with respect to modelling and analysis of discrete event systems with inherent concurrency. Worth mentioning is especially the ability of combining different methods on a common representation. This variety ranges from informal (animation) via semi-formal (systematic testing) up to formal (exhaustive analysis) methods and comprises qualitative as well as quantitative evaluation techniques. But maybe most valuable is the fact that among the formal methods suitable for strong verification purposes there is an increasing amount of promising methods avoiding the construction of the complete interleaving state space, and by this way the well-known state explosion problem.

This paper gives an overview on these methods and demonstrates their strength and limitations by a running example. The discussion covers

- **static analysis techniques**, constructing no state space at all,
  e.g. structural properties allowing conclusions on behavioural properties, linear-algebraic analysis revealing invariants, local reduction rules to minimize the net structure,
- **lazy state space construction**, building reduced (interleaving) state spaces, which are generally much smaller than the complete state space for highly concurrent systems,
  e.g. stubborn set reduction to decide deadlock freedom or un-/reachability of special states, and to prove the validity of formulae in a nexttime-free linear time temporal logic (LTL\X),
- **alternative state space construction**, exploiting concurrency to build partial order (true concurrency) descriptions of the system behaviour,
  e.g. finite prefix of branching processes for model checking, and - just emerging - concurrent automata (CA), combining the advantages of reachability graphs and branching processes (the nodes are global states; the arcs are labelled with semi-words of transition events; each branching corresponds actually to a conflict;).

As example serves an adopted version of the pusher problem for which in [7] a control program has been synthesized automatically. By this way, this paper presents a reversal check for that synthesis. General transformation rules are sketched to transform programmable logic controller (plc) programs into ordinary place/ transition Petri nets. Therefore, the rich amount of Petri net analysis techniques and tools can be applied for computer-aided analysis of plc programs.

The tool kit used comprises PED (hierarchical Petri net editor) [5], INA (structural properties, place/transition invariants, stubborn set reduced deadlock and reachability analysis, net reductions) [9], PROD (stubborn set reduced deadlock analysis and model checking - LTL\X) [10], and PEP (prefix-based model checking - CTL_0, linear-algebraic analysis) [1]. For short descriptions of the tool features which have been proven to be suitable see e.g. [2]. More details can be found in the referred tool manuals.

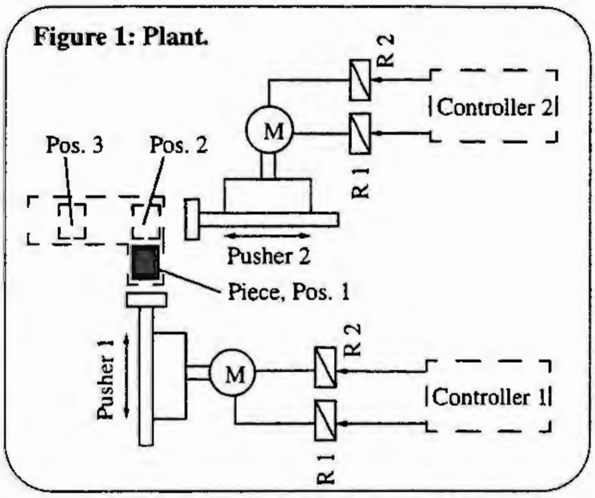
2 Task Description

To make the paper self-contained, the running example, adopted from [7], is shortly sketched.

The example consists basically of two concurrently working pushers moving work pieces (see figure 1). The work piece is moved from position one to position two by the first pusher, and from position two to position three by the second pusher. Both pushers are driven by electric motors which can be controlled by corresponding relays into two moving directions.

Starting from this basic situation, chains of concurrent pushers may be constructed in order to move pieces step by step from the input position via a number of inner positions to the output position.

Figure 1: Plant.



3 Requirement Specification

In addition to the task description, a list of informally specified safety and progress properties is given. Typical properties of this type are:

(a) **safety**

- At any time, a pusher can be driven in one direction only.
- To avoid collisions, it is not allowed to move adjacent pushers at the same time.
- No pusher motion must be driven too far/near.
- While moving a pusher, a new work piece must not arrive in its input position.

(b) **progress**

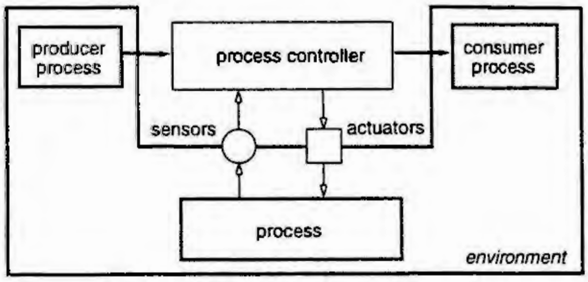
- After an active phase of a pusher, its successor will be activated before the predecessor will be started again.
- It is guaranteed that each pusher works infinitely often (livelock freedom).
- Any work piece entering the plant will finally leave the plant.

(c) **consistency**

Additional properties to be verified emerge during modelling reflecting useful (self-) consistency checks (see section 4.1).

4 Modelling with Hierarchical Petri Nets

The model of the total system may be characterized by a strong separation of control program and environment into different parts. The control program consists generally of a finite and static set of communicating processes. The environment model is composed of small reusable components: the producer/consumer processes of the work flow, and the devices of the controlled process.



4.1 Environment Model

For each device type exists a net component - building step-by-step a growing reusable component library to describe the uncontrolled plant behaviour. Each physical device is basically characterized by its finite set of discrete states, and additionally by the commands (externally visible transitions - they grey ones) forcing the device to change its current state (see fig. 3). Obviously, each device must be in one and only one state at any time. In terms of Petri net theory, the states of a device form a place invariant. In our example, there are two types of devices (relays, pushers). Accordingly, there are two consistency conditions. E.g. it holds for all pushers *Pi*:

(P1) $(Pi\_too\_near + Pi\_basic + Pi\_norm + Pi\_ext + Pi\_too\_far) = 1$

or expressed as temporal formula ($\dot{\vee}$ stands for exclusive or):

(P1*) $\mathbf{AG}\ \left(Pi\_too\_near\ \dot{\vee}\ Pi\_basic\ \dot{\vee}\ Pi\_norm\ \dot{\vee}\ Pi\_ext\ \dot{\vee}\ Pi\_too\_far\right)$

The composite model is structured into two layers. The top layer of a transport system with two pushers (Figure 2) consists of six macro components. Each macro transition P1 and P2 contains the process environment model given in figure 3, but prefixed with the instance names P1 or P2, respectively. For a more systematic analysis procedure (see section 5 and section 6), two versions of pushers are considered: without and with explicit error states (too_near, too_far). In the initial state (marking), all relays are off, the pushers are in their basic positions, and the plant is empty (contains no work piece).

4.2 Control Program Model

The pattern of the essential parts of the controller macro transitions (con1, con2) is given in Figure 4. The original plc programs are written in IEC 1131-3 (see left part). These programs are (automatically) translated into ordinary place/transition nets. For an example, how this could look like, consider the right side in Figure 4.

In order to avoid unnecessary restrictions of the concurrency degree, it could be helpful to exploit a special test arc feature for modelling of the transitions' side conditions. In that case, the amount of data, which has to be searched through during the analysis steps, may become much smaller, provided the analysis tools are prepared to handle test arcs.

4.3 Requirement Specification in Model Terms

Finally, the informally given requirement specifications have to be transformed into the terms of the formal model.

(a) safety

- At any time, a pusher can be driven in one direction only:

(P2) $\mathbf{AG}\ (\neg(Pi\_R1\_on \wedge Pi\_R2\_on))\ ,\ \forall i$

- To avoid collisions, it is not allowed to move adjacent pushers at the same time:

(P3) $\left(\sum_{i=1}^{2} Pj\_Ri\_on + \sum_{i=1}^{2} Pk\_Ri\_on\right) \leq 1\ \ ,\forall i, \forall j, k : j + 1 =$

- No pusher motion must be driven too far/near:

(P4) $\mathbf{AG}\ (\neg Pi\_too\_near)\ ,\ \forall i$

(P5) $\mathbf{AG}\ (\neg Pi\_too\_far)\ ,\ \forall i$

- While moving a pusher, a new work piece must not arrive in its input position:

(P6) $\mathbf{AG}\ (posi\_full \rightarrow Pi\_basic)\ ,\ \forall i$

(b) progress

- After an active phase of a pusher, its successor will be activated before the predecessor will be started again:

(P7) $\mathbf{AG}\ (Pi\_norm \vee Pi\_ext \rightarrow$

$\mathbf{AF}\ (\neg(Pi\_norm \vee Pi\_ext)\ \mathbf{AU}\ (Pj\_norm \vee Pj\_ext))\ )\ ,\ \forall i, j : i + 1 = j$

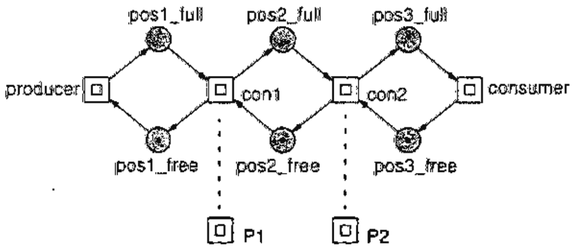
- It is guaranteed that each pusher works infinitely often (livelock freedom), e.g. (*en(t)* stands for the conjunction of all preplaces of *t*:

(P8) $\mathbf{AG}\ (\mathbf{AF}\ (en(Pi\_basic2norm)))\ ,\ \forall i$

- Any work piece entering the plant will finally leave the plant (which may be considered as a consequence of (P7) and (P8)), i.e. in case of a two-pusher chain:

(P9) $\mathbf{AG}\ (pos1\_full \rightarrow \mathbf{AF}\ pos3\_full)$

Figure 2: Top layer of a transportation system with two pushers.



Drawing convention:
Shadowed nodes are so-called logical (fusion) nodes. They serve as connectors to avoid immoderate edge crossing. All logical nodes with the same name are logically identical.

Figure 3: Process environment model of each controller.

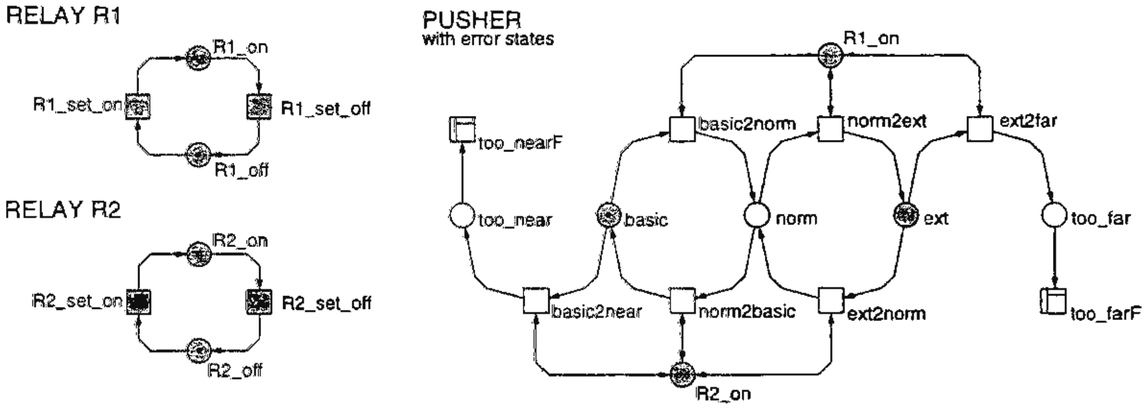
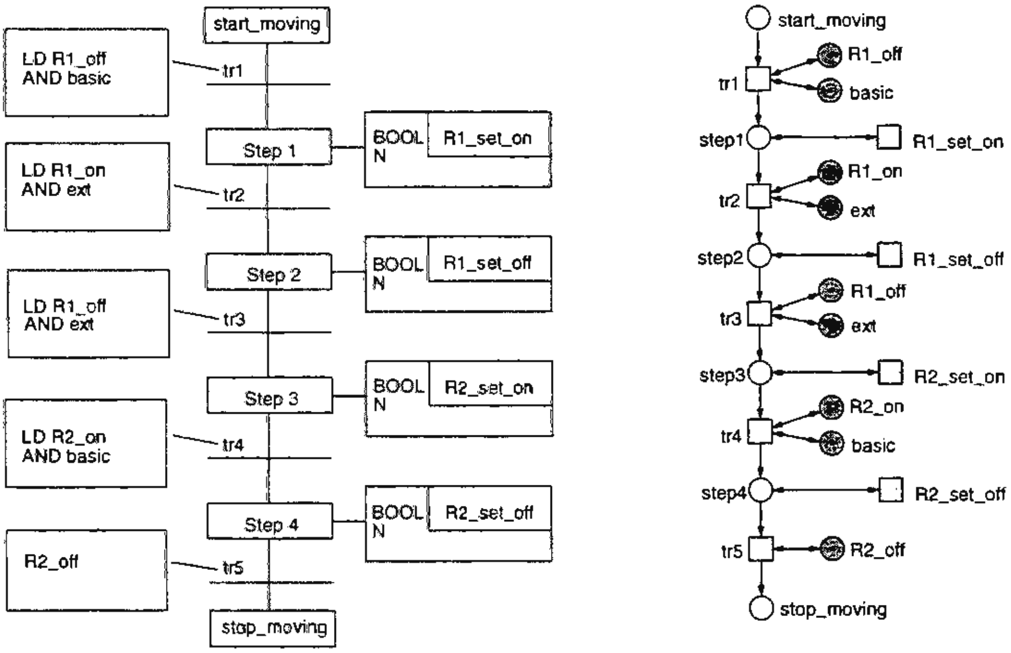


Figure 4: Part of the controller program and its Petri net model.



5 Qualitative Analysis

We present a two-step analysis. At first, the pusher model without explicit error states is discussed in this section. Afterwards, the error states are integrated leading to the notion of time (see section 6).

5.1 General Analysis

General analysis deals with properties which should be valid independently of the intended functional behaviour of the system. Basically, these are boundedness and liveness.

boundedness: The net is covered by semi-positive place invariants (INA). Moreover, the token sum of all these place invariants equals to 1. So we are able to conclude the 1-boundedness of the net (a necessary precondition for PEP's model checker).

liveness: The deadlock freedom can be proven efficiently by construction of stubborn set reduced reachability graphs (INA, PROD), which are generally much smaller than the complete state space. Additionally, it can be shown efficiently that the net is covered by semi-positive transition invariants as necessary (but not sufficient) condition for liveness. But liveness (no dead system parts) can't be proven by classical Petri net theory for longer pusher chains, due to the lack of suitable net structures (the given nets are not Extended Simple, net reduction does not help), and due to the state explosion by considering all interleaving transition sequences (reachability graph). However, based on the branching processes' prefix, for each transition the liveness has been proven (PEP) by model checking the temporal formula: $\mathbf{AG\ EF}\ (en(t))$.

5.2 Special Analysis

(a) safety

There are different analysis techniques available to prove the unreachability of unsafe states (P2) - (P6):

Facts (INA): The unsafe states may be modelled as facts (special transitions which are expected to become never enabled). But, the evaluation of bad states (a state where a fact is enabled) by the given tool kit requires the reachability graph. That's why we will avoid this approach.

Stubborn set reduction (INA): The net is transformed in such a way that the unsafe states become dead states. Then the stubborn set reduced reachability graph has to be constructed. Because any dead states are preserved under this reduction, the original net does not contain any unsafe states if the transformed net does not reach any dead states. This technique could be useful if the required net transformation is done by the analysis tool.

Place invariants (INA): A sufficient condition for the unreachability of a given marking m is fulfilled if the there exists at least one place invariant x for which the token conservation equation

$$\sum_{p \in P} x(p) \cdot m_0(p) = \sum_{p \in P} x(p) \cdot m(p)$$

is not valid. To check this equation, complete markings must be specified. But unsafe states are usually given in terms of submarkings (containing "don't care" places). This main disadvantage is overcome in the next approach.

Trap equation (PEP): Based on a linear upper approximation of the state space, a sufficient condition for linear properties of the type $A \cdot m \leq b$ has been introduced in [4]. The implementation is integrated in the latest version of PEP. We use it to prove (P3).

Model checking of temporal formulae: Model checking, combined with stubborn set reduction (PROD, LTL\X) or based on the finite prefix of branching processes (PEP, CTL_0), provides generally the most convenient method to raise safety questions, esp. because set of (unsafe) states may be characterized in a concise manner. Both model checkers run very fast. Due to the evaluation method, they are applicable also to larger systems of which the size of the interleaving state space is unknown.

(b) progress

(P7) - (P9) use a richer set of (temporal) logical operators. Therefore, model checking facilities are unavoidable. Due to the **AF** and **AU** operators, these properties can be proven only by PROD. We use it to prove (P7) - (P9) for any pusher chain.

(c) consistency

For any pusher chain, (P1) is analyzable by INA, and in the version of (P1*) by PROD or PEP. But for larger systems, it is generally a cumbersome task to prove this type of properties by finding the suitable place invariants.

A summary on the analysis efforts necessary to gain the results mentioned above is given in the following table.

Table 1: **Overview on analysis efforts.**

# pushers	P / T	R	R_{stub}	prefix (B / E)	CA[a)], events
1	24 / 21	88	22	96 / 45	26
2	42 / 38	464	42	213 / 99	45
3	60 / 55	3.088	79	366 / 170	82
4	78 / 72	18.848	133	555 / 258	119
5	96 / 89	118.624	204	780 / 363	173
6	114 / 106	738.368	292	1041 / 485	228
7	132 / 123	4.614.208	397	1338 / 624	299
8	150 / 140	?	519	1671 / 780	372
9	168 / 157	?	658	2040 / 953	460
10	186 / 174	?	814	2445 / 1143	551

a) 2 nodes (global states) and 2 arcs (labelled with semi-words of events), for any pusher chain.

6 Quantitative Analysis

In case of explicit error states within the model (Pi_too_far, Pi_too_near), it has to be proven that a pusher, after having reached the expected extension, is switched off fast enough. Obviously, we have now to take into consideration also the timing behaviour of the given system.

In terms of interval Petri nets [6] this means that the error transitions modelling the pusher motions into unsafe states (Pi_ext2far, Pi_basic2near) may be enabled, but will never fire due to the influence of time. Therefore, the proof of the unreachability of explicit error states ((P4), (P5)) can be traced back to the proof that the related error transitions are dead at the initial state.

This may e.g. happen because the transitions Ci_tr2 and Pi_R1_set_off (disabling Pi_ext2far) fire always before Pi_ext2far is willingly to fire. Generally, a proof like that depends essentially on the chosen interval times (but can be done by INA, at least as long as the reachability graph fits into memory). But in this concrete case, we are able to conclude - by evaluating a suitable part of the reachability graph (or at best a non-interleaving version of it) - that for any time intervals for which the relations

$$lft\,(Ci\_tr2) < eft\,(Pi\_ext2far) \wedge lft\,(Ri\_set\_off) < eft\,(Pi\_ext2far)$$

hold, the dangerous transitions Pi_ext2far will never fire. Similar relations hold for Pi_basic2near.

7 Conclusions

All qualitative (i.e. timeless) properties have been proven without construction of the reachability graph (interleaving state space). Up to now, the quantitative (i.e. time-dependent) analysis of interval nets is based on reachability graph construction and evaluation. But in [8], a method has been proposed to describe the behaviour of interval nets by a finite prefix of branching processes. It seems to be worth thinking over how to combine both approaches. Nevertheless, all proves were carried out automatically by help of general Petri net analysis tools. Therefore, they are reproducible in an objective way.

References

[1] Best, E.; Grahlmann, B.: PEP - Programming Environment Based on Petri Nets, Documentation and User Guide; Univ. Hildesheim, Institut für Informatik, Nov. 1995.

[2] Heiner, M.; Deussen, P.; Spranger, J.: A Case Study in Developing Control Software of Manufacturing Systems with Hierarchical Petri Nets; Proc. 1st Int. Workshop on Manufacturing and Petri Nets held at ICATPN '96, Osaka, June '96, pp. 177-196.

[3] Heiner, M.; Popova-Zeugmann, P.: Worst-case Analysis of Concurrent Systems with Duration Interval Petri Nets; BTU Cottbus, Dep. of CS, Techn. Report I-02/1996, available on http://www.informatik.tu-cottbus.de.

[4] Melzer, S.; Esparza, J.: Checking System Properties via Integer Programming; ESOP '96, Linköping, LNCS 1058, pp. 250-264.

[5] PED, http://www-dssz.Informatik.TU-Cottbus.De/~wwwdssz/ped.html.

[6] Popova-Zeugmann, L.: On Time Petri Nets; J. Information Processing and Cybernetics EIK 27(91)4, pp. 227-244.

[7] Rausch, M.; Lüder; A.; Hanisch, H.-M.: Combined Synthesis of Locking and Sequential Controllers; Proc. WODES '96, Edinburgh/ UK, Aug. 1996, pp. 133-138.

[8] Semenov, A.; Yakovlev, A.:Verification of Asynchronous Circuits Using Petri Net Unfolding; Proc. DAC '96, Las Vegas, June 1996, pp. 59-63.

[9] Starke, P. H.: INA - Integrated Net Analyzer; Manual, Berlin 1992.

[10] Varpaaniemi, K. et al.: PROD Reference Manual; Helsinki Univ. of Technology, Digital Systems Laboratory, Series B: Techn. Report No. 13, August 1995.

EXTENDED NET CONDITION/EVENT SYSTEMS FOR MODELLING OF MANUFACTURING AND PROCESS TECHNOLOGY

M. Rausch
Carnegie Mellon University, Depart. of Electr. and Comp. Engineering
5000 Forbes Ave., Pittsburgh, PA 15213-3890, USA, **rausch@des.ece.cmu.edu**

Abstract. With the modeling formalism of the Net Condition Event Systems (NCES) it is possible to build large modular models, but without logical function blocks. In this paper the NCES is extended so that we can include logical functions. The definition of an Extended Net Condition Event System (ENCES) is given. Furthermore the resulting firing rule and algorithms for connecting ENCES modules to new ENCES modules is described.

Introduction

The possibility to model large plants in an easy way depends on the modeling formalism and the technique to build models. The discrete behavior of a component we can describe with a Petri net. But to describe the i/o behavior and to interconnect modules Net Condition Event Systems (NCES) were introduced. With NCES it is possible to model large plants easily.

The definition of NCES given in [3, 4] has same disadvantages. It is not possible to perform logical operations on the signals. Furthermore, NCES modules can only describe MOORE automata. The result in some applications, especially in models of chemical plants, is that we must build more complicated models than would be necessary if logical function blocks were available. These models are hard to understand. The example below illustrates the problem. Figure 1 shows the plant and Figure 2 shows the NCES model. The modules for the pumps have two outputs: $co_1 = 1$, if the pump is on, $co_2 = 1$, if the pump is off. The module for the level has two inputs. If input $ci_1 = 1$ the level increases, if $ci_2 = 1$ the level will decrease. The problem of modeling this plant with these modules is that the level will increase if *one* pump works and decrease if *all* pumps do not work. Unfortunately, the module for the level has only one input for decrease and one for increase. To keep the plant model it would be useful to have a logical block to combine the pump signals. Without the logical block we must change the module for the level so that it accepts an input from each pump, but this is not in the spirit of modular modeling.

To rectify this situation we want to extend the NCES in this paper (to ENCES) so we can include logical functions. We note that we can describe MEALY automata with the new model form. The requirement is that all operations used by NCES are possible by ENCES.

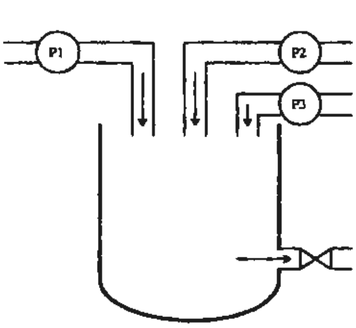


Figure 1: Plant

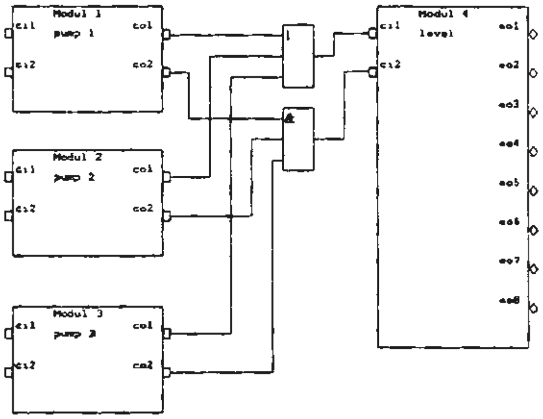


Figure 2: NCES Model with logical connections

The basic idea of NCES is that the dynamic behavior of the modules is modeled by Petri nets. The modules are connected by two types of (binary) signals which are piecewise constant (condition signals) or pointwise nonzero (event signals). Event signals are used for synchronization of modules in such a way that a state transition in a particular module forces a state transition in another module if that state transition is enabled.

Boolean blocks

We introduce two new operators. The *matrix-and-operator* $\triangle$ of matrix M_1 and vector $\underline{v}_1$ resulting in vector $\underline{v}_2$ is defined as:

$$\begin{aligned} \underline{v}_2 &= M_1 \triangle \underline{v}_1 \\ \underline{v}_2(i) &= \begin{cases} 1 & \text{if} \quad \forall M(j,i) > 0 : \underline{v}_1(j) = 1 \ \wedge \ \forall M(j,i) < 0 : \underline{v}_1(j) = 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

The *matrix-or-operator* ∇ of matrix M_1 and vector $\underline{v}_1$ resulting in vector $\underline{v}_2$ is defined as:

$$\begin{aligned} \underline{v}_2 &= M_1 \nabla \underline{v}_1 \\ \underline{v}_2(i) &= \begin{cases} 1 & \text{if} \quad \exists M(j,i) > 0 : \underline{v}_1(j) = 1 \ \vee \ \exists M(j,i) < 0 : \underline{v}_1(j) = 0 \ \vee \ \nexists M(j,i) \neq 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

We use these two operators to describe boolean blocks. In the boolean blocks the input signals can be AND and OR connected and also negated. The inputs signal have two different values, 1 and 0 (binary signals) the output signals too. The structure of a logical block is a disjunct normal form (Figure 3). This is not a restriction because all boolean functions can be transformed into this form. Furthermore we show that we can cascade the logical blocks. The connection of a logical block and an NCES is an Extended Net Condition Event System (ENCES) (see Figure 5).

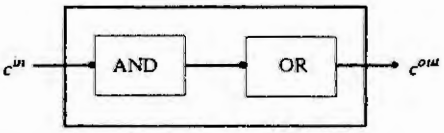


Figure 3: Principle of a boolean condition block (BCB)

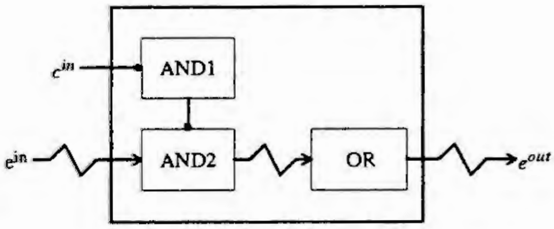


Figure 4: Principle of a boolean event block (BEB)

For the description with matrices we need two matrices; one for the AND and one for the OR connection. Every row in the AND matrix corresponds to one input signal, every column in this matrix describes one conjunction. Every row of the transposed OR matrix corresponds to one output signal.

A Boolean Condition Block (BCB) is define as

$$BCB = \{C^{in}, C^{out}, AND, OR\} \tag{3}$$

C^{in}	set of m condition input signals
C^{out}	set of n condition output signals
$AND^{m \times r}$	matrix which describe the AND connections between the input signals
$OR^{r \times n}$	matrix which describe the OR connections between the conjunctions

The input- output behavior is defined by

$$\underline{C}^{out} = OR \ \nabla \ (AND \ \triangle \ \underline{C}^{in}) \tag{4}$$

Example: We have 7 input signals ($x_1 \dots x_7$) and 3 output signals ($y_1 \dots y_3$). The following boolean equations describe the function between inputs and outputs. $y_1 = \overline{x}_1 x_2 \overline{x}_5 \vee x_2 \overline{x}_3 \vee \overline{x}_4 x_5$, $y_2 = x_1 \overline{x}_4 x_7 \vee x_2 x_5$ and $y_3 = x_1 \vee x_2 \overline{x}_6 \vee \overline{x}_3 x_4 \overline{x}_5 \overline{x}_7$. We can write these equations as two matrices in the following way. The first matrix is the AND matrix. Every row corresponds to an input ($x_1, x_2, \dots$), every column represents one conjunction. The second matrix is the transposed OR matrix. Every column is the input for one conjunction, every row describes all conjunctions (value different from zero) which are OR connected. Hence we can these two matrices describe more than one boolean function.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} = AND$$

$$OR^T = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

A Boolean Event Block (BEB) is define as

$$BEB = \{C^{in}, E^{in}, E^{out}, AND1, AND2, OR\} \tag{5}$$

C^{in} set of l condition input signals
E^{in} set of m event input signals
E^{out} set of n event output signals
$AND1^{l \times r}$ matrix which describe the AND connections between the condition input signals
$AND2^{m \times r}$ matrix which describe the AND connections between the event input signals
$OR^{r \times n}$ matrix which describe the OR connections between the condition and the event conjunctions

The input- output behavior is defined by:

$$E^{out} = OR \ \nabla \ \left(\begin{bmatrix} AND1 \\ AND2 \end{bmatrix} \triangle \begin{bmatrix} \underline{C}^{in} \\ \underline{E}^{in} \end{bmatrix} \right) \tag{6}$$

In an BEB the incoming event signals are connected. In additional they can be enabled with condition signals (see Figure 4).

Extended Net Condition Event Systems

The ENCES module is built from an NCES module, a BCB and a BEB (Figure 5).

Definition

A Extended Net Condition Event System is a tuple

$$ENCES = \{P, T, F, \underline{m}_0, Bic, Bie, Boc, Boe, AND_{BCB}, AND1_{BEB}, AND2_{BEB}, OR_{BCB}, OR_{BEB}\} \tag{7}$$

where:

P is the set of m places p
T is the set of n transitions t
$F^{m \times n}$ is the incidence matrix
$\underline{m}_0^{m \times 1}$ is the initial marking
Bic is the set of o condition inputs
Bie is the set of u event inputs
Boc is the set of q condition outputs
Boe is the set of r event outputs

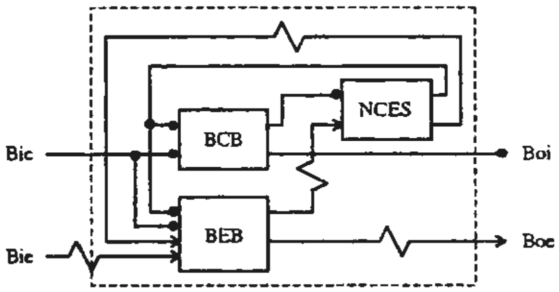


Figure 5: Structure of an ENCES module

$AND_{BCB}^{(m+o) \times s}$ conjunction matrix between condition signals
$AND1_{BEB}^{(m+o) \times v}$ conjunction matrix between condition signals to enable the event signals
$AND2_{BEB}^{(n+u) \times v}$ conjunction matrix between event signals
$OR_{BCB}^{s \times (n+q)}$ disjunction matrix between condition conjunctions
$OR_{BEB}^{v \times (n+r)}$ disjunction matrix between condition and event conjunctions

The meaning of the matrices of an NCES are changed because the AND and the OR matrices have the function for connecting into the module and for connecting with the inputs and outputs, too.

We split the AND and the OR matrices into two parts: $AND^{net} = AND(1 : m, \bullet)$ describe all connections with the net and $AND^{io} = AND((m + 1) : (m + o), \bullet)$ describe all connections with the inputs. In the same way $OR^{net} = OR(\bullet, 1 : n)$ describe the connections between the boolean block and the net and $OR^{net} = OR(\bullet, (n + 1) : (n + q))$ describe the connections between the boolean block and the outputs.

If we have no OR connections, this means in the OR matrix there is only one value per line different from zero, and we have not negations, we have an NCES and the following relations obtain: $Bc = AND^{io} \cdot OR^{net}$, $CK = AND^{net} \cdot OR^{net}$, $Cs = AND^{net} \cdot OR^{io}$

Firing rule

Since we have expanded the net description, we must define a new firing rule.

For an autonomous system: $Bic = 0$ and $Bie = 0$.

A transition t has three degrees of enabling:

1. *Marking enabled*
   A transition $t_i \in T$ is marking enabled, if $\min(M - F_m(\bullet, i)) \geq 0$.
   (F_m is the incidence matrix which only describe the pre arcs.)
2. *Condition enabled*
   A transition $t_j \in T$ is condition enabled, if $\underline{t}_C(j) = 1$ with $\underline{t}_C = OR_C \nabla (AND_C \Delta \underline{m})$
3. *Event enabled*
   A transition $t_k \in T$ is event enabled, if $\underline{t}_E(k) = 1$ with $\underline{t}_E = OR_E \nabla \left(\begin{bmatrix} AND1_E \\ AND2_E \end{bmatrix} \Delta \begin{bmatrix} \underline{m} \\ \underline{t}_e \end{bmatrix} \right)$.
   For every transition $t_e \in T_e$ with $T_e = \{t_n | \underline{t}_e(n) = 1\}$ must obtain that t_e is marking, condition and event enabled.

Connection of ENCES modules

For connection of ENCES's we need the following in [1] defined operators.

Definition: The *diagonal operator* ⊠ of matrices M_1 and M_2 resulting in M_{12} is defined as:

$$M_{12} = M_1 \boxtimes M_2 = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} \tag{8}$$

Definition: The *sequential union operator* ⨄ for ordered sets $S^1 = \{s_1^1, \ldots, s_n^1\}$ and $S^2 = \{s_1^2, \ldots, s_m^2\}$ resulting in S^{12} is defined as:

$$S^{12} = S^1 \biguplus S^2 = \{s_1^1, \ldots, s_n^1, s_1^2, \ldots, s_m^2\} = \{s_1^{12}, \ldots, s_n^{12}, s_{n+1}^{12}, \ldots, s_{n+m}^{12}\} \tag{9}$$

We apply the same operator to concatenated vectors.

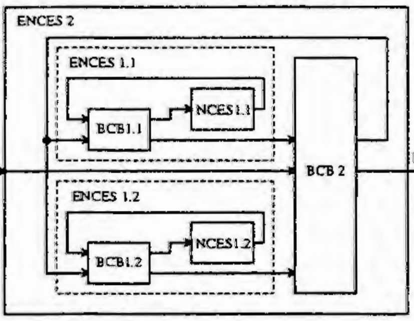


Figure 6: Connection of two ENCES to a new ENCES

$$\begin{array}{lllll} 1 & = & Bic_1^{io} & = & Boc_2^{net} \\ 2 & = & Boc_1^{net} & & \\ 3 & = & Bic_1^{net} & & \\ 4 & = & Boc_1^{io} & = & Bic_2^{net} \end{array}$$

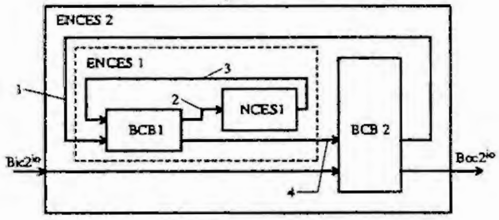


Figure 7: ENCES with two BCBs

The following describes operations with BCB, all things obtain similar for BEB. If we build an ENCES from other ENCES's we have the structure shown in Figure 6. The output signals from BCB_2 go to $BCB_{1.1}$ and $BCB_{1.2}$. From this we can build larger modules.

One BCB and an NCES is an ENCES. From two ENCES's described by two BCB's and two NCES's we can build a new NCES and a new BCB in the following way:

$$\begin{array}{ccc}
P_1 = P_{1.1} \uplus P_{1.2} & AND_1^{net} = AND_{1.1}^{net} \boxtimes AND_{1.2}^{net} & OR_1^{net} = OR_{1.1}^{net} \boxtimes OR_{1.2}^{net} \\
T_1 = T_{1.1} \uplus T_{1.2} & AND_1^{io} = AND_{1.1}^{io} \boxtimes AND_{1.2}^{io} & OR_1^{io} = OR_{1.1}^{io} \boxtimes OR_{1.2}^{io} \\
Bic_1^{net} = Bic_{1.1}^{net} \uplus Bic_{1.2}^{net} & Boc_1^{net} = Boc_{1.1}^{net} \uplus Boc_{1.2}^{net} & Bic_1^{io} = Bic_{1.1}^{io} \uplus Bic_{1.2}^{io} \\
Boc_1^{io} = Boc_{io}^{net} \uplus Boc_{1.2}^{io} & m_0^1 = m_0^{1.1} \uplus m_0^{1.2} &
\end{array}$$

Then we have the structure shown in Figure 7. We must transform the two BCB's to one BCB. The matrix representation is the following:

$$\begin{array}{ll}
\text{BCB 1:} & Boc_1^{net} = OR_1^{net} \nabla (AND_1 \triangle (Bic_1^{net} \uplus Bic_1^{io})) \\
 & Boc_1^{io} = OR_1^{io} \nabla (AND_1 \triangle (Bic_1^{net} \uplus Bic_1^{io})) \\
\text{BCB 2:} & Boc_2^{net} = OR_2^{net} \nabla (AND_2 \triangle (Bic_2^{net} \uplus Bic_1^{io})) \\
 & Boc_2^{io} = OR_2^{io} \nabla (AND_2 \triangle (Bic_2^{net} \uplus Bic_1^{io}))
\end{array}$$

It obtains: $Bic_1^{io} = Boc_2^{net}$ and $Boc_1^{io} = Bic_2^{net}$

The first step to build one boolean block is to connect the input signals with the AND_1 matrix. We compute:

$$\begin{array}{ll}
\text{BCB 1:} & Boc_1^{*io} = (OR_1^{io} \boxtimes \mathbf{I}^{|Bic_2^{io}| \times |Bic_2^{io}|}) \nabla ((AND_1 \boxtimes \mathbf{I}^{|Bic_2^{io}| \times |Bic_2^{io}|}) \triangle (Bic_1^{net} \uplus Bic_1^{io} \uplus Bic_2^{io})) \\
\text{BCB 2:} & Boc_2^{net} = OR_2^{net} \nabla (AND_2 \triangle Bic_1^{*io}) \\
 & Boc_2^{io} = OR_2^{io} \nabla (AND_2 \triangle Bic_1^{*io})
\end{array}$$

The second step is to compute a $CODE$ matrix with algorithm 1 (part 1). The new AND_1^* matrix is: $AND_1^* = (AND_1 \boxtimes \mathbf{I}^{|Bic_2^{io}| \times |Bic_2^{io}|}) \cdot CODE$. The multiplication with the $CODE$ matrix realizes all combinations of conjunctions which are necessary to build the new AND_1^* matrix. Then we can simplify the AND_1^* matrix through comparison of the columns. It is equivalent to simplifying a boolean equation.

Algorithm 1: (matlab program)

```
OR2sum = sum(OR2);      % sum of every column
CODE = []; OR_h = [];  % initial value
for i=1:size(OR2sum,2)
   if OR2sum(1,i) > 0
      help1 = (xi_function(OR1,(AND2(:,i))')')';
      help2 = OR2(:,i) * ones(size(help1,2));
      CODE = [CODE help1];
      OR_h = [OR_h help2];
   end;             % if
end;                % for
```

```
AND_star=[]; OR_star=[]; ANDstar=AND1*CODE;
sum_and1=sum((AND1~=0) * CODE ~=0);
sum_and2=sum((AND1 * CODE) ~= 0);
for i=1:size(sum_and1,2)
      x=sum_and1(1,i):
   if (x==sum_and2(1,i)) & (x~=0)
      AND_star=[AND_star ANDstar(:,i)];
      OR_star =[OR_star OR_h(i,:)];
   end;  % if
end;     % for
```

The second part of the algorithm realizes the simplification of the matrices for this cases if one signal is AND connected with its negation signal ($x \wedge \overline{x} = 0$). Then the conjunction is zero and we do not need this conjunction in the following.

The Ξ operator (`xi_function`) realizes a permutation of columns in a matrix.

Example: In the first matrix should permute column 1 and row 3.

$$OR_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad V = [\ 1 \quad 0 \quad 1\ ] \qquad \Xi(OR_1, V) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Now we have the structure in Figure 8. The one difference from an ENCES is the feedback connection.

$$Boc^{net} \uplus Boc_1^{*io} \uplus Boc_2^{io} \quad = \quad OR_1^*\nabla(AND_1^*\Delta(Bic_1^{net} \uplus Bic_1^{io} \uplus Bic_2^{io}))$$

The transformation of a feedback connection into the structure shown in Figure 9 is equivalent to substituting a boolean variable with a boolean term. If the variables which should be substituted are negated we must use the rule from DE MORGAN. All operations which are possible with logical functions are also possible with matrices.

The result of the transformation is the structure in Figure 9 and the following function:

$$Boc^{net} \uplus Boc^{io} = OR^{new}\nabla(AND^{new}\Delta(Bic^{net} \uplus Bic^{io}))$$

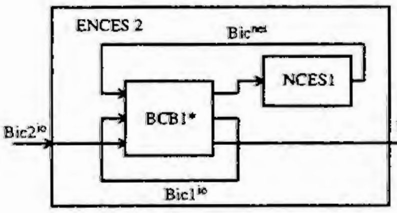


Figure 8: ENCES with feedback connection

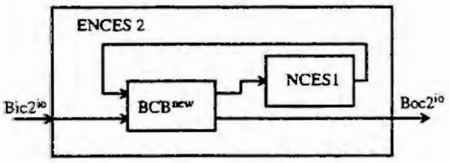


Figure 9: Pure ENCES

Conclusion and further work

With Extended Net Condition Event Systems we can build all models which we described with Net Condition Event Systems. In addition, in ENCES we have logical blocks so we can connect signals with boolean operations. Hence the model formalism of the ENCES is more powerful as the NCES. For example with ENCES we can describe a MEALY automata; with NCES this is not possible. Furthermore to model plants the ENCES as powerful as the Boolean Condition Event Systems defined by KROGH and KOWALEWSKI [2]. But the ENCES has same advantages: the connecting operations do not compute the whole reachability space. So it is possible to build larger models.

In the future we must expand the developed algorithm to synthesized a controller [5] for the new model form. In addition, all tools must be expanded. Furthermore we will check whether colored tokens are useful for modeling and whether we can develop algorithms to synthesize controllers with such models.

References

[1] Lüder, A., Anderssohn, U. and Hanisch, H.-M., Logic Controller Synthesis for Net Condition/Event Systems with Forbidden Paths. Submitted to the European Control Conference '97, Brussels, Belgium, 1.-4. July 1997.

[2] Krogh, B.H. and Kowalewski, S., Boolean Condition/Event Systems: Computational Representation and Algorithms. In: Preprints IFAC 12th World Congress, Sydney, Australia, July 1993, 327–330.

[3] Rausch, M., Modulare Modellbildung, Synthese und Codegenerierung ereignisdiskreter Steuerungssysteme. phd thesis, Otto-von-Guericke-Universität Magdeburg, Department of Electrical Engineering, 1996.

[4] Rausch, M. and Hanisch, H.-M., Net Condition/Event Systems with Multiple Condition Outputs. In: Symposium on Emerging Technologies and Factory Automation, Paris, France, 10.–13. Oct., Band 1, INRIA/IEEE, 1995, 592–600.

[5] Rausch, M., Lüder, A. and Hanisch, H.-M., Combined Synthesis of Locking and Sequential Controllers. In: Internat. Workshop on Discrete Event Systems (WODES'96), Edinburgh, UK, 19.–22. Aug. 1996, 133–138.

INVARIANCE OF PREDICATES AND STATE FEEDBACK LOGIC SYNTHESIS OF CONTROLLED TIME PETRI NETS

Chen Haoxun
Systems Engineering Institute
Xi'an Jiaotong University, Xi'an 710049, P.R. China

Abstract. Synthesis of state feedback logic for the control problem of maintaining a predicate on the state set of a timed discrete event system is considered in the setting of controlled time Petri nets. A kind of invariance for predicates is introduced and a fixpoint algorithm is proposed for computing the extremal invariant predicate. On the basis of this, maximally permissive state feedback logic can be systematically synthesized.

1. Introduction

Given a discrete event system (DES), the control problem is to synthesize a controller which realizes the closed-loop behaviours of the DES within a prespecified set of desirable behaviours. A well-developed theoretical framework for control synthesis of DESs at the logical level has been established by Ramadge and Wonham [4] [5], which has been extended to controlled Petri nets (CtlPNs) recently [3].

When time constraints are explicitly concerned in the dynamics and performance specifications of DESs, their control becomes much more complicated, but of considerable applied interest, particularly in the design of hard real-time systems [1]. Brandin and Wonham [1] extended the automaton model of untimed DESs by adding a clock-tick event and augmenting the state space with a timer for each activity to describe timed DESs. The resulting framework retains most of the concepts introduced in [4]. However, the addition of a global clock greatly increases the number of transitions in the system. In Cofer and Garg [2], the behaviour of a class of timed DESs, timed event graphs, is described by sequences of event occurrence times. By using a max-algebra model for timed event graphs, they demonstrate that the control problem can be viewed from the Ramadge-Wonham perspective. However, the max-algebra based approach is only applicable to decision-free timed DESs.

In this paper, we consider control of timed DESs in the setting of CTPNs. The control problem is to ensure by appropriate state feedback control that a given predicate on the state set of a CTPN remains invariantly true if it is initially satisfied. We first focus on controlled time safe Petri nets (CTSPNs), a class of CTPNs in which the number of tokens in each state place does not exceed one for any marking reachable from the initial marking. A kind of invariance for predicates is introduced for CTSPNs, and a fixpoint algorithm for computing the extremal invariant predicate is proposed. On the basis of this, maximally permissive state feedback logic can be systematically synthesized. The results are then generalized to CTPNs with unsafe "resource-type" places. The theoretical results obtained in this paper are illustrated with a real-time control problem of batch chemical process.

2. Controlled Time Safe Petri Nets

A time safe Petri net (TSPN) is represented by $N_t = (P,T,F,D,M_0,C_0)$, where P is the set of *places*, T is the set of *transitions*, $F \subseteq (P \times T) \cup (T \times P)$ is the set of directed arcs connecting places with transitions. D: $P \rightarrow \mathbf{Z} \times \mathbf{Z}$ assigns *a time interval* $D(p) = [d_1(p), d_2(p)]$ to each place $p \in P$ where $\mathbf{Z}$ is the set of nonnegative integers and $0 \leq d_1(p) \leq d_2(p) < +\infty$. M_0: $P \rightarrow \{0,1\}$ is *the initial marking*, C_0: $P \rightarrow \mathbf{Z}$ is *the initial clock time vector* with $0 \leq C_0(p) \leq d_2(p)$, $C_0(p)=0$ if $M_0(p)=0$. Let $N = (P,T,F,M_0)$ be the untimed Petri net *derived from* N_t, $\mathrm{R}(N,M_0)$ be the set of markings reachable from the initial marking M_0. For TSPN N_t, it is assumed that $R(N,M_0) \subseteq \{0,1\}^P$, i.e., the number of tokens in each place *does not exceed one* for any marking reachable from the initial marking.

For each place $p \in P$, $d_1(p)$ represents the time that a token must remain in the place at least. That is, when a token is put in a place p, it remains *unavailable* for a time $d_1(p)$. After a time $d_1(p)$ has passed the token in the place becomes *available* and remains in the place until the token leaves the place. It is assumed that in each place p there is a local digital clock measuring the time that a token has remained in the place, with $d_2(p)$ representing the maximum tick count of the local clock. That is, after the tick count of the local clock in a place p reaches its maximum value $d_2(p)$, the count will keep invariantly as if the clock was frozen, until a token leaves the place. Note that since each place has at most one token, the local clocks are attached to places instead of being attached to the tokens in the places.

The *state* of a TSPN is given by its current *marking* M: $P \rightarrow \{0,1\}$and its current *clock time vector* C: $P \rightarrow \mathbf{Z}$, with $0 \leq C(p) \leq d_2(p)$, $C(p)=0$ if $M(p)=0$. Let $\mathcal{C}(M)$ be the set of valid clock time vectors for M, i.e., $\mathcal{C}(M) = \{ C\colon P \rightarrow \mathbf{Z} \mid 0 \leq C(p) \leq d_2(p),\ C(p)=0 \text{ if } M(p)=0\}$. The state set of the TSPN is given by $\mathcal{X} = \{ X = (M,C) \mid M \in R(N,M_0),\ C \in \mathcal{C}(M)\}$.

A transition $t \in T$ is said to be *enabled* at a state $X = (M,C)$ if for each $p \in P$ with $(p, t) \in F$, $M(p) = 1$ and $C(p) \geq d_1(p)$. This notion can be generalized to a set of transitions.

Definition 1: A set of transitions $T' \subseteq T$ in a TSPN is said to be *enabled* at a state $X = (M,C)$ if all transitions $t \in T'$ are enabled and ${}^\circ t_1 \cap {}^\circ t_2 = \varnothing$ for any $t_1, t_2 \in T'$, where ${}^\circ t$ denotes the set of input places of transition t, i.e., ${}^\circ t = \{ p \in P \mid (p, t) \in F\}$.

The family of all enabled sets of transitions at a state X is denoted by $\mathcal{T}_e(X)$, i.e., $\mathcal{T}_e(X) = \{ T' \subseteq T \mid T'$ is enabled at a state X }

In order to precisely specify the behaviour of a TSPN, we introduce an additional event, written *tick*, to represent "tick of a global clock". The global clock likes a clock indicating Greenwich standard time. All local clocks attached to state places of the TSPN are driven by the global clock. In addition, we introduce *forcible transitions* (*events*), elements of a subet $T_{\text{for}} \subseteq T$. A forcible transition is an event that can *preempt* a tick of the global clock. At a given state of a TSPN, *tick* is defined only when no forcible transition is enabled at the state. State transitions (changes) occur only when an enabled set of transitions in the TSPN fires, or when a *tick* event occurs.

Let $X = (M, C)$, $X' = (M', C')$. With the above assumptions, the state transition function δ : $\{2^T \cup \{tick\}\} \times \mathcal{X} \rightarrow \mathcal{X}$. of N_t can be defined as follows:

$\delta(T', X) = X'$, if $T' \subseteq T$, $T' \in \mathcal{T}_e(X)$, where

$$M'(p) = M(p) - |p^\circ \cap T'| + |{}^\circ p \cap T'| \tag{1}$$

$$C'(p) = \begin{cases} C(p), \text{ if } p^\circ \cap T' = {}^\circ p \cap T' = \varnothing \\ 0 \quad , \text{ otherwise} \end{cases} \tag{2}$$

$\delta(T', X)$ is undefined, if $T' \subseteq T$, $T' \notin \mathcal{T}_e(X)$.

$\delta(tick, X) = X'$, if $\mathcal{T}_e(X) \cap T_{\text{for}} = \varnothing$, where

$$M'(p) = M(p) \tag{3}$$

$$C'(p) = \begin{cases} C(p)+1, & \text{if } M(p)=1, C(p) < d_2(p) \\ C(p), & \text{if } M(p)=1, C(p) = d_2(p) \\ 0 \quad , & \text{otherwise} \end{cases} \tag{4}$$

$\delta(tick, X)$ is undefined, if $\mathcal{T}_e(X) \cap T_{\text{for}} \neq \varnothing$.

To exculde the physically unrealistic possibility that a *tick* event might be preempted indefinitely by

To exculde the physically unrealistic possibility that a *tick* event might be preempted indefinitely by repeated firing of a set of transitions within a fixed unit time interval, we impose a technical condition that any transition in a TSPN can not occur more than one time within one unit time interval.

In what follows, we will use another function $\bar{\delta}$, which is the same as δ except for that $\bar{\delta}$ $(tick,X)$ is defined for all $X \in \mathcal{X}$ with its value given by (3) and (4).

A controlled time safe Petri net (CTSPN) is a time safe Petri net augmented with control places to represent external control inputs. Formally, a CTSPN can be represented by $N_{ct} = (N_t, P_c, B_c)$, where $N_t = (P,T,F,D,M_0,C_0)$ is a TSPN, P_c is the finite set of *control places*, $B_c \subseteq (P_c \times T)$ is the set of directed arcs connecting control places to transitions. Places belonging to P are called *state places*. The *state* of the CTSPN N_{ct} is given by the state of its underlying TSPN N_t.

The set of *controllable transitions* $T_c \subseteq T$ is defined as the set of transitions $t \in T$ with an associated control place, i.e., $T_c = \{\ t \in T \mid \exists p \in P_c,\ (p,t) \in B_c\}$. Transitions not in T_c, i.e. transitions in $T_u = T \backslash T_c$, are referred to *uncontrollable* transitions.

An uncontrollable transition that is enabled can never be disabled (erased) by control action. However, an uncontrollable transition may be a forcible transition in the sense that it occurs immediately when it is enabled. By contrast, the occurrence time of an uncontrollable transition that is not forcible is undeterministic. In this paper, we assume that all controllable transitions are forcible, i.e., $T_{for} \supseteq T_c$.

A *control* u: $P_c \rightarrow \{0,1\}$ assigns a binary token count to each control place. The set of all possible controls is denoted by $\mathcal{U}$.

A set of transitions $T' \subseteq T$ in N_{ct} is said to be *state-enabled* at a state X if it is enabled in N_t at the same state. A set of transitions $T' \subseteq T$ in N_{ct} is said to be *control-enabled* under a control u if $u(p)=1$ for all $p \in P_c$ with $t \in T'$ satisfying $(p,\ t) \in B_c$, with the convention that any uncontrollable transition is always control-enabled. A set of transitions $T' \subsetneq T$ in N_{ct} is said to be *enabled* for a state $X = (M,C)$ and control u if it is both state-enabled and control-enabled.

The family of all enabled sets of transitions for a state X and control u is denoted by $\mathcal{T}_{ce}(X,u)$, i.e., $\mathcal{T}_{ce}(X,u) =$ $\{\ T' \subseteq T \mid\ T'$ is enabled for a state X and control u $\}$

Similar to N_t, state transitions (changes) of N_{ct} occur only when an enabled set of transitions in the CTSPN fires, or when a *tick* event occurs. Also, at a given state of N_{ct}, *tick* is defined only when no forcible transition is enabled at the state.

The state transition function δ_c: $\{2^T \cup \{tick\}\} \times \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$. of N_{ct} can be defined as follows:

$\delta_c(T',X,u) = \delta(T',X)$, if $T' \subseteq T$, $T' \in \mathcal{T}_{ce}(X,u)$.

$\delta_c(T',X,u)$ is undefined, if $T' \subseteq T$, $T' \notin \mathcal{T}_{ce}(X,u)$.

$\delta_c(tick,X,u) = \bar{\delta}(tick,X)$, if $\mathcal{T}_{ce}(X,u) \cap T_{for} = \varnothing$.

$\delta_c(tick,X,u)$ is undefined, if $\mathcal{T}_{ce}(X,u) \cap T_{for} \neq \varnothing$.

3. State Feedback

A state feedback f for the CTSPN N_{ct} is defined as a total function f: $\mathcal{X} \rightarrow \mathcal{U}$. The application of f to N_{ct} yields the closed loop system N_{ct}^f defined as follows:

In N_{ct}^f, a set of transitions $T' \subseteq T$ is enabled for a state X iff T' is enabled for state X and control $f(X)$. A *tick* event can occur in N_{ct}^f only when no forcible transition is enabled in the closed loop system. State transition function δ^f of N_{ct}^f is similar to δ_c except for that $f(X)$ substitutes for u.

A state feedback f_1 is said to be *more permissive* than a state feedback f_2 (denoted by $f_1 > f_2$) if for all state $X \in \mathcal{X}$, $f_1(X) \geq f_2(X)$ and $f_1(X') > f_2(X')$ for some $X' \in \mathcal{X}$.

The set of all state feedback laws will be denoted by $\mathcal{F}$.

4. Invariance of Predicates

A predicate on the state set $\mathcal{X}$ of a CTSPN is a function Q: $\mathcal{X} \rightarrow \{0,1\}$, i.e., a characteristic function on $\mathcal{X}$. Actually, each predicate Q corresponds to a subset of $\mathcal{X}$, i.e, $Q \leftrightarrow \mathcal{X}_Q = \{ X \in \mathcal{X} \mid Q(X) = 1 \}$. Let $\mathcal{Q}$ denote the set of all predicates on $\mathcal{X}$.

Let $T_e = T \cup \{tick\}$ and define the predicate

$$D_t(X) = \begin{cases} 1, \text{ if } \delta(t, X) \text{ is defined} \\ 0, \text{ otherwise} \end{cases}, \quad t \in T_e$$

and the predicate transformation wp_t: $\mathcal{Q} \rightarrow \mathcal{Q}$ by

$$\mathrm{wp}_t(Q)(X) = \begin{cases} 1, \text{ if } \delta(t, X) \text{ is defined and } Q(\delta(t, X)) = 1 \\ 0, \text{ otherwise} \end{cases}, \quad t \in T$$

$$\mathrm{wp}_{tick}(Q)(X) = \begin{cases} 1, \text{ if } \bar{\delta}(tick, X) \text{ is defined and } Q(\bar{\delta}(tick, X)) = 1 \\ 0, \text{ otherwise} \end{cases}$$

$\mathrm{wp}_t(Q)$ is called the *weakest precondition* of Q under t [5]. The *weakest liberal precondition* of Q is defined by $\mathrm{wlp}_t(Q) = \mathrm{wp}_t(Q) \vee \sim D_t$ [5].

Definition1: A predicate $Q \in \mathcal{Q}$ is said to be $(T_u, tick, T_{for})$-invariant (w.r.t. N_{ct}) if

$$Q \leq \{ \bigwedge_{t \in T_u} \mathrm{wlp}_t(Q) \} \wedge \{ \bigvee_{t \in T_{ef}} \mathrm{wp}_t(Q) \}, \tag{5}$$

where $T_{ef} = T_{for} \cup \{tick\}$.

We have the following theorem:

Theorem 1: A predicate $Q \in \mathcal{Q}$ is $(T_u, tick, T_{for})$-invariant if and only if there is a state feedback f such that for N_{ct}^f, Q remains invariantly true whenever it is initially satisfied, where f: $f(X) = u$ for some $u \in \mathcal{U}(X)$, $\mathcal{U}(X) = \{ u \in \mathcal{U} \mid Q(\delta_c(T^*, X, u)) = 1$ for any $T^* \in \mathcal{F}_{ce}(X, u)$, and $\delta_c(tick, X, u)$ is undefined or $Q(\delta_c(tick, X, u)) = 1\}$.

$(T_u, tick, T_{for})$-invariant is also briefly called *tick*-invariant hereafter.

Theorem 1 implies that if Q is a *tick*-invariant predicate, then a state feedback to ensure the invariance of Q can be synthesized by selecting control $f(X)$ from set $\mathcal{U}(X)$.

Let $\mathcal{F}(Q) = \{ f \in \mathcal{F} \mid f$: $f(X) = u$ for some $u \in \mathcal{U}(X)\}$

It can be shown that a state feedback f ensures the invariance of Q in N_{ct}^f iff $f \in \mathcal{F}(Q)$ and a maximal element in $\mathcal{F}(Q)$ can be thought as one of the most permissive (the least restrictive) state feedback among all state feedback laws that ensure the invariance of Q. However, due to possible concurrence of several transitions in CTSPNs, such a maximal element may be not unique as that has been proved for CtlPNs.

5. Extremal Invariant Predicate

For $Q \in \mathcal{Q}$, we define

$$TI_{<}(Q) = \{ Q' : Q' \in \mathcal{Q}, Q' \leq Q \text{ and } Q' \text{ is } tick\text{-invariant}\}$$

Proposition 1: $TI_{<}(Q)$ is nonempty and is closed under arbitrary disjunctions (unions).

At this point, we can assert the existence of a unique maximal element of $TI_{<}(Q)$, which will be denoted by $Q^{\uparrow}$ as in [5]. $Q^{\uparrow}$ can be thought as the best *tick*-invariant approximation to Q among the predicates which are stronger than Q.

As in [5], we can develop a fixpoint algorithm for the computation of $Q^{\uparrow}$. In order to do so, we introduce the map: $H: \mathcal{Q} \rightarrow \mathcal{Q}$ with

$$H(Q') = Q \wedge [(\bigwedge_{t \in T_u} \mathrm{wlp}_t(Q')) \wedge (\bigvee_{t \in T_d} \mathrm{wp}_t(Q'))] \qquad (6)$$

Proposition 2: $Q^{\uparrow}$ is the unique maximal fixpoint of the map H.

Define the sequence of predicates:

$$Q_0 = Q, \quad Q_{j+1} = H(Q_j), j=0,1,2,...... \qquad (7)$$

The sequence has the following properties:

Proposition 3: The sequence defined by (7) is monotone decreasing and $Q^{\uparrow} = \lim_{j \rightarrow +\infty} Q_j$.

From the proposition, we can compute $Q^{\uparrow}$ by the iterative process given by (7) and if Q is a finite set the iterative process will terminate in finite steps.

6. Generalization

The results in previous sections can be generalized to controlled time Petri nets with unsafe "resource-type" state places. Here, a state place p is said to be a resource-type place if $D(p) = (0, 0)$ ($d_1(p)=d_2(p)=0$) and "unsafe" means that the number of tokens in the place may exceed one. This is because for a place of this type lifetimes of all tokens in it are the same. We can use one local clock to measure all lifetimes of the tokens. In fact, in this case, it's no matter how long a token has remained in the place.

Since the Petri net models for most DESs of practical interest have two types of places - "process" places and "resource" places - with the process places being safe (a process place having one token indicates that the underlying system has executed to the step the place stands for), with this generalization, the framework for the synthesis of state feedback logic for timed DESs proposed in this paper can deal with real-time control problems of these DESs.

7. Example

A plant for the treatment of two chemical products is shown in Fig. 1. (a). Product 1 is first processed in tank 1 (duration: 3 time units) and then processed in tank 3 (duration: 2 time units). Product 2 is first processed in tank 2 (duration: 4 time units) and then processed in tank 3 (duration: 2 time units). The tank 3 is exclusively used by product 1 and product 2, and at any time each tank can treat at most one unit of each product. Due to the nature of chemical treatment, product 1 (resp. product 2) can not remain in tank 1 (resp. tank 2) for more than 1 time unit after it finishes its treatment in tank 1 (resp. tank 2), otherwise the product is unusable. Our objective is to ensure that no unusable product comes out.

The plant is modeled by a CTSPN shown in Fig. 1. (b). For simplicity, all control places of the CTSPN and their associated arcs are omitted.

Places $\{p_{i1}, p_{i2}\}$identify the different stage treatments of product i, i=1,2, and r_j represents whether tank j is available for treatment (is empty), j=1,2,3. Transitions $\{t_{i1}, t_{i2}\}$represents the initiation of the different stage treatments of product i, and t_{i3} denotes the completion of the final stage treatment of product i, i=1,2. $T_c = \{t_{11}, t_{12}, t_{21}, t_{22}\}$, $T_u = \{t_{13}, t_{23}\}$ and we take $T_{for} = T_c \cup T_u$. That is, both of the uncontrollable transitions t_{13} and t_{23} are forcible. Namely, the completion of the final stage treatment of a product occurs immediately when it's time for the treatment to complete. We take $D(p_{11}) = (3, 5)$, $D(p_{21}) = (4, 6)$, $D(p_{12}) = D(p_{22}) = (2, 2)$, $D(r_1) = D(r_2) = D(r_3) = (0, 0)$ (explained below). In terms of predicates, our objective is to ensure by appropriate control action that the predicate Q = $\{(M,C)\in\mathcal{I}$: $C(p_{11})\leq 4$, $C(p_{21})\leq 5\}$remains invariantly true whenever it is initially satisfied. Note that when the specifications of a CTSPN specify an upper bound for the time that a token can stay in a state place, the maximum tick count for the place can be taken as the upper bound plus 1 without essential loss in describing either of the uncontrolled behaviours and controlled behaviours of the CTSPN. For this reason, we take $d_2(p_{11})$=4+1=5, $d_2(p_{21})$=5+1=6.

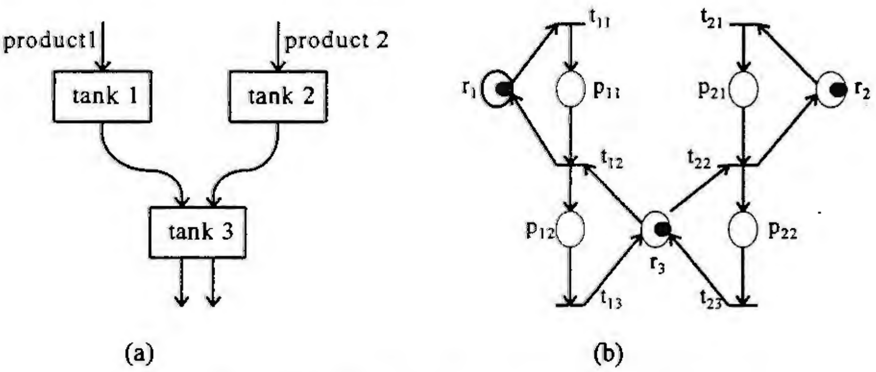


Fig. 2. Flowsheet of a chemical plant and its CTSPN model

Maximally permissive state feedback logic can be synthesized by using the method proposed in previous sections of the paper. For paper length limitation, the details for this are omitted.

8. Conclusion

The paper provides a theoretical framework for the synthesis of maximally permissive state feedback logic for timed discrete event systems modeled by CTPNs. The model introduces the maximum tick count for each place to reduce the state space size of the underlying system and can deal with both forcing control and disablement control. In particular, the fixpoint algorithm for computing the extremal invariant predicate makes possible to synthesize maximally permissive state feedback logic as efficiently as its untimed precursor.

Acknowledgment. The work of this paper is supported by NSFC under grant No. 69504008 and the author thanks Prof. Hans-Michael Hanish of Magdeburg University Germany for his organization of the special session at the symposium.

References

1. B. A. Brandin and W.M. Wonham, Supervisory control of timed discrete-event systems, *IEEE Trans. on Automatic Control*, Vol.39, No.2, pp.329-341, 1994.
2. D. D. Cofer and V. K. Garg, Supervisory control of real-time discrete-event systems using lattice theory, *IEEE Trans. on Automatic Control*, Vol.41, No.2, pp.199-209, 1996.
3. L.E. Holloway and B.H. Krogh, Synthesis of feedback control logic for a class of controlled Petri nets, *IEEE Trans. on Automatic Control*, Vol.35, No.5, pp.514-523, 1990.
4. P.J. Ramadge and W.M., Wonham, Supervisory control of a class of discrete event processes, *SIAM J. Control and Optimization*. Vol.25, No.1, pp.206-230, 1987.
5. P.J. Ramadge and W.M., Wonham, Modular feedback logic for discrete event systems, *SIAM J. Control and Optimization*. Vol.25, No.5, pp.1202-1218, 1987.

A DISCRETE MODELLING PROCEDURE FOR CONTINUOUS PROCESSES BASED ON STATE-DISCRETISATION

Heinz A. Preisig, Marc J.H. Pijpers & Martin Weiss
Eindhoven University of Technology
5600 MB Eindhoven, The Netherlands
e-mail: H.Preisig@ctrl.phys.TUE.NL

Abstract A mathematically rigorous method for computing the complete non-deterministic automaton for a hybrid plant is presented. Knowledge of the plant (linear, time-constant), discretisation of the state space and discrete-event inputs to the plant, are assumed.

Introduction

Discrete-event controllers are widely used in industrial practice in the form of programmable logical controllers. They are used in a wide range of applications dominantly in discrete manufacturing but also in processing plants where they supervise start-up, shut-downs and other sequential control operations [1,6]. Whilst the applications are numerous, the design of this type of controllers is far less understood than is the design of continuous or digital controllers. One of the impediments in the pursuit of better design methods is the lack of reliable and complete methods for the modelling of hybrid systems. This paper addresses this problem.

Problem Definition

The stage for the development is set in Figure 1 showing the continuous plant in the centre affected by its environment. The relevant parts of the environment and the plant itself is observed by a event detector sometimes also called quantizer [2] or state domain observer [4,5]. It generates a signal to the supervisor, the discrete-event controller, whenever an event occurs. The problem is posed to generate a complete discrete-event dynamic model for the discrete-event dynamic plant indicated by the dotted box. With the environment ultimately extending to the rest of the universe, the obvious decision of including only the relevant parts of the environment must be taken.

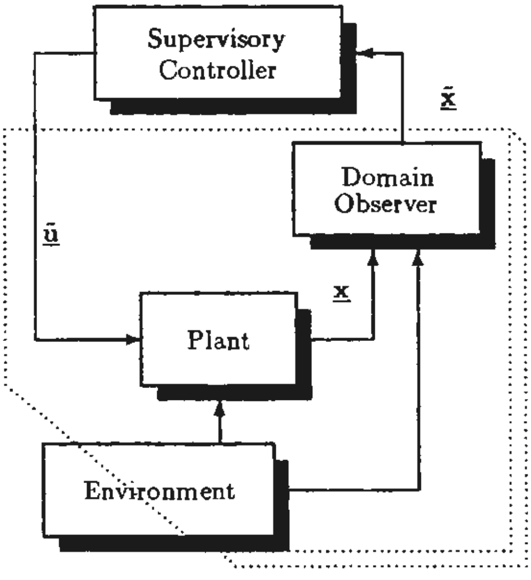


Figure 1 : Definition of the DED plant.

An event we define here as a transition of a continuous state variable across a boundary splitting the continuous state domain into two sub-domains [4,5]. These boundaries may be defined by inherent limitations of the model or plant, operational requirements or other limitations of the system. Typical examples are alarm and warning limits, or boundaries of operational domains. A set of boundaries is defined for each continuous state dividing the continuous domain into a set of discrete sub-domains. The sub-domains represent discrete states the event-discrete system may assume.

For generality it is assumed that the continuous state is defined on the real axis. The validity range thus spreads over the range $(-\infty, \infty)$. The mapping of the discrete state variables must be defined. Different definitions are possible. A definition that is based on the ordered set of domains proved useful. Let the ordered set of boundaries be $\mathcal{S}_{x_i}$ of n_i boundaries for a continuous state variable x_i,

$$\mathcal{S}_{x_i} := \{\beta_i^1, \ldots, \beta_i^l, \beta_i^{l+1}, \ldots, \beta_i^{n_i}\} . \qquad (1)$$

The discrete state can be symbolised by the ordinal number of boundaries:

$$\tilde{x}_i(k) := l, \qquad \text{for} \qquad x_i(t) \in (\beta_i^l, \beta_i^{l+1}] \quad ; \quad l \in \{1, 2, \ldots, n_i - 1\} \tag{2}$$

with $\beta_i^1 = -\infty$ and $\beta_i^{n_i} = +\infty$ the extreme boundaries of the real axis.[1] The numerical values of the discrete states are defined as the ordinal number of the sub-domains. Requiring the plant state to be continuous, the change of the discrete states is limited to $\{+1, -1\}$.[2] The change in the discrete state is thus limited to unity with the sign indicating the direction. Assuming in addition that the connection between the domain observer and the event-discrete controller is strictly sequential, only one event, namely the change in one state variable, can be detected and reported to the controller. This last assumption excludes the anyhow unlikely occurrence of simultaneous events.

Development of the Base Algorithm

To explain the approach we introduce the example of a pendulum. The pendulum is enclosed in a box. Four light beams travel horizontally across the box. Two in each of the co-ordinates spanning the floor of the box. The position of the pendulum, which is suspended from the roof of the box, is given by the detectors which report interruption of the corresponding light beam by the pendulum. The model describes the movement of the pendulum as it is projected onto the floor in the two-dimensional state variables of the floor co-ordinates. The beams split the state domain into 9 sub-domains bounded by the wall of the enclosing box. The state-space representation of the pendulum as a two-dimensional oscillator is given by the equation :

$$\dot{\underline{x}} = \begin{pmatrix} -1 & 2 \\ -3 & -1 \end{pmatrix} \underline{x} \quad ; \quad \underline{x} := [x_1,\ x_2]^T \tag{3}$$

The position of the beams define the inside boundaries (-0.5,0.5) and the wall position (-1,1) the outside boundaries of the subdomains.

$$\mathcal{S}_{x_1} \equiv \mathcal{S}_{x_2} := \{-1, -0.5, 0.5, 1\} \tag{4}$$

Thus the domain of the continuous states, bounded by the wall, and the corresponding discrete state variables are:

$$\begin{array}{ccc} \text{Continuous domain} & & \text{Discrete domain} \\ x_1, x_2 \in [-1, 1] & \longrightarrow & \tilde{x}_1, \tilde{x}_2 \in \{1, 2, 3\} \end{array} \tag{5}$$

Transitions across the outer boundaries are not allowed. The discrete-event dynamic model of the pendulum is easy to construct. It takes the form of a non-deterministic automaton because the transition depends not only on the discrete state but also on the trajectory on which the process moves or, which is the same, on the initial conditions of the pendulum. To illustrate this point, observe that in Figure 2, which shows a number of sample trajectories, changes occur, for example, from discrete state (1,2) to (2,2) or (1,3). Since the observer sitting outside the box does not know the initial condition or has incomplete information about the current trajectory, the automaton will be non-deterministic.

The mathematical analysis is based on the analysis of the intersection of the trajectories with the boundaries. The result is a general algorithm for the computation of all possible transitions. The state transition concept is used directly, as it has been define above. This limits the analysis to a directionality analysis of intersections of all trajectories with the boundary of a sub-domain representing a discrete state. We first construct the automaton table empirically to demonstrate the basic idea. Taking state (1,1) as a starting point, we observe that the pendulum must move to state (1,2) next. This provides us the first entry in the list. Starting with (1,2), two transitions may take place. Dependent on which trajectory the process is following it may fall into (1,3) or into (2,2). The analysis is readily completed, resulting the non-deterministic automaton as it is shown in Table 1.

[1] The definition of the two sub-domains $(-\infty, \beta^2]$ and $[\beta^{n-1}, +\infty)$ is from the technical point of view a useful approach. In most technical applications, the validity range of the state variables is limited to a finite interval. If in an application values are measured outside this range, an exception must have occurred and should be treated correspondingly. The two additional sub-domains can thus be defined as exception states. They add to the completeness of the description.

[2] In the case, where the state is first sampled in a digital device, the sampling must be fast enough to detect any of the transitions. This gives rise to a Nyquist-type of criterion for the design of the sampling operation.

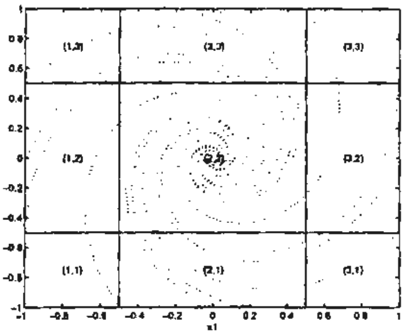


Figure 2 : The view on the bottom of the box. The solid lines show the position of the beams and the dotted lines show sample trajectories

current state	possible next state
(1,1)	(1,2)
(1,2)	(1,3),(2,2)
(1,3)	(2,3)
(2,1)	(1,1),(2,2)
(2,2)	(1,2),(2,3),(3,2),(2,1)
(2,3)	(2,2),(3,3)
(3,1)	(2,1)
(3,2)	(2,2)(3,1)
(3,3)	(3,2)

Table 1 : All possible transitions for each of the nine discrete states for the pendulum

For a linear plant, the boundaries split into two parts : One section where the sign of the first derivative of the corresponding state is positive and another where the sign is negative. The two sections are separated by a point where the derivative is zero. Figure 3 shows this for the left inner boundary of the state variable x_1. The point separating the two parts of the boundary line is defined by the intersection of the boundary line

$$\beta_1^2 - x_1 = 0 \quad ; \quad \beta_1^2 := -0.5 \tag{6}$$

with the equilibrium line for x_1 defined by :

$$\dot{x}_1|_{x_1:=\beta_1^2} = \underline{\mathbf{a}}_{1\circ} \begin{pmatrix} \beta_1^2 \\ x_2 \end{pmatrix} = 0 \tag{7}$$

where $\underline{\mathbf{a}}_{1\circ}$ the first row vector of matrix $\underline{\underline{\mathbf{A}}}$. The equilibrium line for x_1 is the sloped dashed line in Figure 3. The small circles mark the two intersection points on the corresponding boundary. The dashed line, set next to the boundary, indicate the direction of the transition, which is readily verified by the sample trajectories. The directions of the transitions are given by the signs of the first derivative of $\dot{x}_1$ for any value of x_2 greater than the intersection point on that boundary. For example in the case of boundary β_1^2:

$$\text{sign}(\dot{x}_1) = \text{sign}(a_{1,1}\beta_1^2 + a_{1,2}(x_2^{1,2} + \delta x_2)) = \text{sign}(a_{1,2}\delta x_2) = \text{sign}(a_{1,2}) \cdot \text{sign}(\delta x_2) \tag{8}$$

The coefficient of $a_{1,2}$ determines thus the direction of the trajectory above and below the equilibrium line.

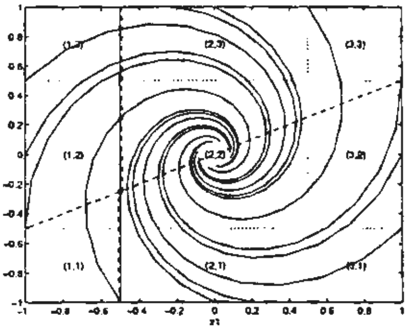


Figure 3 : Analysis of one boundary (solid line). The sloped dashed line shows the equilibrium line for x_1 as a function of x_2.

$(\tilde{x}_1, \tilde{x}_2)$	$D\tilde{x}_1$		$D\tilde{x}_2$	
(1,1)				+1
(1,2)		+1		+1
(1,3)		+1		
(2,1)	−1			+1
(2,2)	−1	+1	−1	+1
(2,3)		+1	−1	
(3,1)		+1	−1	
(3,2)	−1			
(3,3)			−1	

Table 2 : Table showing current states and all possible changes. The columns are labelled with the discrete derivative $D(\tilde{\underline{\mathbf{x}}}) := \tilde{\underline{\mathbf{x}}}(k+1) - \tilde{\underline{\mathbf{x}}}(k)$

This procedure can also be applied to all other inner boundaries. The calculated events are all possible events that can occur and can be represented in an automaton table, as shown in Table 2.

Extension to General Time-Constant Linear Systems

The method, as it has been discussed for the two-dimensional problem, can be extended to an arbitrary-dimensional problem. The extension to the n-dimensional problem is not quite trivial and is best done through the intermediate step of analysing a three dimensional problem first. The application of the idea in the three-dimensional case asks for the projection into two dimensions, for every interval of the third variable. The equilibrium line of the 2-dimensional case expands into a plane and the projection of this plane into a band. The band splits the boundary line in the projection plane into three sections. On the extremes the two sections with unique signs for the transitions and a third section cut out by the two boundary lines of the band in which transitions in both directions are possible. In the higher dimensional case, a hyper-plane, representing the equilibrium for the variable being analysed is projected onto a two-dimensional plane spanned by the co-ordinate being analysed for events versus a second variable which can be freely chosen from the others. The band now extends into a set of bands for a combination of the remaining state variables. The problem is then left to find the outermost boundary defining the band in which transitions in both directions occur. Below we calculate all possible transitions for state variable x_i for the domain of a second state variable x_j for which the element $a_{i,j}$ in the system matrix $\underline{\underline{\mathbf{A}}}$ is non-zero. For all other state variables, that is $\mathcal{X}_k := \{x_k | \forall k, k \neq i,j\}$ the sub-domain is fixed to $x_k \in (\beta_k^{d_k}, \beta_k^{d_k+1}]$. Again the intersection of the equilibrium surface with the projection plane is computed, but now for all boundaries[3] as defined. For each combination of boundaries for the set $\mathcal{X}_k$ a band is generated in which transitions in both directions are possible. The maximum width of all the bands determines then the range in x_j in which both transitions are possible. The bands are computing by solving

$$\dot{x}_i \,\big|_{\beta_i^l, x_k \in (\beta_k^{d_k}, \beta_k^{d_k+1}] \forall k, k \neq \{i,j\}} (x_j) = 0 \qquad , \qquad \underline{\tilde{\mathbf{u}}} := \text{given} \tag{9}$$

for all values of x_k as defined. The spread of the band is then given by :

$$\text{minimum}: \quad \underline{x}_j^{i,l} \quad := \quad \frac{-1}{a_{i,j}} \left(\sum_{\substack{k \\ a_{i,k} > 0 \\ k \neq i,j}} a_{i,k}\, \beta_k^{d_k+1} + \sum_{\substack{k \\ a_{i,k} < 0 \\ k \neq i,j}} a_{i,k}\, \beta_k^{d_k} + a_{i,i}\, \beta_i^l + \underline{\mathbf{b}}_{i\circ}\, \underline{\tilde{\mathbf{u}}} \right) \tag{10}$$

$$\text{maximum}: \quad \overline{x}_j^{i,l} \quad := \quad \frac{-1}{a_{i,j}} \left(\sum_{\substack{k \\ a_{i,k} < 0 \\ k \neq i,j}} a_{i,k}\, \beta_k^{d_k+1} + \sum_{\substack{k \\ a_{i,k} > 0 \\ k \neq i,j}} a_{i,k}\, \beta_k^{d_k} + a_{i,i}\, \beta_i^l + \underline{\mathbf{b}}_{i\circ}\, \underline{\tilde{\mathbf{u}}} \right) \tag{11}$$

Special attention must be paid to the case where boundaries are at ± infinity. Let $\underline{\hat{\mathbf{a}}}_{i\circ}$ be the i-th row vector of $\underline{\underline{\mathbf{A}}}$ with

$$\hat{a}_{i,k} = \begin{cases} a_{i,k} & \text{if} \quad \beta_i^l = \infty \\ 0 & \text{if} \quad \beta_i^l \neq \pm\infty \\ a_{i,k} & \text{if} \quad \beta_i^l := -\infty \end{cases} \tag{12}$$

Three cases must be considered :

$$x_i^{i,l} = \begin{cases} -\infty & \text{if} \quad -\sum_k \operatorname{sign}(\beta_k)\, \hat{a}_{i,k} < 0 \\ 0 & \text{if} \quad -\sum_k \operatorname{sign}(\beta_k)\, \hat{a}_{i,k} = 0 \\ +\infty & \text{if} \quad -\sum_k \operatorname{sign}(\beta_k)\, \hat{a}_{i,k} > 0 \end{cases} \tag{13}$$

The minimum and maximum is readily selected.

[3] The representation, as it is done here, maps the space out completely. The consequence is obvious : With the number of dimensions and the number of discretisations the dimensionality of the discrete space grows very rapidly, a problem known as the state-explosion problem. On the other hand, if for no other reasons than for safety any model must be complete. In this context, the state explosion problem must be approached through a hierarchical approach and through "constraint modelling" as it was briefly introduced in [3].

Filling the Automanton Table

The automaton table can now be filled readily. A whole set of entries can be filled in simultaneously, namely for the three sections of the state variable x_j. The lower section and the upper section, assuming that the maximum is not at infinity and that the minimum is bigger than negative infinity, is :

$$\{\tilde{x}_j\}^l := \{1, k+1\} \quad ; \quad \overline{x}_j^{i,l} \in [\beta_j^k, \beta_j^{k+1}) \tag{14}$$

$$\{\tilde{x}_j\}^u := \{k, n_j - 1\} \quad ; \quad \underline{x}_j^{i,l} \in (\beta_j^k, \beta_j^{k+1}] \tag{15}$$

The two sets overlap, which is the section in which both transitions are possible. The other discrete variables are all fixed. If the transition across this upper section of the boundary is positive, which again is determined by the sign of the coefficient $a_{i,j}$, the transitions are

$$\{\mathrm{O}[\tilde{x}_i := l-1, \tilde{x}_j, \{d_m | \forall m, m \neq \{i,j\}\}]\} \rightarrow \{\mathrm{O}[\tilde{x}_i := l, \tilde{x}_j, \{d_m | \forall m, m \neq \{i,j\}\}]\} \quad \forall \tilde{x}_j \in \{\tilde{x}_j\}^u \tag{16}$$

and for the lower section :

$$\{\mathrm{O}[\tilde{x}_i := l, \tilde{x}_j, \{d_m | \forall m, m \neq \{i,j\}\}]\} \rightarrow \{\mathrm{O}[\tilde{x}_i := l-1, \tilde{x}_j, \{d_m | \forall m, m \neq \{i,j\}\}]\} \quad \forall \tilde{x}_j \in \{\tilde{x}_j\}^l \tag{17}$$

The operator $O(\diamond)$ orders the components of the state vector. In practice it is implemented through index mapping.

Dealing with Exceptions

This algorithm frequently runs into two problems. The first one is due to orthogonal hyper-planes. In this case, the width of the band is zero and the two boundaries are best replace by a "void band" enforcing that the two sections do not overlap. This "void band" is defined by making the minimum **bigger** by a small quantity, being the machine constant, and the maximum **smaller** by the same small quantity. If it happens that the intersection of the hyper-plane with the boundary coincides with the intersection of the boundaries the void band also excludes the cross-over point of the two boundaries. The second problem occurs when the state equations are decoupled. In that case the bands are running in parallel with the boundaries and do not intersect. The analysis for the completely decoupled system is straightforward. Below we show the analysis for the more complex case, where the states are only partially decoupled, that is some elements in the system matrix are zero.

Given a state-space model of a system, where one variable x_j is decoupled, two cases can be distinguished. **First case**: Element $a_{j,j}$ is non-zero. In this case the j^{th} state equation is set equal to zero

$$\dot{x}_i = a_{j,j} x_j + \underline{\mathbf{b}}_{j\circ} \underline{\tilde{\mathbf{u}}} = 0 \tag{18}$$

The solution of equation (18) is the value $x_{i,0}$

$$x_{j,0} = \frac{-1}{a_{j,j}} \left(\underline{\mathbf{b}}_{j\circ} \underline{\tilde{\mathbf{u}}}\right) \tag{19}$$

The line $x_j = x_{j,0}$, which we call an equilibrium line for the direction x_j, splits the boundaries along x_j into two sections: One section where the sign in the x_j direction is positive and one section where the sign is negative. The sign of possible transitions in the x_i direction is given by the sign of $\dot{x}_i$ above and below $x_{i,0}$

$$\mathrm{sign}(\dot{x}_j) = \mathrm{sign}\left(a_{j,j}\left(x_{j,0} + \delta x_j\right) + \underline{\mathbf{b}}_{j\circ} \underline{\tilde{\mathbf{u}}}\right) = \mathrm{sign}(a_{j,j}) \cdot \mathrm{sign}(\delta x_j) \tag{20}$$

In equation (20) (decoupled situation) the variations in x_j are examined while in equation (8) (coupled situation) only the variations in x_i, $i \neq j$ are analysed. Here the equilibrium line is orthogonal to x_j thus the domain where the $\mathrm{sign}(\dot{x}_i) := \pm 1$ applies to all x_j.

A **second case** in the decoupled situation occurs if every element of the j^{th} row in the system matrix $\underline{\underline{\mathbf{A}}}$ is equal to zero. In this case a solution of equation (18) only exists if:

$$\dot{x}_j = 0 \Rightarrow \underline{\mathbf{b}}_{j\circ} \underline{\tilde{\mathbf{u}}} = 0 \tag{21}$$

The state-space is not split in two sections i.e. the direction of the possible transitions has everywhere the same sign:

$$\mathrm{sign}(\dot{x}_j) = \mathrm{sign}\left(\underline{\mathbf{b}}_{j\circ} \underline{\tilde{\mathbf{u}}}\right) \tag{22}$$

Implementation

Our implementation of the algorithm in MatLab requires the two system matrices $\underline{\underline{\mathbf{A}}}, \underline{\underline{\mathbf{B}}}$, the discrete input for which the entries should be computed and the sets of boundaries. It computes one column of the automaton table, namely all possible state transitions, for all discrete states given a particular command input $\bar{\underline{\mathbf{u}}}$. By looping through all possible inputs, the complete table is generated. For simplicity of the data structure, the implementation gives the event-discrete derivative for each state variable in coded form ($\{1, 2, 3\}$ for $\{-1, +1, \pm 1\}$).

Extenstions

The algorithm can be readily extended to the case where the plant is nonlinear in the command inputs. The plant model can then be of the general form:

$$\dot{\underline{\mathbf{x}}} = \underline{\underline{\mathbf{A}}}(\bar{\underline{\mathbf{u}}})\,\underline{\mathbf{x}} + \underline{\underline{\mathbf{B}}}(\bar{\underline{\mathbf{u}}}) \tag{23}$$

Extensions to the completely nonlinear case have been done but refinements are still in progress.

Conclusions

The objective of this research was to develop an algorithm that maps a hybrid plant into a complete discrete-event dynamic representation, given the plant as a linear, time-constant set of ordinary differential equations, given the state discretisation and given the discrete-event input commands to the plant. The representation is required to be complete, that is the whole state space is mapped out and all transitions are being computed.

An algorithm that achieves this goal has been presented. It builds directly on the basic definition of an event, namely the crossing of a boundary separating two section of the continuous state. The algorithm does not require integration, but is simply analysing the directionality of the trajectories crossing the boundaries. The existence of points on the boundary where a transition may take place is computed resulting in a corresponding entry in the non-deterministic automaton table.

References

1. Engell, S, S. Kowalewski & B.H. Krogh; Discrete Events and Hybrid Systems in Process Control, CPC V, 1995.
2. Lunze, J.; Qualitative Modelling of Linear Dynamical Systems with Quantized State Measurements; Automatica, Vol 30, No 3, pp 417-413; 1994.
3. Preisig, H.A.; Event-Discrete Modelling of Manufacturing Systems : Reduction of the State Space; Proceedings of I-CIMPRO'96, June 3-4, Eindhoven, The Netherlands; (1996);pp 434-443.
4. Preisig, H.A.; A Mathematical Approach to Discrete-Event Dynamic Modelling of Hybrid Systems; Comp. & Chem. Eng.; Vol 20; (1996); pp S1301-S1306.
5. Preisig, H.A.; More on the Synthesis of as Supervisory Controller From First Principles; Proceedings IFAC World Congress; Sydney, Australia; 1993; Vol V, p 275.
6. Preisig H.A.; Discrete-Event Controlled Systems in the Chemical Processing Industry; Proceedings of DYCORD+92, IFAC Symposium on Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes; College Park, Maryland, USA; 1992; 277.

NONDETERMINISTIC AUTOMATA AS APPROXIMATIONS FOR CONTINUOUS SYSTEMS - AN APPROACH WITH AN ADJUSTABLE DEGREE OF ACCURACY

J. Raisch
Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart
Pfaffenwaldring 9, D-70550 Stuttgart, Fed. Rep. of Germany
Tel.: (49 (0)711) 685-6194, Fax: (49 (0)711) 685-6371, email: raisch@isr.uni-stuttgart.de

Abstract. This contribution addresses the following problem: a continuous dynamical system (state evolving in $\mathbb{R}^n$) with quantized measurements and symbolic control inputs is to be approximated by a discrete model. The central features of the proposed solution scheme are: 1) the approximating model is a nondeterministic automaton; 2) the automaton retains timing information by incorporating a clock which is synchronized with the sampling grid of the underlying continuous system; 3) its degree of accuracy is *not* completely determined by the coarseness of measurement quantization, but can be adjusted to suit various specifications. Furthermore, it is shown that the set of approximating automata obtained by this method is totally ordered (in the sense of approximation accuracy), and that this ordering is preserved under feedback. The latter result provides theoretical justification for applying this approximation scheme when designing discrete (supervisory) controllers for continuous or hybrid plants.

1. Introduction.

The motivation for the work presented here stems from the following "standard problem" in hybrid control systems design: the plant state evolves in $\mathbb{R}^n$ and is affected by real-valued unknown but bounded disturbances, whereas control input and measurement signals are discrete-valued, or symbolic[1]. One is then concerned with finding an appropriate feedback structure mapping symbolic measurement signals into (sets of) symbolic control inputs. Such problems frequently arise in chemical process control (and other areas of application): process models typically have a continuous state space, but continuous *and* discrete control inputs and measurement signals. Often, there is also a clearly defined hierarchical structure where feedback loops from continuous measurement to continuous control variables are interpreted to be "low level" (dealing, for example, with set-point regulation), and discrete signals are used for "high level", or supervisory, control. Examples for the latter are start-up/shut-down procedures, handling of "irregularities" (represented by discrete alarm signals), and, more generally, the "protection layer" in chemical process control systems [7]. Subsuming plant model and continuous control loops into one continuous entity (henceforth simply referred to as "the plant") gives the "standard problem" described above. There are a number of ways to tackle this problem (see the "Hybrid Systems" volumes [4, 3, 1]): the approach suggested in [12, 2, 5, 6] is especially useful if design specifications are in terms of discrete variables only; it is based on approximating the continuous plant by a finite state machine, hence converting the hybrid control problem into a purely discrete one, which can subsequently be solved using standard tools from DES (discrete event systems) theory.

This contribution is based on joint work with *S. O'Young* [8, 9]. Its purpose is to explain and refine the notion of "approximating a continuous system by a finite state machine". In particular, it will be shown 1) how timing information can be retained on the discrete level by incorporating a clock which is synchronized with the sampling grid of the underlying continuous system; 2) how approximation accuracy can be increased at the expense of model complexity. Furthermore, it will be demonstrated that the model set consisting of the continuous "base" system and its approximating automata is totally ordered, and that this order is invariant under feedback. It will be argued that it is the latter result which provides theoretical justification for using discrete approximation as an integral step in hybrid control systems design.

[1] The problem will be discussed in a discrete-time framework, i.e. the domain of all signals will be $\{t_0, t_1, \ldots\}$, with $t_i - t_{i-1}$ being constant. This understood, the adjectives "discrete" and "continuous" will in the sequel only be used to refer to the *codomain* of signals: the codomain of a discrete, or discrete-valued, signal is a set of symbols, which, for our purposes, will be assumed to be finite (e.g. { "valve open", "valve closed" } or { "liquid level too low", "ok", "too high" }). The codomain of a continuous, or continuous-valued, signal are the real numbers.

The paper is organized as follows: In Section 2, the continuous plant model is introduced. Section 3 explains the key property of any "sensible" discrete approximation in terms of model behaviours. Section 4 describes the actual approximation algorithm. The results on ordering are stated in Section 5.

2. The Continuous Plant Model.

Let the plant be modelled by a discrete-time dynamic system in (non-linear) observer form[2]:

$$\begin{aligned} x[k+1] &= f(y_d[k], u_d[k])x[k] + g(y_d[k], u_d[k]) + h(y_d[k], u_d[k])w[k] \;, && (1) \\ y_d[k] &= Q_y(C_y x[k]) := q_y(x[k]) \;, && (2) \end{aligned}$$

where $k \in \{0, 1, 2, \dots\}$ is the time index, $x[k] \in \mathbb{R}^n$ the state at time t_k, and $w[k] \in \mathbb{R}^r$ an unknown but bounded disturbance: $w[k] \in W := \{w \mid w \in \mathbb{R}^r, \|w\|_\infty \leq 1\}$ with $\|w\|_\infty := \max_i |w_i|$. $u_d[k] \in U_d$ and $y_d[k] \in Y_d$ are control and measurement symbols, respectively. The sets U_d and Y_d are finite:

$$U_d = \{u_d^{(1)}, \dots, u_d^{(\alpha)}\}, \quad Y_d = \{y_d^{(1)}, \dots, y_d^{(\gamma)}\}.$$

The only assumption regarding the functions $f : Y_d \times U_d \to \mathbb{R}^{n \times n}$, $h : Y_d \times U_d \to \mathbb{R}^{n \times r}$, and $g : Y_d \times U_d \to \mathbb{R}^n$ is that the "combined" function $(f, g, h) : Y_d \times U_d \to \mathbb{R}^{n \times n} \times \mathbb{R}^n \times \mathbb{R}^{n \times r}$ is injective. The measurement map $q_y : \mathbb{R}^n \to Y_d$ is onto. It induces an equivalence relation on $\mathbb{R}^n$; its cosets (or equivalence classes) are referred to as y-cells. C_y in (2) is a real $p \times n$-matrix. The "quantizer" Q_y partitions $\mathbb{R}^p$ into finitely many rectangular boxes with edges parallel to the coordinate axes.

To keep exposition as simple as possible, it is assumed that the measurement signal is the only system output. This implies that specifications will have to be formulated in terms of control and measurement symbols only. In general, however, it might be more convenient to introduce another (non-measurable) output signal, $z_d[k]$, which can then be used to specify desired closed-loop properties. This is a straightforward extension of the current framework; details can be found in [9, 10].

3. Behaviours.

Let $T \subset \mathbb{R}$ be the chosen sampling grid, i.e. $T = \{t_0, t_1, \dots\}$. Denote the set of all functions from T into $(Y_d \times U_d)$ by $(Y_d \times U_d)^T$. Then $\mathcal{B}_c \subseteq (Y_d \times U_d)^T$ and $\mathcal{B}_d \subseteq (Y_d \times U_d)^T$ are the sets of input/output trajectories which can be generated by the continuous model (1) - (2) and a (not yet specified) discrete approximation, respectively (see [13] for a survey on "behavioural" systems theory).

Clearly, a *conditio sine qua non* for *any* discrete approximation is that its behaviour $\mathcal{B}_d$ must contain the discrete-time behaviour $\mathcal{B}_c$ of the underlying continuous model: $\mathcal{B}_c \subseteq \mathcal{B}_d$ implies that every sequence of input/measurement symbols that the continuous plant model can generate, can also be produced by the discrete approximation. If this condition were violated, the continuous system could respond to a given control signal with an unacceptable output which would not be predictable by the discrete approximation. Hence, this unacceptable output could not be suppressed by a control strategy based on the discrete approximation – the approximation would be useless for the purposes of control systems design.

In general, $\mathcal{B}_c$ is a *proper* subset of $\mathcal{B}_d$, and the "smaller" the difference $\mathcal{B}_d \setminus \mathcal{B}_c$, the more accurate the discrete approximation. Equality of the two sets would imply that the discrete model (which, for our purposes, will always be finite-state) and the underlying continuous model (by definition infinite-state) exhibit exactly the same behaviour on the chosen sampling grid. In other words, the continuous model could be reduced, or abstracted, to a discrete one without any loss of accuracy.

In the next section, we propose an approximation scheme which, as will be shown in Section 5, satisfies the key condition $\mathcal{B}_c \subseteq \mathcal{B}_d$. Moreover, it allows adjusting the "size" of $\mathcal{B}_d \setminus \mathcal{B}_c$ via an integer design parameter.

[2] As only a small class of nonlinear systems can be transformed into observer form, this assumption may seem quite restrictive. However, there are a number of reasons why it makes sense: first, it can be argued that the model may be of less than full generality because we admit uncertainty. In this paper, for lack of space, we only address the problem of signal uncertainty (unknown disturbances), but (norm-bounded) model uncertainty can be treated with equal ease within the proposed framework [10]. Second, the plant model covers the practically important case of switched linear systems; third, the assumptions guarantee that the translation from the continuous plant model to the approximating automaton in Section 4 is computationally straightforward: basically, it reduces to checking for the existence of solutions to a set of *linear* inequalities. Conceptually, everything remains the same if we admit a more general model. On the computational level, however, we would have to deal with sets of nonlinear inequalities, and what can be said now, in definite terms, about solvability would mostly change into speculation.

4. Nondeterministic Automata as Discrete Approximations.

At time t_k, the observed plant behaviour consists of a string of measurement symbols, $y_d[k], \ldots, y_d[0]$, and a string of previously applied control symbols, $u_d[k-1], \ldots, u_d[0]$. For practical reasons, we discard "old" data and work with finite strings of symbols not exceeding a given maximum length.

$$s^*[k] := \begin{cases} ([y_d[k], \ldots, y_d[0]], [u_d[k-1], \ldots, u_d[0]]), & \text{if } k = 0, 1, \ldots, v-1 \\ ([y_d[k], \ldots, y_d[k-v]], [u_d[k-1], \ldots, u_d[k-v]]), & \text{if } k \geq v \end{cases} \tag{3}$$

is called the *current history* of the process, if t_k represents the present sampling instant. If t_k is *any* (not necessarily the present) sampling instant, $s^*[k]$ is simply referred to as a *history* of the process at time t_k. It is important to note that, in our terminology, the *current history* consists of observed data, whereas a *history* is just a postulated (and therefore nonunique) collection of symbols from U_d and Y_d of the form (3). $y_d^*[k]$ and $u_d^*[k-1]$ are the strings of measurement and control symbols in $s^*[k]$. The "forgetting operator" $\mathcal{F}$ deletes the "oldest" symbol from strings $y_d^*[k]$ and $u_d^*[k-1]$, if $k \geq v$:

$$\mathcal{F}(y_d^*[k]) := \begin{cases} [y_d[k], \ldots, y_d[0]], & \text{if } k = 0, 1, \ldots, v-1 \\ [y_d[k], \ldots, y_d[k-v+1]], & \text{if } k \geq v. \end{cases} \tag{4}$$

Definition 1 *The history $s^*[k+1]$ is a* successor *of $s^*[k]$ if there exists a symbol $y_d^{(i)} \in Y_d$ and an input $u_d^{(j)} \in U_d$ such that $y_d^*[k+1] = [y_d^{(i)}\ ,\ \mathcal{F}(y_d^*[k])]$ and $u_d^*[k] = [u_d^{(j)}\ ,\ \mathcal{F}(u_d^*[k-1])]$. Similarly, $s^*[k]$ is referred to as a* predecessor *of $s^*[k+1]$.*

Definition 2 *A history $s^*[k]$ is called* feasible *if its strings of input and measurement symbols are compatible with the plant model (1), (2), i.e. if there exists an $x[\max(k-v,0)] \in \mathbb{R}^n$ and disturbances $w[i] \in W$, $i = \max(k-v,0), \ldots, k-1$, such that applying the input string $u_d^*[k-1]$ actually produces $y_d^*[k]$ as string of measurements.*

The computational procedure to determine whether a history is feasible is very much straightforward: Consider the quantization box corresponding to a certain measurement symbol $y_d[k]$. Denote the vectors of its upper and lower bounds by $\hat{y}[k]$ and $\check{y}[k]$, respectively: $\check{y}[k] < \{\zeta | Q_y(\zeta) = y_d[k]\} \leq \hat{y}[k]$, where the "$<$" and "$\leq$"-signs are understood to be elementwise. Elements of $\hat{y}[k]$ can be $+\infty$; elements of $\check{y}[k]$ may be $-\infty$. $\rho := \max(k-v, 0)$. The case $k = 0$ is trivial. For $k \geq 1$, define

$$\hat{y}_P := \begin{bmatrix} \hat{y}[k-1] \\ \vdots \\ \hat{y}[\rho] \end{bmatrix}, \quad \check{y}_P := \begin{bmatrix} \check{y}[k-1] \\ \vdots \\ \check{y}[\rho] \end{bmatrix}, \quad g_P := \begin{bmatrix} g_{k-1} \\ \vdots \\ g_\rho \end{bmatrix}, \quad w_P := \begin{bmatrix} w[k-1] \\ \vdots \\ w[\rho] \end{bmatrix},$$

i.e. collect the "forcing terms" $g_{k-1} := g(y_d[k-1], u_d[k-1]), \ldots, g_\rho := g(y_d[\rho], u_d[\rho])$ in g_P and the (unknown) disturbance inputs $w[k-1], \ldots, w[\rho]$ in w_P. 1_r and I_r denote a column vector with $\min(k,v)r = (k-\rho)r$ "ones" and the $(\min(k,v)r = (k-\rho)r)$-dimensional identity matrix. f_i and h_i are short for (the matrices) $f(y_d[i], u_d[i])$ and $h(y_d[i], u_d[i])$. Then, the history $s^*[k]$ is feasible if and only if the set of solutions $[x'[k]\ w_P']'$ for the following linear inequality is nonempty:

$$\begin{bmatrix} \check{y}[k] \\ \check{y}_P \\ -1_r \end{bmatrix} + \begin{bmatrix} 0 \\ \Phi_{PU} \\ 0 \end{bmatrix} g_P < \begin{bmatrix} C_y & 0 \\ \begin{bmatrix} C_y f_{k-1}^{-1} \\ \vdots \\ C_y \prod_{i=\rho}^{k-1} f_i^{-1} \end{bmatrix} & \Phi_{PW} \\ 0 & I_r \end{bmatrix} \begin{bmatrix} x[k] \\ w_P \end{bmatrix} \leq \begin{bmatrix} \hat{y}[k] \\ \hat{y}_P \\ 1_r \end{bmatrix} + \begin{bmatrix} 0 \\ \Phi_{PU} \\ 0 \end{bmatrix} g_P \tag{5}$$

where

$$\Phi_{PU} := \begin{bmatrix} C_y f_{k-1}^{-1} & 0 & \ldots & 0 \\ C_y f_{k-2}^{-1} f_{k-1}^{-1} & C_y f_{k-2}^{-1} & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ C_y \prod_{i=\rho}^{k-1} f_i^{-1} & \ldots & \ldots & C_y f_\rho^{-1} \end{bmatrix}, \quad \Phi_{PW} := -\Phi_{PU} \begin{bmatrix} h_{k-1} & 0 & \ldots & 0 \\ 0 & h_{k-2} & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & \ldots & h_\rho \end{bmatrix},$$

and $\prod$ means "right product" (i.e.:$\prod_{i=\rho}^{k-1} f_i^{-1} = f_\rho^{-1} \dots f_{k-1}^{-1}$). Existence of a solution for this set of inequalities can be checked using the "feasibility part" of any linear programming algorithm.

Now, we are in a position to introduce a finite state (Moore) machine as our discrete plant model: its *state set*, $X_d := \{x_d^{(1)}, \dots, x_d^{(N)}\}$, is the set of all feasible histories (3). We denote the strings of control and measurement symbols associated with a particular $x_d^{(i)}$ by $u^*(x_d^{(i)})$ and $y^*(x_d^{(i)})$, respectively. $x_d^{(i)}$ belongs to the set of possible initial states, if $y^*(x_d^{(i)})$ is a string of length one (and $u^*(x_d^{(i)})$ a string of length zero). This reflects the fact that, before time t_0 (when we switch the control system on), we do not have any control over the plant. Hence, $x[0]$ can be anywhere in the continuous plant state set, and any measurement symbol in Y_d is possible at time t_0. $(x_d^{(i)}, u_d^{(j)}, x_d^{(k)})$ is a *transition* iff $x_d^{(k)}$ is a successor of $x_d^{(i)}$ and $u_d^{(j)}$ is the leftmost symbol in the string $u^*(x_d^{(k)})$. The control symbol $u_d^{(j)}$ is called the *transition label*. The measured ***output*** in each state $x_d^{(i)}$ is simply the leftmost symbol in $y^*(x_d^{(i)})$. It is clear that in most cases the resulting finite state machine will be nondeterministic: several transitions exiting from a state may carry the same label. This is illustrated in the example below. To introduce explicit time, this transition structure has to be augmented by a simple clock process which enforces that any two control symbols are "separated" by a `tick` event, the latter representing the passage of one sampling interval (see the example below). The overall (timed) discrete approximation is then simply the synchronous composition of the two state machines.

Example. Suppose, we are dealing with a system with two measurement symbols ($\gamma = 2$) and two control symbols ($\alpha = 2$). Let $v = 1$. Then, we have $2 + 2^3 = 10$ histories. Suppose furthermore that checking the set of linear inequalities (5) for the existence of solutions leaves us with 8 feasible histories, i.e. 8 states in our automaton:

$$\begin{array}{llll}
x_d^{(1)} := [y_d^{(1)}] & x_d^{(3)} := [[y_d^{(1)}, y_d^{(1)}], u_d^{(1)}] & x_d^{(5)} := [[y_d^{(1)}, y_d^{(2)}], u_d^{(2)}] & x_d^{(7)} := [[y_d^{(2)}, y_d^{(2)}], u_d^{(1)}] \\
x_d^{(2)} := [y_d^{(2)}] & x_d^{(4)} := [[y_d^{(1)}, y_d^{(1)}], u_d^{(2)}] & x_d^{(6)} := [[y_d^{(2)}, y_d^{(1)}], u_d^{(1)}] & x_d^{(8)} := [[y_d^{(2)}, y_d^{(2)}], u_d^{(2)}]
\end{array}$$

Then, the (Moore) transition structure follows by inspection. It is shown in the left part of Fig. 1. Initial states are shaded; states which produce a $y_d^{(1)}$-output are shown as squares, states generating a $y_d^{(2)}$-symbol as circles. $u_d^{(1)}$-transitions are drawn as solid lines, $u_d^{(2)}$-transitions as broken lines. The same conventions are used for the clock in the right part of Fig. 1: the initial state is shaded; $u_d^{(1)}$- and $u_d^{(2)}$-transitions are represented by a solid and a broken line, respectively.

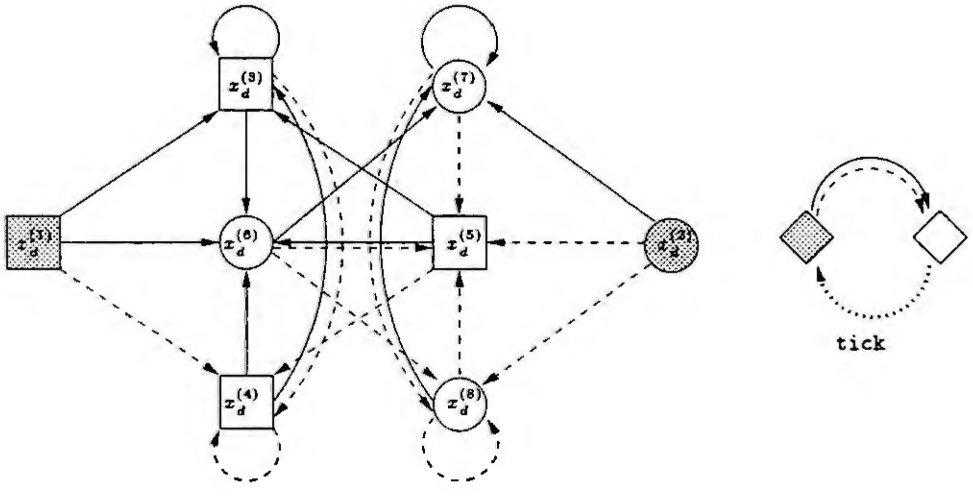


Figure 1: Moore transition structure (left) and clock (right).

It is obvious that a given history is feasible (and hence constitutes a state $x_d^{(j)}$ in our transition structure) iff the set of all continuous plant states $x[k]$ which are compatible with the history and the disturbance assumptions in Section 2 is non-empty. Such a (non-empty) set will be denoted by $X(x_d^{(j)})$, $j = 1, \dots, N$, and it can be interpreted as a set-valued estimate of the underlying continuous plant state based on the (observed) history $x_d^{(j)}$. Computationally, it is just the projection of the set of solution vectors $[x'[k]\ w'_P]'$ of (5) onto the first n components. Clearly, $\bigcup_{j=1}^{N} X(x_d^{(j)}) = \mathbb{R}^n$, and, in general, $X(x_d^{(i)}) \cap X(x_d^{(j)}) \neq \emptyset$. Hence, the sets $X(x_d^{(j)})$ form a *cover* of the continuous plant state set. Intuitively, the smaller the sets $X(x_d^{(j)})$, the more accurate the discrete approximation (the notion of accuracy will be made precise in the next section). Obviously, a state set $X(x_d^{(j)})$ never increases (and, in general, decreases) in "size", if the history $x_d^{(j)}$ is extended further into the past: hence, increasing v, the maximum length of histories, is equivalent to generating a finer "granularity" for the finite cover of $\mathbb{R}^n$ – v can be seen as a design parameter, which may be used to improve the accuracy of the discrete model. This of course implies that the number of states, and hence the complexity of the discrete model, also increases.

This constitutes a major refinement when compared to the approximation procedures in [2, 12, 5]. These approaches correspond to the special case $v = 0$ – they are based on *partitioning* $\mathbb{R}^n$ via the measurement map $q_y : \mathbb{R}^n \to Y_d$: all states which are mapped to the same measurement symbol $y_d \in Y_d$ are "lumped"; the cosets of the measurement map can be interpreted as the state set of the discrete approximation. Hence, the maximal accuracy of the approximation is completely determined by the measurement map q_y.

5. Ordering and Approximation Accuracy.

The issue of approximation accuracy will now be formalized and investigated in terms of model behaviours. Recall that $\mathcal{B}_c$ has been defined as the behaviour of the underlying continuous model. Refer to the approximating automaton obtained by choosing the design parameter as $v = 0, 1, \ldots,$ as the "discrete model v", and denote its behaviour by $\mathcal{B}_{dv}$. Model v_i is called "more accurate" than model v_j if $\mathcal{B}_{dv_i} \subseteq \mathcal{B}_{dv_j}$ – it predicts the future behaviour more precisely. Although not very surprising, the following result is important:

Theorem 1 $\mathcal{B}_c \subseteq \mathcal{B}_{dv_i} \subseteq \mathcal{B}_{dv_j}; \quad v_i, v_j = 0, 1, \ldots; \quad v_i \geq v_j.$

Theorem 1 implies that, in terms of accuracy, the continuous "base" model and its discrete approximations form a totally ordered set. This result is illustrated in the left part of Fig. 2.

Proof: The proof is by induction. Denote the behaviours of the continuous system and the discrete model v_i up to time t_k by $\mathcal{B}_c[k]$ and $\mathcal{B}_{dv_i}[k]$, respectively. Hence, $\mathcal{B}_c[k], \mathcal{B}_{dv_i}[k] \subseteq (Y_d \times U_d)^{\{t_0, t_1, \ldots, t_k\}}$.

1. $\mathcal{B}_c[0] \subseteq \mathcal{B}_{dv_i}[0] \subseteq \mathcal{B}_{dv_j}[0]$; $v_i, v_j = 0, 1, \ldots$; $v_i \geq v_j$. This is trivial because $\mathcal{B}_c[0] = \mathcal{B}_{dv_i}[0] = \mathcal{B}_{dv_j}[0] = (Y_d \times U_d)$ – there is no a-priori information on the system state, therefore any measurement symbol in Y_d can occur at time t_0. By definition, there is no restriction on the control input, hence any $u_d[0] \in U_d$ is possible.

2. Assume $\mathcal{B}_c[k-1] \subseteq \mathcal{B}_{dv_i}[k-1] \subseteq \mathcal{B}_{dv_j}[k-1]$; $v_i, v_j = 0, 1, \ldots$; $v_i \geq v_j$. Pick any element $b^*[k-1] = [(y_d[0], u_d[0]), \ldots, (y_d[k-1], u_d[k-1]) \in \mathcal{B}_c[k-1]$. Then we need to show that for any such $b^*[k-1]$ and any "extension" $(y_d[k], u_d[k]) \in (Y_d \times U_d)$ the following implications hold:

$$\begin{aligned} [b^*[k-1], (y_d[k], u_d[k])] \in \mathcal{B}_c[k] \quad &\Rightarrow \quad [b^*[k-1], (y_d[k], u_d[k])] \in \mathcal{B}_{dv_i}[k], \qquad (6)\\ &\Rightarrow \quad [b^*[k-1], (y_d[k], u_d[k])] \in \mathcal{B}_{dv_j}[k], \quad v_i \geq v_j. \qquad (7) \end{aligned}$$

By definition, none of the models imposes any restrictions on the control symbol $u_d[k]$. Therefore, extension of $b^*[k-1]$ by $u_d[k] \in U_d$ does not jeopardize membership in any of the model behaviours, and (6) and (7) need only be shown with respect to the measurement symbol $y_d[k]$. The three statements in (6) and (7) are equivalent to saying that (5) has a non-empty solution set for $\rho = 0$, $\rho = \max(k - v_i, 0)$, and $\rho = \max(k - v_j, 0)$, respectively. Clearly, $0 \leq \max(k - v_i, 0) \leq \max(k - v_j, 0)$. Because of the special matrix structure in (5), increasing ρ is equivalent to omitting a number of inequalities. Hence, if the solution set is non-empty for $\rho = 0$ (for $\rho = \max(k - v_i, 0)$), it will also be non-empty for $\rho = \max(k - v_i, 0)$ (for $\rho = \max(k - v_j, 0)$). This shows that (6) and (7) hold and concludes the proof.

Now, suppose we come up with a supervisory controller[3] design based on the discrete model v. If the design is any good, the supervised approximating automaton satisfies the specifications, i.e. it is guaranteed to exhibit a certain "desired" behaviour and to avoid certain "forbidden" patterns. In terms of its behaviour, $\mathcal{B}_{dv_S}$ (the subscript "S" indicates that the model is under supervision), this can be written as: $\mathcal{B}_{dv_S} \subseteq \mathcal{B}_{\text{desired}}$ and $\mathcal{B}_{dv_S} \cap \mathcal{B}_{\text{forbidden}} = \emptyset$. Does the continuous "base" system, when subjected to the *same* control law, also satisfy the specifications, i.e. does $\mathcal{B}_{c_S} \subseteq \mathcal{B}_{\text{desired}}$ and $\mathcal{B}_{c_S} \cap \mathcal{B}_{\text{forbidden}} = \emptyset$ hold? Or, in other words, does it make sense to base the design of symbolic feedback controllers for continuous systems on a discrete approximation? As the following result shows, this is indeed the case (see also the right part of Fig. 2):

Theorem 2 *Suppose both continuous system (1) - (2) and discrete approximating models v_i are subjected to the same feedback law. Then, their supervised behaviours are ordered in the following sense:*

$$\mathcal{B}_{c_S} \subseteq \mathcal{B}_{dv_{iS}} \subseteq \mathcal{B}_{dv_{jS}}, \; v_i, v_j = 0, 1, \ldots, \; v_i \geq v_j.$$

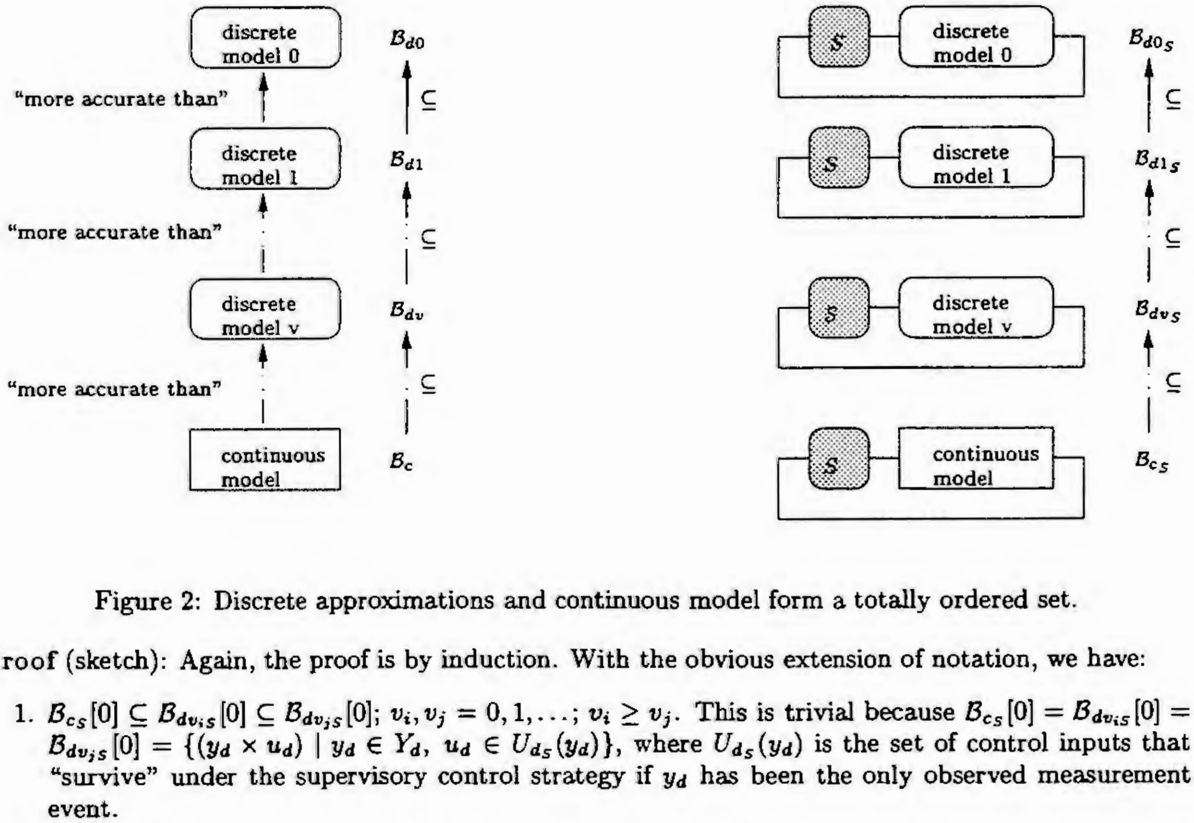


Figure 2: Discrete approximations and continuous model form a totally ordered set.

Proof (sketch): Again, the proof is by induction. With the obvious extension of notation, we have:

1. $\mathcal{B}_{c_S}[0] \subseteq \mathcal{B}_{dv_{iS}}[0] \subseteq \mathcal{B}_{dv_{jS}}[0]$; $v_i, v_j = 0, 1, \ldots$; $v_i \geq v_j$. This is trivial because $\mathcal{B}_{c_S}[0] = \mathcal{B}_{dv_{iS}}[0] = \mathcal{B}_{dv_{jS}}[0] = \{(y_d \times u_d) \mid y_d \in Y_d, \; u_d \in U_{d_S}(y_d)\}$, where $U_{d_S}(y_d)$ is the set of control inputs that "survive" under the supervisory control strategy if y_d has been the only observed measurement event.

2. Assume $\mathcal{B}_{c_S}[k-1] \subseteq \mathcal{B}_{dv_{iS}}[k-1] \subseteq \mathcal{B}_{dv_{jS}}[k-1]$; $v_i, v_j = 0, 1, \ldots$; $v_i \geq v_j$. Pick any element $b^*[k-1] = [(y_d[0], u_d[0]), \ldots, (y_d[k-1], u_d[k-1]) \in \mathcal{B}_{c_S}[k-1]$. Then we need to show that for any such $b^*[k-1]$ and any "extension" $(y_d[k], u_d[k]) \in \{(y_d, u_d) \mid y_d \in Y_d, \; u_d \in U_{d_S}(y_d, b^*[k-1])\}$ the following implications hold:

$$\begin{aligned} [b^*[k-1], (y_d[k], u_d[k])] \in \mathcal{B}_{c_S}[k] \quad &\Rightarrow \quad [b^*[k-1], (y_d[k], u_d[k])] \in \mathcal{B}_{dv_{iS}}[k], && (8) \\ &\Rightarrow \quad [b^*[k-1], (y_d[k], u_d[k])] \in \mathcal{B}_{dv_{jS}}[k], \quad v_i \geq v_j. && (9) \end{aligned}$$

This can be done in exactly the same way as in the previous proof.

[3] the term "supervisory control" is from the DES (discrete event systems) literature. It refers to the situation where feedback does *not* uniquely define the next control input, but merely narrows the choice to a subset of U_d. Hence, it is more general than the traditional notion of feedback: past and present measurement information are not mapped into U_d, but the power set (the set of all subsets) of U_d. See the standard reference [11] for more details.

6. Conclusions.

In this contribution it has been shown how to approximate a continuous dynamical system with quantized measurements and symbolic control inputs by a discrete model. The approximating model is a nondeterministic automaton that captures the notion of time and allows the degree of accuracy to be adjusted via a design parameter. It has been proven that the behaviour of the underlying continuous model is contained in the behaviour of the approximating automaton, and that this inclusion is preserved under feedback. Hence, any feedback law that forces the approximating automaton to obey a given set of (timed or untimed) specifications, will also guarantee that the continuous "base" system meets the specifications. In other words: the design of a discrete (supervisory) feedback structure for a continuous system can be based on a discrete approximation. An important open question is: What is the least accurate (the least complex) discrete approximation which still allows the specifications to be met? This is the subject of current investigations.

Acknowledgments. The author wishes to thank Siu O'Young, University of Toronto, and Eberhard Klein, Universität Stuttgart, for many helpful discussions. This work was done while the author visited the Systems Control Group at the University of Toronto. Support from Deutsche Forschungsgemeinschaft under grants Ra 516/2-1 and Ra 516/2-2 is gratefully acknowledged.

References.

[1] R. Alur, T. A. Henzinger, and E. D. Sontag, editors. *Hybrid Systems III*, Lecture Notes in Computer Science, Vol. 1066. Springer-Verlag, 1996.

[2] P. J. Antsaklis, J. A. Stiver, and M. Lemmon. Hybrid system modelling and autonomous control systems. In [4].

[3] P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors. *Hybrid Systems II*, Lecture Notes in Computer Science, Vol. 999. Springer-Verlag, 1995.

[4] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, Lecture Notes in Computer Science, Vol. 736. Springer-Verlag, 1993.

[5] J. Lunze. Ein Ansatz zur qualitativen Modellierung und Regelung dynamischer Systeme. *at – Automatisierungstechnik*, 41:451–460, 1993.

[6] J. Lunze. Stabilization of nonlinear systems by qualitative feedback controllers. *International Journal of Control*, 62:109–128, 1995.

[7] NAMUR-Recommendation 31: Anlagensicherung mit Mitteln der Prozeßtechnik. NAMUR (Normenarbeitsgemeinschaft für Meß- und Regelungstechnik in der Chemischen Industrie).

[8] J. Raisch and S. D. O'Young. A DES approach to control of hybrid dynamical systems. In [1], pages 563–574.

[9] J. Raisch and S. D. O'Young. Time-driven supervisory control of hybrid dynamical systems. Proc. 5th International Conference on CONTROL'96, IEE, Exeter, UK. 1996.

[10] J. Raisch and S. D. O'Young. Discrete approximation and supervisory control of continuous systems. Report 96-6, Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart, 1996.

[11] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event systems. *SIAM J. Control and Optimization*, 25:206–230 1987.

[12] J. A. Stiver and P. Antsaklis. Modeling and analysis of hybrid control systems. In *Proc. 31st IEEE Conference on Decision and Control*, 1992.

[13] J. C. Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Transactions on Automatic Control*, 36:259–294, 1991.

GENERATING TIMED DISCRETE MODELS OF CONTINUOUS SYSTEMS

Olaf Stursberg, Stefan Kowalewski and Sebastian Engell
Process Control Group (CT-AST), Department of Chemical Engineering
University of Dortmund, 44221 Dortmund, Germany
email: {olaf | stefan | engell} @ast.chemietechnik.uni-dortmund.de

Abstract. The paper presents a semiquantitative modeling procedure as an approach to systematically derive timed discrete models from continuous models. In particular, it is described how a timed Condition/Event system can be obtained as an approximation of a given DAE-system with a technically motivated partition of the continuous state space. The procedure consists of two main steps: First, the feasible transitions between neighboring discrete states are identified. Then, upper and lower bounds for the residence time in discrete states are determined. We illustrate the proposed method by applying it to a simple mixing tank system. The generated semiquantitative model is transformable into frequently used types of timed discrete systems as shown for the paradigm of timed condition/event-systems.

1 Introduction

This contribution presents an approach to the systematic generation of timed discrete approximations for continuous systems. The need for this abstraction arises in the context of model-based analysis of discretely controlled production units in the chemical process industry. There, the problem is to verify that the behavior of a (piecewise) continuous system will meet certain specifications when a given discrete controller is coupled to the plant. A description of the continuous behavior is obtained by balancing mass, energy and (less often) momentum over the process or a process unit leading to a DAE-System:

$$(1) \qquad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}), \quad 0 = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}),$$

where $\boldsymbol{x}$, $\boldsymbol{u}$ and $\boldsymbol{v}$ represent the vectors of state variables, input quantities and internal algebraic variables. The n-dimensional vector $\boldsymbol{f}$ of nonlinear functions describes the system's dynamics and the algebraic equations are summarized in $\boldsymbol{h}$. For the purpose of this investigation, we assume that all state variables are measurable and that the system is time-invariant and its behavior non-chaotic.

To find a discrete substitution for a system in terms of Eq. (1), two distinct approaches are conceivable: One is the integration of the DAE-system, which implies the problem of requiring (theoretically) an infinite number of calculations, because discrete states introduce starting *regions* (instead of single starting points in continuous systems). Investigations following the integration approach are described e. g. in [2], [6] and [8], where the partition of the state space depends on the continuous dynamics of the system under consideration. A different way to evaluate trajectories $x(t)$ of (1) is to generate a purely qualitative model by approximating the continuous dynamics by untimed discrete models and connecting these to a similar model of the controller. The approximation can be realized either by a more or less intuitive estimation of causal dependencies or by employing the principles of *qualitative simulation* (see e.g. [4]).

The advantage of a completely qualitative discrete model is that appropriate techniques of analysis are available, but it is not sufficient if quantitative information on the dynamical behavior is crucial for verification. In this case, timed discrete models containing information about durations and / or instances of discrete events are necessary. The combination of a discrete state representation with quantitative time information is called a *semiquantitative model* here. We present a procedure which determines feasible transitions in a partitioned state space and computes upper and lower bounds for the residence times in the states. The result is transferable into well-known representations of discrete systems. The main differences in comparison to existing methods are that our procedure is based on a given partition of the state space (not a dynamically generated one) and that it captures all possible transitions (which is necessary for a worst-case reachability analysis). Furthermore, it has a higher level of abstraction than approaches as in [5], since the transitions are only introduced when landmarks are crossed.

2 A semiquantitative modeling procedure

The proposed procedure consists of two main steps, the determination of the possible transitions between adjacent discrete states and the evaluation of trajectories including intervals for their durations. The starting point of our method is Eq. (1), where we omit the algebraic equations and internal algebraic variables for the purpose of this investigation. Hence, the description of the continuous system is given by an ODE-system together with a partition of the continuous state space X, which is assumed to be determined by the requirements of the technical process. In chemical plants for example, it is common practice to measure only whether state variables are below or above certain thresholds and to calculate controller commands depending on this discrete information. This threshold detection corresponds to a partition of the range of each state variable into a finite number of discrete intervals (qualitative states), e. g. "low", "normal" , "high" and "critical" for the temperature in a chemical reactor. Formally, we describe the partitioning by a mapping D: $X \rightarrow \{1, ..., p(1)\} \times ... \times \{1, ..., p(n)\}$ which divides the continuous state space into a finite set of n-dimensional partition elements. D is characterized by an ordered set of landmarks $L_j = \{l_{j,0}, ... l_{j,p(j)}\}$ for each state variable x_j, $j \in \{1, ..., n\}$. A landmark $l_{j,k}$, $k \in \{1, ..., p(j)\}$ corresponds to one of the $p(j)$ thresholds mentioned above. For each of the partition elements, the mapping D generates an index vector $\boldsymbol{d} = (d_1, ..., d_n)$, which specifies in its n components the number of the actual discrete intervals (defined by two consecutive landmarks) for all x_j:

(2) $$d_j = D(x_j) = \begin{cases} k & \text{if } x_j \in [l_{k-1}, l_k[\;\; k \in \{1, ..., p(j)-1\} \\ p(j) & \text{if } x_j \in [l_{p(j)-1}, l_{p(j)}] \end{cases} .$$

If the range of x_j has no upper or lower bound, we write $l_{p(j)} = \infty$ or $l_0 = -\infty$ respectively, with open boundaries for the corresponding intervals. In the following, we use the symbol ${}^{\mathrm{D}}\boldsymbol{x}$ to denote the vector of intervals which refers to the index vector $\boldsymbol{d}$. Hence, D: $X \rightarrow \{1, ..., p(1)\} \times ... \times \{1, ..., p(n)\}$ produces a partitioned state space ${}^{\mathrm{D}}X$ consisting of $\pi = p(1) \cdot ... \cdot p(n)$ elements ${}^{\mathrm{D}}\boldsymbol{x}$ where each of these represents a box-type *cell*:

(3) $${}^{\mathrm{D}}X = \{{}^{\mathrm{D}}x_1, ..., {}^{\mathrm{D}}x_\pi\}.$$

A mapping similar to D is applied to the space U of input variables leading to a partitioned input space ${}^{\mathrm{D}}U$. To formulate a model in the partitioned spaces analogous to the continuous description, we have to use interval arithmetic: The operations in f are replaced by combinations of binary operations $\omega \in \{+, -, \cdot, /\}$ with $A \,\omega\, B = \{x = a \,\omega\, b \mid a \in A, b \in B\}$, $A = [a_1, a_2]$, $B = [b_1, b_2]$, $0 \notin B$ for $\omega = /$ and unary operations $\varrho(x)$, $x \in A$ being defined by: $\varrho(A) = [\min(\varrho(x)), \max(\varrho(x))]$. Denoting the discrete anology of f by φ and the intervals of the derivative by ${}^{\mathrm{D}}\dot{x}$, the discretized ODE-System results in:

(4) $${}^{\mathrm{D}}\dot{\boldsymbol{x}} = \varphi\left({}^{\mathrm{D}}\boldsymbol{x},\ {}^{\mathrm{D}}\boldsymbol{u}\right).$$

Please note that ${}^{\mathrm{D}}\dot{x}$ does not define a partition of the gradient field. In the continuous system, the transient behavior after a single discrete change of u at time t^* is characterized by a trajectory starting in $\boldsymbol{x}(t^*)$ and, if not unstable or periodic, ending in a steady state $\boldsymbol{x}_s$ with $f(x_s) = 0$. The corresponding behavior of the discretized system according to (4) is given in terms of transition sequences between two cells ${}^{\mathrm{D}}\boldsymbol{x}_a$, ${}^{\mathrm{D}}\boldsymbol{x}_b \in {}^{\mathrm{D}}X$ with $\boldsymbol{x}(t^*) \in {}^{\mathrm{D}}\boldsymbol{x}_a$ and $x_s \in {}^{\mathrm{D}}\boldsymbol{x}_b$ $(a, b \in \{1, ..., \pi\})$. To increase the accuracy of the discrete model it is often advantageous to introduce additional discrete states into ${}^{\mathrm{D}}X$ at the coordinates of distinguished steady states (especially if $\boldsymbol{x}_s$ lies in "large cells"). Technically this enlargement is attained by introducing two landmarks $l_{j,k}$, $l_{j,k+1}$ into the set L_j of each state variable, forming an interval of appropriate length with $x_{s,k} \in [l_{j,k}, l_{j,k+1}[$.

To determine the behavior of the discretized system, we first investigate single transitions within the partitioned state space ${}^{\mathrm{D}}X$. We classify transitions in those between two adjacent cells and and those which represent the residence in a cell ${}^{\mathrm{D}}\boldsymbol{x} \in {}^{\mathrm{D}}X$, e. g. for $\boldsymbol{x}_s \in {}^{\mathrm{D}}\boldsymbol{x}$. Two cells ${}^{\mathrm{D}}\boldsymbol{x}_a$, ${}^{\mathrm{D}}\boldsymbol{x}_b$ are said to be *adjacent* in ${}^{\mathrm{D}}X$, if for their corresponding index vectors $\boldsymbol{d}_a$, $\boldsymbol{d}_b$ holds: $\exists_1\, k \in \{1, ..., n\}$: $d_{b,k} = d_{a,k} \pm 1$, $\forall\, l \in \{1, ..., n\} \neq k$: $d_{b,k} = d_{a,k}$. The transitions within ${}^{\mathrm{D}}X$ which correspond to a physically feasible trajectory in the underlying continuous system are defined as so-called *elementary transitions*:

Def. 1: *Elementary transition*

1. For a fixed ${}^{\mathrm{D}}\boldsymbol{u}$ the transition between two adjacent cells ${}^{\mathrm{D}}x_a$, ${}^{\mathrm{D}}x_b \in {}^{\mathrm{D}}X$, $a, b \in \{1, ..., \pi\}$ with corresponding index vectors $\boldsymbol{d}_a$, $\boldsymbol{d}_b$ is an elementary transition $\phi_{d_a \rightarrow d_b}$, if:

   $$\exists\, x \in \partial_{d_a, d_b} : {}^{\mathrm{D}}\dot{x}_k(x) \gtrless 0,\ k = \{1, ..., n\} \text{ with } d_{b,k} = d_{a,k} \pm 1,$$

   where the boundary separating the cells ${}^{\mathrm{D}}\boldsymbol{x}_a$ and ${}^{\mathrm{D}}\boldsymbol{x}_b$ is denoted by ∂_{d_a, d_b}.

2. A transition representing permanent residence in a current cell ${}^{\mathrm{D}}\boldsymbol{x}_a \in {}^{\mathrm{D}}X$ with corresponding index vector $\boldsymbol{d}_a$ is an elementary transition $\phi_{d_a \rightarrow d_a}$ for constant ${}^{\mathrm{D}}\boldsymbol{u}$, if a continuous trajectory defined by Eq. (1) exists that does not leave the cell after reaching it.

The computation of the set Φ of elementary transitions is carried out as shown in Fig. 1 for $n = 2$: A grid is introduced in the partitioned state space with a gridsize chosen suitably small to capture all important changes of the gradient field. For the gridpoints x^g lying on the border of a cell ${}^{\mathrm{D}}\boldsymbol{x}$, the derivative interval ${}^{\mathrm{D}}\dot{x}_k\left(\boldsymbol{x}^g\right)$ in direction orthogonal to the considered cell boundary (k-th component of ${}^{\mathrm{D}}\dot{x}$) is calculated, and the corresponding elementary transition is an element of Φ, if ${}^{\mathrm{D}}\dot{x}_k\left(x^g\right)$ represents a flow out of the cell.

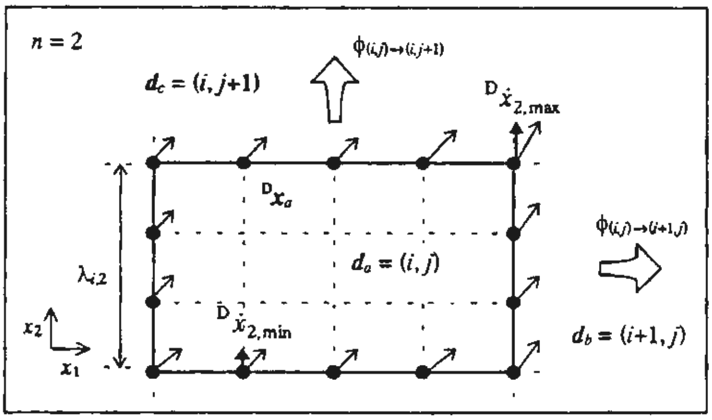


Fig. 1 – Determination of elementary transitions and residence times

Mapping the partition elements to discrete states and the feasible elementary transitions to egdes, an untimed discrete model of the continuous system is obtained. We extend this model to a timed one by assigning the so-called *residence times* to elementary transitions. The residence time represents the time period for which the state stays inside a cell between entering and leaving it. The following definition assigns a so-called *residence time interval* to an elementary transition. This interval gives a conservative estimation of the residence time within the current cell for all continuous trajectories corresponding to the elementary transition, i. e. it is limited by the times at which the transition can happen at the earliest or must occur at the latest:

Def. 2: *Residence time interval*

Given an elementary transition $\phi_{d_a \to d_b}$ between two cells ${}^{\mathrm{D}}x_a, {}^{\mathrm{D}}x_b \in {}^{\mathrm{D}}X$, $a, b \in \{1, \ldots, \pi\}$ with corresponding index vectors d_a, d_b. For $d_a \neq d_b$ the residence time interval $\Delta t_{d_a \to d_b} = [t_{\min}, t_{\max}]$ is bounded by:

$t_{\min} = 0$, $t_{\max} = \left[\dfrac{\lambda_{a,k}}{{}^{\mathrm{D}}\dot{x}_{k,\max}({}^{\mathrm{D}}x_a)}, \dfrac{\lambda_{a,k}}{{}^{\mathrm{D}}\dot{x}_{k,\min}({}^{\mathrm{D}}x_b)}\right]$, and for $d_a = d_b$: $\Delta t_{d_a \to d_a} = [0, \infty[$.

The lower bound of $\Delta t_{d_a \to d_b}$ is zero in the case of an elementary transition to an adjacent cell since the system state may already be infinitesimally close to the boundary. As shown in Fig. 1, the upper bound of the residence time for an elementary transition $\phi_{(i,j)\to(i,j+1)}$ is obtained by determining the minimal value of the flow in direction of the transition (k = 2) and the cell length $\lambda_{i,2}$ as the largest distance to cover (compare [9]). The residence time for the self-loop transition $\phi_{d_a \to d_a}$ is arbitrary unless switching to a new input vector ${}^{\mathrm{D}}u$ enforces an elementary transition $\phi_{d_a \to d_b}$ with $d_a \neq d_b$.

Based on the residence times, we can build a timed transition model. To represent this model, we use the so-called *Elementary Transition Table* (ETT): As shown in Tab. 1, the ETT contains, for each discrete state d_i, the elementary transitions, the enabling input vectors and the upper boundary for the residence times. From the ETT, the system's dynamic behavior is easily derived: For this purpose, we define *discrete trajectories* in the partitioned state space ${}^{\mathrm{D}}X$ and assign *durations* to these trajectories:

Def. 3: *Discrete trajectories*

A discrete trajectory is a sequence: $\phi_{d_1 \to d_m} = \left(\phi_{d_1 \to d_2}, \phi_{d_2 \to d_3}, \ldots, \phi_{d_{m-1} \to d_m}\right)$ of m-1 elementary transitions $\phi_{d_k \to d_{k+1}} \in \Phi$, $k \in \{1, \ldots, m\text{-}1\}$.

The *duration* $\Delta t_{d_1 \to d_m}$ of a discrete trajectory is given by the sum of the residence time intervals of the involved elementary transitions. Using these intervals, it has to be taken into account that their lower boundaries depend on the largest distance to cover within the current cell. In the case of two consecutive elementary transition with the same direction in ${}^{\mathrm{D}}X$, the whole cell length λ in this direction has to be run through. Hence the lower time bound for this trajectory step is given by the lower bound of $t_{\max}$ in Def. 3. In the case of two successive elementary transitions with different directions in ${}^{\mathrm{D}}X$, the smallest possible distance to cover is zero, which is also the value of the lower time bound for the trajectory step, too (see [1] for more details).

d	$\phi_{d_a \to d_b}$	${}^D u$	$t_{\max, d_a \to d_b}$
...			
d_{i-1}	...	...	...
d_i	...	...	...
	$\phi_{(..k,..)\to(..k-1,..)}$	${}^D u_r$	$t_{\max,r}$
		...	...
	$\phi_{(..k,..)\to(..k,..)}$	${}^D u_s$	$[0, \infty[$
		...	...
	$\phi_{(..k,..)\to(..k+1,..)}$	${}^D u_t$	$t_{\max,t}$
		...	...
	...	...	...
d_{i+1}	...	...	...
...			

Table 1 – Scheme of an elementary *T*ransition *T*able (ETT)

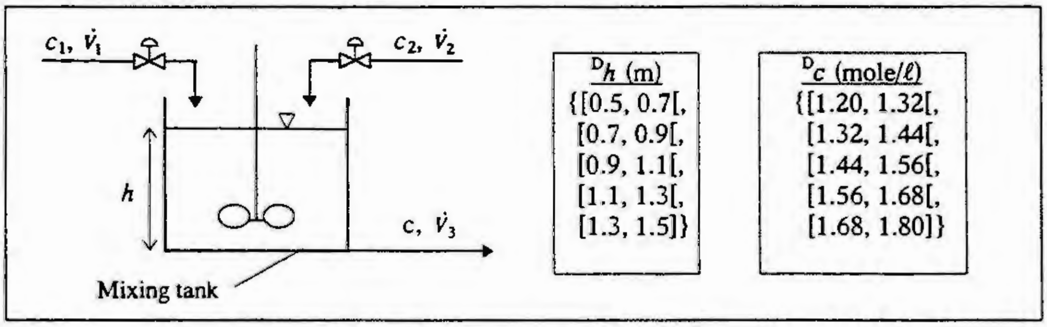
For trajectories with multiple changes of the input vector ${}^D\boldsymbol{u}$, we make the assumption that the input switches only at time points where a transition between two cells occurs (except for elementary transitions $\phi_{d_a \to d_a}$). With this restriction, we reduce the indeterminism in discrete trajectories, since we do not have to consider changing gradient fields as long as the system stays inside the cell. This assumption is reasonable since controller actions (in terms of ${}^D\boldsymbol{u}$) are usually triggered by the crossing of a landmark. The discrete trajectories in ${}^D\boldsymbol{X}$ are inherently non-deterministic, i.e. the calculation of trajectories produces a branching tree of possible paths. To find those with smallest and largest durations, the application of the principles of *dynamic programming* is appropriate. The proposed modeling procedure including the computation of durations of discrete trajectories was implented in MATLAB for one- and two-dimensional systems [1].

3 Application to a simple chemical process

The proposed semiquantitative modeling method was applied to the mixing tank sketched in Fig. 2. This system consists of two controlled inlet streams $\dot{V}_1$, $\dot{V}_2$ with different concentrations c_1, c_2 of a dissolved substance, a free outlet stream $\dot{V}_3$ and a tank hold-up with liquid height h and concentration c. Balancing the total mass and the mass of dissolved substance over the tank, the following differential equations can be derived for the state variables h and c:

$$(5) \qquad \frac{\partial h}{\partial t} = \frac{1}{k_1} \cdot \left(\dot{V}_1 + \dot{V}_2 - k_2 \cdot \sqrt{h}\right), \qquad \frac{\partial c}{\partial t} = \frac{1}{k_1 \cdot h} \cdot \left[\dot{V}_1 \cdot (c_1 - c) + \dot{V}_2 \cdot (c_2 - c)\right],$$

where k_1 and k_2 are geometrical parameters. Choosing these parameters and the inlet concentrations as constant ($k_1 = 1\ \text{m}^2$, $k_2 = 0.02\ \text{m}^{2.5}/\text{s}$, $c_1 = 1$ mole/ℓ, $c_2 = 2$ mole/ℓ), the problem is to find a discrete approximation of the 2-dimensional continuous system given by (5). The ranges of h and c are partitioned into 5 discrete intervals each as listed in Fig. 2.



${}^D h$ (m)	${}^D c$ (mole/ℓ)
{[0.5, 0.7[,	{[1.20, 1.32[,
[0.7, 0.9[,	[1.32, 1.44[,
[0.9, 1.1[,	[1.44, 1.56[,
[1.1, 1.3[,	[1.56, 1.68[,
[1.3, 1.5]}	[1.68, 1.80]}

Fig. 2 – The mixing tank example

For the sake of simplicity, the number of discrete values per input variable is restricted to one: ${}^{D}\boldsymbol{u} = ({}^{D}\dot{V}_1, {}^{D}\dot{V}_2) = (0.008, 0.015)$ m^3/s. Inserting two intervals [1.320, 1.325[m, [1.650, 1.655[mole/ℓ into the partitioned state space to represent the steady state $(h_s, c_s) = (1.322$ m, 1.652 mole/ℓ), ${}^{D}X$ consists of 49 discrete states. Each of them is subdiscretized by a rough grid with a meshsize of $\lambda_{i,k}/2$ in direction of both coordinates k. From this, a set Φ consisting of 85 elementary transitions is evaluated. Some of them are listed in Tab. 2.a, where the index vector $\boldsymbol{d}$ corresponds to a discretized state vector ${}^{D}\boldsymbol{x} = ({}^{D}h, {}^{D}c)$ and where the derivative values respectively the input vector are omitted.

(a)

d	$\phi_{d_a \to d_b}$	$t_{\max,d_a \to d_b}$ [s]
(1, 1)	$\phi_{(1,1)\to(2,1)}$	[22.6, 31.9]
	$\phi_{(1,1)\to(1,2)}$	[5.8, 11.0]
(2, 1)	$\phi_{(2,1)\to(3,1)}$	[31.9, 49.7]
	$\phi_{(2,1)\to(2,2)}$	[8.1, 14.1]
(3, 1)	$\phi_{(3,1)\to(4,1)}$	[49.7, 98.8]
...	$\phi_{(3,1)\to(3,2)}$	[10.4, 17.3]
...		
(6, 5)	$\phi_{(6,5)\to(6,5)}$	[0, ∞[
...		

(b)

$\phi_{d_a \to d_b}$	Number of elementary transitions	Number of paths	$t_{\max,d_a \to d_b}$ [s]
$\phi_{(1,1)\to(3,2)}$	3	3	[31.9, 98.9]
$\phi_{(1,1)\to(4,3)}$	5	10	[89.4, 232.8]
$\phi_{(1,1)\to(5,4)}$	7	35	[200.6, 1326.0]

Table 2 – Excerpt of the ETT (a) and some example trajectories (b)

From Tab. 2.a, we gain information about possible behaviors of the mixing tank including statements referring to durations: For example, the transition (for the chosen ${}^{D}\boldsymbol{u}$) from an initially full tank ($h \in [1.32, 1.5]$ m) with a mixture of low concentration ($c \in [1.2, 1.32]$ mole/ℓ) to a state of low level ($h \in [0.7, 0.9]$ m) and high concentration ($c \in [1.68, 1.80]$ mole/ℓ) is impossible because the trajectory $\phi_{(7,1)\to(2,7)}$ is not feasible. Fig. 3 gives an overview of all elementary transitions. In Tab. 2.b three examples of discrete trajectories including the number of involved elementary transitions, the number of possible paths between start and end state, and the corresponding durations are shown. The figures reveal that with an increasing number of elementary transitions the search space grows exponentially and larger uncertainties of durations occur. Some modifications of the modeling procedure reducing these disadvantages are described in [1].

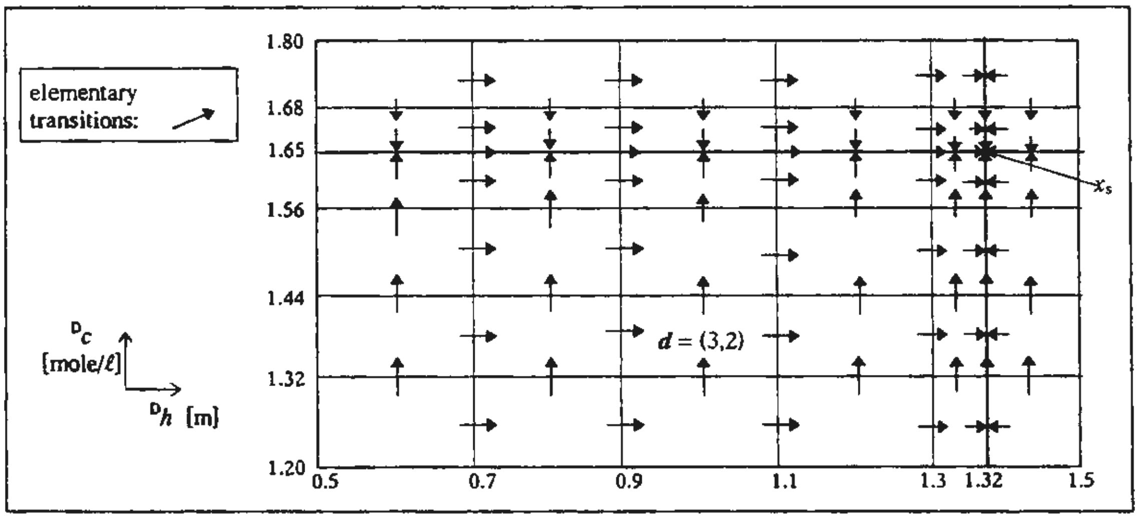


Fig. 3 – Elementary transitions within the discrete model of the mixing tank

4 Transformation into Timed Condition/Event-Systems

In order to take advantage of available analysis techniques and tools, the ETT can be transformed into existing modeling paradigms. In principle, any discrete formalism which offers the possibility to include quantitative timing information, e.g. timed automata or timed Petri nets, can be chosen for this purpose [1]. We use *Timed Condition/Event-systems (TC/E-systems)* as the modeling paradigm. *Condition/Event (C/E)-systems* (according to [7]) were introduced to model interconnected discrete event systems in a modular, block diagram and signal flow oriented fashion. They are based on two classes of continuous time signals which can both be input and output signals of one system: *condition signals* and *event signals*.

A condition signal is a symbolic-valued, piecewise constant function of time; its values correspond to discrete states. An event signal is a symbolic-valued, pointwise nonzero function of time and carries information about currently occuring state transitions. A C/E-system has a *condition input signal* $u(t)$, an *event input signal* $v(t)$, a *condition output signal* $y(t)$ and an *event output signal* $z(t)$. TC/E-systems are an extension of C/E-systems, in which C/E-timers are introduced as a new class of C/E-systems to represent timing information. A C/E-timer can be regarded as an alarm clock over real time t with a certain threshold time T. The clock is reset and started by the input event "$t := 0$" and reaching a threshold time T is indicated by sending out an event "$t = T$". The condition outputs "$0< t <T$" and "$t \geq T$" determine whether the threshold is not yet or was already reached [3].

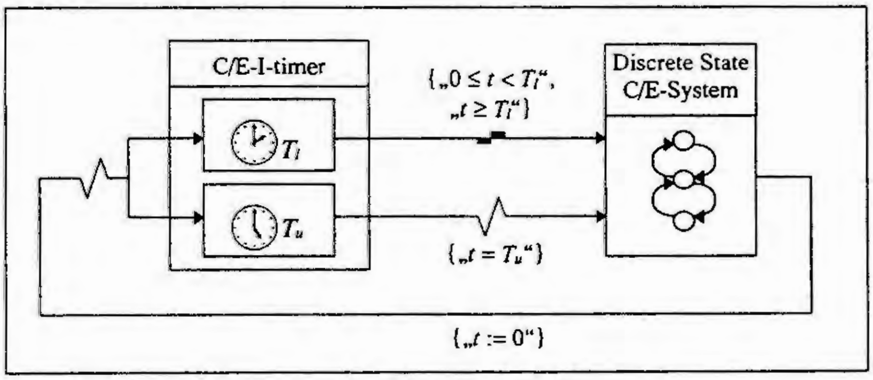

C/E-I-timer
T_l
T_u
{„$0 \leq t < T_l$“, „$t \geq T_l$“}
{„$t = T_u$“}
Discrete State C/E-System
{„$t := 0$“}

Fig. 4 – C/E-I-timer

The timing information in a ETT consists of lower and upper bounds for the instances at which a transition takes place. Thus, a lower threshold T_l gives the time at which the transition can happen at the earliest and a upper threshold T_u gives the time at which the transition has to take place eventually. Two C/E-timers, one for T_l and one for T_u, are needed to realize this in a TC/E-system (see Fig. 4). Both timers, corresponding to one row of the ETT, are combined in one system, which is called a *C/E-Interval-timer* (C/E-I-timer).

The transformation of a ETT into a TC/E-system consists of two steps: First, the discrete state part of the TC/E-system is built by copying the discrete dynamics represented by the discrete states in ${}^{D}X$ and the elementary transitions. Second, for each row of the ETT one C/E-I-timer is introduced with T_l and T_u representing the boundaries of the timing interval. The available reachability analysis procedure for TC/E-systems [3] can be applied to analyze the semiquantative model.

5 Conclusions

We have proposed a modeling technique which generates discrete approximations of continuous systems given in terms of ODE-systems. The aims were on one hand side to capture all possible behaviors of the continuous system with acceptable computational effort, i. e. without integrating the ODE-system explicitly. One the other hand, we intended to include quantitative information into the discrete model to preserve timing information. Thus, a *semiquantitative* modeling technique was chosen. It produces a non-deterministic model, represented by a table of elementary transitions which contains discrete states, feasible transitions and residence times as sufficient information to describe the system's dynamic behavior on a discrete level.

As in all methods abstracting from exact quantitative functions, spurious trajectories, i. e. trajectories with no correspondence in the continuous system, can be produced. To overcome this disadvantage, we currently investigate two approaches: The first tries to reproduce continuous trajectories within the discrete cells as sequences of gridpoints, leading to a modified ETT where pairs of *entering* and *leaving* elementary transitions are listed. The second approach introduces a subpartition into the cell state space (via integrating (1)), which divides a cell into sections of homogenous flow direction.

Acknowledgement

The work of the first author was supported by the German Research Council (DFG) within the graduate school *Modeling and Model-based Development of Complex Technical Systems*.

References

1. Hoffmann, I., Kowalewski, S., Preußig, J. and Stursberg, O., Towards Systematic Derivation of Timed and Hybrid Automata from Continuous Models. Presented at Hybrid Systems'96, Ithaca, New York, Oct. 1996.
2. Kokar, M. M., On Consistent Symbolic Representations of General Dynamic Systems. IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 8, 1995.
3. Kowalewski, S. and Preußig, J., Timed Condition/Event Systems: A Framework or Modular Discrete Models of Chemical Plants and Verification of Their Real-Time Discrete Control. In: Tools and Algorithms for the Construction and Analysis of Systems (LNCS 1055), 1996.
4. Kuipers, B., Qualitative Simulation. Artificial Intelligence 29 (1986), 289-338.
5. Lunze, J., Qualitative Modelling of Linear Dynamical Systems with Quantized State Measurements. Automatica, Vol. 30 (3), 1994, 417-431.
6. Niinomi, T., Krogh, B. H., Cury, J. E. R., Synthesis of Supervisory Controllers for Hybrid Systems based on Approximating Automata, Conference on Decision and Control, New Orleans, 1995.
7. Sreenivas, R. S., Krogh, B. H., On Condition/ Event Systems with Discrete State Realizations. In: Discrete Event Dynamic Systems - Theory and Applications I, 1991, 209-236.
8. Stiver, J. A., Antsaklis, P. J., State space Partitioning for Hybrid Control Systems. Proc. of the American Control Conference, San Francisco, California, June 1993, 2303-2304.
9. Tetrault, M., Marcos, B., Lapointe, J., Temporal Duration Reasoning in Qualitative Simulation. Artificial Intelligence in Engineering 7 (1992), 185-197.

HYBRID FLOW NETS FOR BATCH PROCESS MODELLING AND SIMULATION

Jean-Marie Flaus
Laboratoire d'Automatique de Grenoble, UMR
ENSIEG, BP 46
F-38402 St Martin d'Hères, FRANCE
Email : flaus@lag.grenet.fr

Abstract: This paper presents a new methodology for modelling and simulation of batch processes called hybrid flow nets, which can be seen as an hybrid and non linear continuous extension of Petri nets. This modelling methodology is illustrated on an example of biotechnological batch plant.

Introduction

Nowadays, batch processes are becoming more and more important in industries dealing with material transformation such as chemical, food biotechnological or pharmaceutical industries. In this production mode, material is processed in discrete quantities following a recipe in a sequentially manner, rather than as a continuous flow transformed according to a fixed procedure. So operating mode description, which is usually not part of the model for continuous processes becomes very important in the representation of a batch process. For some purposes, the model of a batch process boils down to a discrete event model [6] which only describes the equipment usage and the recipe. In this work, we aim at building a global model of the batch plant which includes a discrete part and a continuous part and we propose an hybrid modelling approach.

Various framework have been recently proposed in the litterature in order to deal with such so-called hybrid systems. These range from hybrid automata [1] and rather complex structures based on an extension of continuous model [3] to mixed models based on a discrete part and a continuous with a suitable definition of interface [5]. An interesting approach is hybrid Petri nets [2] which have been built from an extension of Petri nets and are very attractive for modelling linear hybrid processes and have an inherent quality in representing logic in an intuitive and visual way. This approach has been extended in order to represent a larger class of non linear systems to lead to what we have called Hybrid Flow Nets.

This paper is organized as follows: firstly, we present hybrid flow nets, then, we show how this modelling framework can be used for modelling a biotechnlogical plant.

Hybrid Flow Nets

Hybrid Flow Nets [4] can be seen as a non linear extension of hybrid Petri nets. This structure is made of a continuous part, modelled by what we call continuous flow net, and a discrete part, described by a Petri net. These two parts are interacting together.

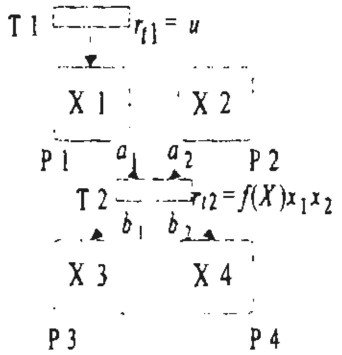


Figure 1: Basic Continuous Flow Net

A continuous flow net (CFN) is defined by a graph structure similar to the one of Petri Nets, where vertices are places and transitions. At each place is associated a real positive variable, called the marking of the place. To each transition is associated a flow of the quantities in the input places of this transition. This flow is defined as such that it is zero if any input place is empty. More formally, a **Continuous Flow Net** (CFN) is defined by a n-uplet:

$C = (P_c, T_c, I_c, O_c, \Phi, U_f, I_f, X_0)$ where :

P_c is a set of places represented graphically by a rounded square. To each place is associated a real number, positive, which is called *marking* or *value* of the place, and denoted x. The set of values makes a vector X where $X(i)$ is the marking of the i-th place. X_0 is the initial marking.

T_c is a set of n_t transitions (or gates), represented graphically by a specific rectangle (figure 1);

$I_c : P \times T \rightarrow \{0, 1\}$ is the *input function* that specifies the arcs directed from places to transitions;

$O_c : P \times T \rightarrow \{0, 1\}$ is the *output function* that specifies the arcs directed from transitions to places;

$\Phi : T \rightarrow F_R$ associates to each transition a function $f_t(X)$ defined from $\Re^m$ to $\Re$, that is bounded $0 \leq f_t(X) \leq F_{\max}$.The flow through a transition t is proportional to the value of the input places of this transition and to the value of the input flows connected to the input of this transition :

$$r_t = f_t(X) \prod_{I_c(i,t)=1} X(i) \quad \text{with } 0 \leq f_t(X) \leq F_{\max}$$

In order to model continuous and discrete aspects of a system, we introduce what we call hybrid flow net. This modelling tool is made of a continuous flow net interacting with a Petri net according to a control interaction, that is to say the Petri net controls the CFN and vice versa . As we are going to see it, the overall philosophy of Petri nets is preserved again: the validation of transition implies that all the input places are not empty and the evolution rule is similar.

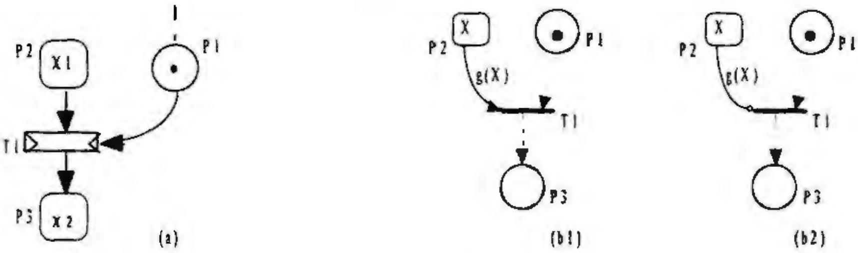


Figure 2: Interaction from continuous part to discrete part (b) and vice versa (a)

The interface of the discrete part to the continuous part is made through the control of a continuous transition by a discrete place. The flow rate of the transition is then equal to :

$$r_h(t_j) = m(p_i).r_t(t_i)$$

where $m(p_i)$ is the marking of the discrete place used for the control and $r_t(t_i)$ is the continuous flow rate defined above.

The influence of continuous part to the discrete one is made via some conditions on the continuous variable that are used to enabled a discrete transition. For example (figure 2), the transition t_j is enabled if $m(p_i) > 0$ and $X(p_d) > g(X)$. Then , the firing of the transition t_j leads to a new marking obtained in the same way as for a classical Petri net.

More formally, an **Hybrid Flow Net** (HFN) is defined by a n-uplet $H = (C, Z, \Psi_{c/d}, \Psi'_{c/d}, \Psi_{d/c})$ where

C is a continuous flow net as defined above;

Z is a Petri net;

$\Psi_{d/c} : P_d \times T_c \rightarrow \{0, 1\}$ specifies the continuous transition T_j controlled by a discrete place P_i.

$\Psi_{c/d} : P_c \times T_d$ specifies the discrete transitions T_j controlled by a place . If $\Psi_{c/d}(i, j) \neq 0$, the transition is enabled iff $X(i) \geq \Psi(i, j)$, and if it is enabled by the rest of the net. If $\Psi_{c/d}(i, j) = 0$, there is no arc between the place and the transition Tj.

$\Psi'_{c/d} : P_c \times T_d$ specifies the discrete transitions T_j inhibited by a place . If $\Psi'_{c/d}(i,j) \neq 0$, the transition is enabled iff $X(i) \leq \Psi(i,j)$, and if it is enabled by the rest of the net. If $\Psi'_{c/d}(i,j) = 0$, there is no arc between the place and the transition Tj.

In the basic definition, the flow through a transition depends on the values of the input places of the transition and on a bounded function $f(X)$. It may be interesting to specify a little bit more the form of this function $f(X)$. To do it, we have introduced the notion of controlled continuous flow net. We will consider two cases:

(a) the flow modulation by a variable q (figure 3.a). In this case, the flow rate through a transition becomes :

$$r_{t_j} = q.f(X) \prod_{I_c(i,j)=1} X(i) \text{ with } 0 \leq f(X) \leq F_{\max}$$

(b) the inhibition of the flow by a variable q (figure 3.b). In this case, the flow rate through a transition becomes

$$r_{t_j} = \frac{1}{q}.f(X) \prod_{I_c(i,j)=1} X(i) \text{ with } 0 \leq f(X) \leq F_{\max}$$

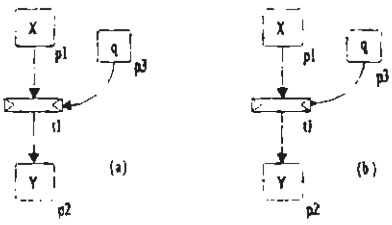


Figure 3: Controlled Flow Net

The graphical representation of controlled transition is shown on figure 3. The arcs from control variables to transitions are directed to the small sides of the transition. Modulation arcs are ended by an arrow while inhibition arcs are ended by a point. It must be noticed that these variables control the flow but are not affected by it. Only the variables with arcs connected to the large side of the transition are flowing out.

A simulation tool for Hybrid Flow Nets, that we cannot describe in details because of space limitation, has been developed in the Matlab environnement.

Application for batch process modelling

Hybrid flow net are especially well suited for batch processes modelling. Petri nets can be used for recipes modelling while several continuous flow nets are used for process modelling, one for each extensive variable that are of interest such as material, enthalpy or concentration. To show the modelling capability of this tool, we are going to consider an academic example of batch process made up of four tanks. The tank 2 holds the substrate, the tank 1 holds the biomass inoculum, the third is a bioreactor and the last one is a storage tank (fig 4). The recipe is the following : first fill the bioreactor with a given quantity of inoculum (sensor H1), then fill the given bioreactor with a given quantity of substrate (sensor H2). Then wait for the concentration of substrate is less than 0.01 g/l while heating with an on/off heater. At the end, transfer the contents of the bioreactor in the storage tank.

To build an hybrid model of this plant, we are going to write the mass, energy and components balances equations. We will denote respectively V_1, V_2, V_3, V_4 the volume of liquid in the tanks 1,2,3 and 4. The flow rate of the valves u_1, u_2, u_3 and u_4 are assumed to be equal to $F_1 = \alpha.V_1\delta(u_1)$; $F_2 = \alpha.V_2\delta(u_2)$; $F_3 = \beta.V_3/(K+V_3)\delta(u_3)$; $F_4 = \alpha.V_4\delta(u_4)$ where α, β and K are three parameters and $\delta(u_i) = 1$ the valve is open and 0 if the valve is closed. The enthalpy of each tank $H = \rho cT$, where ρ is the density of the liquid and c the heat capacity, is denoted H_i. The concentration of biomass and substrate in tank i is denoted respectively X_i and S_i.

The mass balances equations writes :

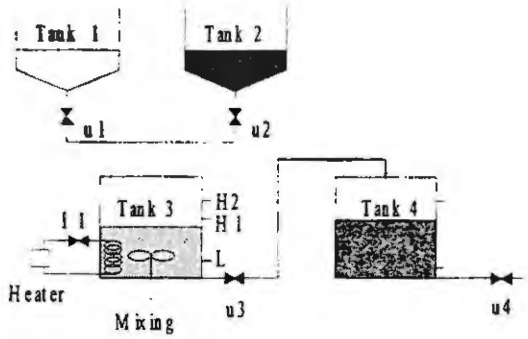


Figure 4: Example of biotechnological batch plant

$$\begin{aligned}\frac{dV_1}{dt} &= -F_1 = -\alpha\delta(u_1).V_1\\ \frac{dV_2}{dt} &= -F_2 = -\alpha\delta(u_2).V_2\\ \frac{dV_3}{dt} &= +F_1 + F_2 - F_3 = \alpha\delta(u_1).V_1 + \alpha\delta(u_2).V_2 - \beta/(K+V_3)\delta(u_3).V_3\\ \frac{dV_4}{dt} &= +F_3 - F_4 = \beta/(K+V_3)\delta(u_3).V_3 - \alpha\delta(u_4).V_4\end{aligned}$$

From these equations, a continuous flow net can be drawn, corresponding to the places p_1, p_2, p_3, p_4 on the figure 5. The second step is to write the energy balance. For a given tank, with a inlet stream of flow rate F_{in} and temperature T_{in} and an outlet stream of flow rate F_{out} and temperature T_{out},the energy equation writes:

$$\frac{dH}{dt} = F_{in}\,\rho c T_{in} - F_{out}\,\rho c T_{out} + Q$$

where Q is the amount of heat supplied by the heater per unit of time, and that is given by $Q = UA(T_c - T)$, with U :overall heat transfer coefficient, A :heat transfer area and T_c the temperature of the heating water. This energy balance equation can be rewritten as follows:

$$\frac{dH}{dt} = F_{in}\,\rho c T_{in} - F_{out}\,\frac{H}{V} + UAT_c - \frac{UA}{\rho c}\frac{H}{V}$$

In the case of the considered example, only the tank 3 is heated. So the equations for each tank are :

$$\begin{aligned}\frac{dH_1}{dt} &= -F_1\,\frac{H_1}{V_1} = -\alpha\delta(u_1)H_1\\ \frac{dH_2}{dt} &= -F_2\,\frac{H_2}{V_2} = -\alpha\delta(u_2)H_2\\ \frac{dH_3}{dt} &= F_1\,\frac{H_1}{V_1} + F_2\,\frac{H_2}{V_2} - F_3\,\frac{H_3}{V_3} + UAT_c\delta(i_1) - \frac{UA}{\rho c}\frac{H_3}{V_3}\delta(i_1)\\ &= F_1\,\frac{H_1}{V_1} + F_2\,\frac{H_2}{V_2} - \frac{\beta}{K+V_3}H_3 + UAT_c\delta(i_1) - \frac{UA}{\rho c}\frac{H_3}{V_3}\delta(i_1)\\ \frac{dH_4}{dt} &= F_{in}\,\rho c T_{in} - F_{out}\,\frac{H}{V} + UAT_c - \frac{UA}{\rho c}\frac{H}{V}\end{aligned}$$

where $\delta(i_1) = 1$ if the heating flow is open. This set of equations can be represented by a flow net as shown in figure 5 where places $p_{11}, p_{12}, p_{13}, p_{14}$ are used to describe the energy variables. The energy flow is the same as the volume flow except for the addition/removal of heat to the tank 3 which is described by two specific transitions on the flow net.

Now, to illustrate that an hybrid flow net can also be used for modelling a reaction, which can be seen as the transfer of a specie into another, we are going to consider the bioreaction occurring in the tank 3. First, let us recall that a bioreaction producing biomass from substrate has a growth that is often modelled with a Monod law under usual assumptions. The biomass and substrate accumulation in the bioreactor is governed by the following set of equations:

$$\begin{aligned}
\frac{d(VX)}{dt} &= \mu(S)V.X + F_{in}.X_{in} - F_{out}X \\
\frac{d(VS)}{dt} &= -\frac{\mu(S)}{Y_{xs}}V.X + F_{in}.S_{in} - F_{out}S \\
\frac{dV}{dt} &= F_{in} - F_{out} \\
\mu(S) &= \frac{\mu_{\max}}{K_s + S}S = \mu^*(S).S \text{ (Monod law)}
\end{aligned}$$

where X_{in} and S_{in} are the biomass and substrate concentration in the inlet stream, X and S the same concentration in the bioreactor and $\mu_{\max}, Y_{xs}, K_s$ are some parameters. In our example, we are going to write balances equations for the total quantity of substrate and biomass in each tank (and not for the concentration). We will denote $Xt_i = X_iV_i$ and $St_i = X_iV_i$. As there is only biomass in the tank 1, we have $St_1 = 0$ and as there is only substrate in the tank 2, we have $Xt_2 = 0$. The substrate and biomass evolution in the bioreactor are governed by:

$$\begin{aligned}
\frac{d(V_3X_3)}{dt} &= \mu^*(S_3)S_3V_3X_3 + F_1.X_1 - F_3X_3 \\
\frac{d(V_3S_3)}{dt} &= -\frac{\mu^*(S_3)}{Y_{xs}}S_3V_3X_3 + F_2.S_2 - F_3S_3
\end{aligned}$$

The evolution of X_t and S_t in each tank are given by the following set of equations:

- Tank 1:

$$\frac{dXt_1}{dt} = -F_1X_1 = -\alpha V_1X_1\delta(u_1) = -\alpha\delta(u_1)Xt_1$$

- Tank 2 :

$$\frac{dSt_2}{dt} = -F_2S_2 = -\alpha V_2S_2\delta(u_2) = -\alpha\delta(u_2)St_2$$

- Tank 3 :

$$\begin{aligned}
\frac{dSt_3}{dt} &= +F_2.S_2 - F_3S_3 - \frac{\mu^*}{Y_{xs}}X_3St_3 \\
&= \alpha\delta(u_2)St_2 - \frac{\beta}{K + V_3}\delta(u_3)St_3 - \frac{\mu^*}{Y_{xs}}X_3St_3 \\
\frac{dXt_3}{dt} &= +F_1.X_1 + \mu^*S_3Xt_3 - F_3X_3 \\
&= \alpha\delta(u_1)Xt_1 + \mu^*X_3St_3 - \frac{\beta}{K + V_3}\delta(u_3)Xt_3
\end{aligned}$$

- Tank 4 :

$$\begin{aligned}
\frac{dSt_4}{dt} &= \frac{\beta}{K + V_3}\delta(u_3)St_3 - F_4S_4 \\
&= \frac{\beta}{K + V_3}\delta(u_3)St_3 - \alpha\delta(u_4)St_4 \\
\frac{dXt_4}{dt} &= \frac{\beta}{K + V_3}\delta(u_3)Xt_3 - F_4X_4
\end{aligned}$$

$$= \frac{\beta}{K + V_3}\delta(u_3)Xt_3 - \alpha\delta(u_4)Xt_4$$

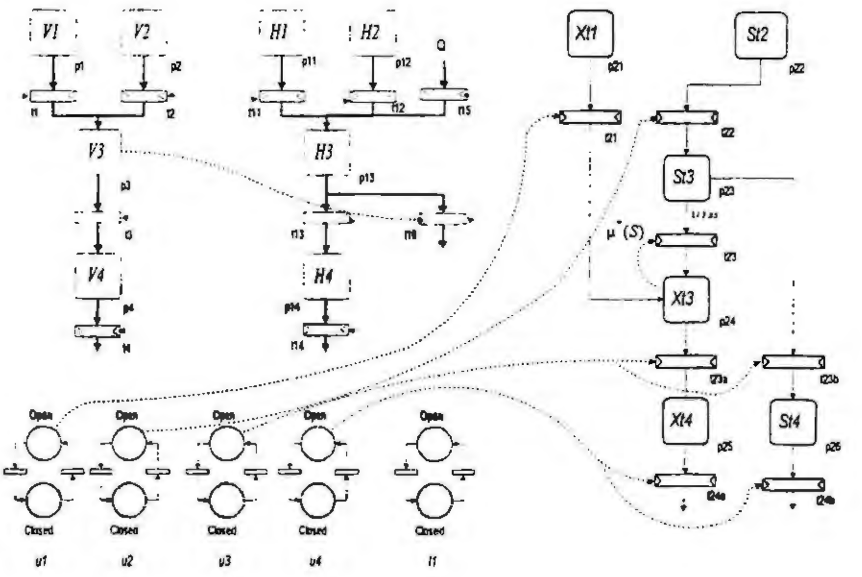


Figure 5: Hybrid Flow Net for the example batch plant

To be complete, the last step is to write the Petri net associated to the recipe, which is straightforward. We will not developped this here because of space limitations.

Conclusion

In this paper, we have presented an application to batch process modelling of a new modelling tool, called hybrid flow net. This tool is able to describe in a graphical way, continuous and discrete flow interacting together. The discrete part is a Petri net while the continuous part is called continuous flow net whose dynamic evolution has be defined so that to be similar to the one of Petri net, with a continuous enabling rule and a continuous firing rule. Hybrid flow nets are well suited for the modelling of batch processes, which can be seen as a set of flows interacting together.

References

[1] R. Alur, C. Courcoubetisa, N. Halwachs, T. Henziger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoritical computer science*, 138:3–34, 1995.

[2] J. L. Bail, H. Alla, and R. David. Hybrid petri nets. In *European Control Conference, Grenoble*, pages 1472–1477, July 2-5 91.

[3] M. S. Branicky, V. Borkar, and S. Mitter. A unified framework for hybrid control. In *11th International Conference on Analysis and Optimization of Systems*. Lecture Notes in Control and Information Sciences, Springer Verlag, June 94.

[4] J. Flaus. Hybrid flow networks for hybrid process modelling. Technical report, Internal Report, LAG, 96.

[5] A. Pnueli and J. Sifakis. Hybrid systems: special issue. *Theoritical Computer science*, 138, 95.

[6] A. Sanchez. *Formal Specification and Synthesis of Procedural Controllers for Process Systems*. Springer Verlag, 1996.

A FORMAL METHODOLOGY FOR REPRESENTING QUALITATIVE EQUATIONS WITH PETRI NETS

Alessandra Fanni, Alessandro Giua
DIEE: Dip. di Ingegneria Elettrica ed Elettronica — Università di Cagliari
P.zza d'Armi - 09123 CAGLIARI, Italy — giua@diee.unica.it — Fax: +39 (70) 675-5900

Abstract. The paper describes a formal procedure to construct a Petri net model corresponding to a given set of qualitative equations. The approach can be used to study both autonomous systems and systems with forcing inputs. The dynamic behavior of the system can be studied as sequences of reachable markings of the net and can be computed with standard Petri net execution techniques. An electrical circuit is considered as applicative example.

1. Introduction

Qualitative simulation is a well know technique for studying continuous or discrete time systems [1, 3, 4].

The major drawback of qualitative simulation is its fundamental ambiguity: given a qualitative model of a system and a set of qualitative inputs, more than one qualitative behavior can generally be found that follows from those initial data. This ambiguity partially depends on the choice of the quantity space used to represent the qualitive value of the variables. By refining the partitions of the real axis that define these quantity spaces it is often possible to mitigate this ambiguity.

Another disadvantage of qualitative analysis derives from the fact that we lack effective simple mathematical tools for carrying out the simulation. Solving a set of qualitative constraints requires ingenuity and the use of heuristics.

We propose a simple way of avoiding this problem. We note that a qualitative system, with its discrete quantity state space can also be seen as a discrete event system (DES) [8]. Thus its behavior may be described by any of the models used to represent a DES. This work focuses on the use of Petri nets models [5].

Petri nets have been used in qualitative simulation by Okuda and Ushio [6, 7]. These authors noted that each *place* of a net may be associated to a state of a variable while the firing of each *transition* corresponds to crossing a landmark.

In this paper we extend and formalize the approach presented in [2], assuming that each *marking* (not place) of a net may be associated to a state of a variable and that a transition may represent more than one landmark crossing. Thus when we consider variables with increasing quantity spaces, we need not modify the structure of the net, but just to change the number of tokens it contains.

The paper presents general algorithms for deriving a Petri net model corresponding to a given set of qualitative equations in a given quantity space. As a simple applicative example we consider an electrical circuit.

2. Generalities

Qualitative models

Qualitative modeling exploits relationships that express qualitative connections between the variables of a physical system.

A qualitative model uses *qualitative variables* with an associated *quantity space* defined as a set of disjoint intervals (possibly of zero length, in which case they reduce to points) that cover the real straight line. The qualitative value of a variable x is denoted $[x]$.

The quantity space usually employed is that comprising the intervals $(-\infty, -\epsilon)$, $(-\epsilon, \epsilon)$, (ϵ, ∞); thus one writes $[x] = -$, $[x] = 0$, and $[x] = +$ to denote the interval to which the value of x belongs. The laws that govern the system behavior are expressed as equations between these qualitative variables. One problem with this approach is the essential ambiguity of the qualitative sum, i.e., $[x] + [y]$ may take any value in $\{-, 0, +\}$ if $[x] = +$ and $[y] = -$, or $[x] = -$ and $[y] = +$.

The qualitative sum ambiguity can be avoided using a finer partition of the real axis that also gives a better description of the system behavior. As an example, we will often use the quantity space $\{-n, -n+1, \dots, -1, 0, 1, \dots, n-1, n\}$. In this case the qualitative sum $[x] + [y]$ follows the same rules of algebraic addition.

Petri nets as qualitative models

A *place/transition net* [5] is a structure $N = (P, T, Pre, Post)$, where P is a set of *places* represented by circles; T is a set of *transitions* represented by bars; $Pre : P \times T \rightarrow \mathbb{N}$ is the *pre-incidence function* that specifies the arcs directed from places to transitions; $Post : P \times T \rightarrow \mathbb{N}$ is the *post-incidence function* that specifies the arcs directed from transitions to places. A *marking* is a vector $M : P \rightarrow \mathbb{N}$ that assigns to each place of a P/T net a non-negative integer number of tokens, represented by black

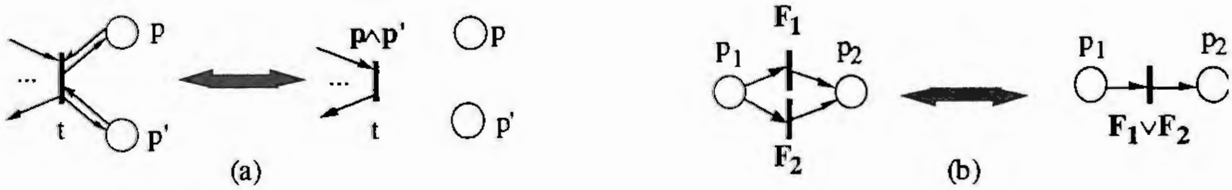


Figure 1: Representation of self-loops (a), and of parallel transitions (b).

dots. A transition $t \in T$ is *enabled* at a marking M iff $M \geq Pre(\cdot, t)$. If t is enabled at M, then t may fire yielding a new marking M' with $M' = M + Post(\cdot, t) - Pre(\cdot, t)$. See [5] for a more comprehensive definition of Petri nets.

We will use Petri nets as qualitative models of physical systems with the following assumptions.

The qualitative value of each variable is associated with the marking of a subset of places in the Petri net. Thus we have the correspondence between qualitative states and markings. The initial state of the system will determine the initial marking M_0 of the net.

The firing of a transition will represent the change of a qualitative variable from one qualitative value to another. Note that a single transition may be enabled by several different markings, thus the same transition may represent different qualitative changes.

The set of all possible states reachable from the initial state will be given by the reachability set $R(N, M_0)$ of the net. The sequence of all possible behaviors is given by all sequences of transitions σ firable from the initial marking.

The change of value of a qualitative variable, say x, is often depending on the value of another one, say v. Thus, in the Petri net model a transition that changes the marking of the places associated to x may depend on the marking of the places associated to v. The influence of v over x may be represented by self-loops, i.e., cycles in the net graph containing only one place and one transition.

Consider a transition t self-looped with places p and p' as in Figure 1.(a). The firing of t is only possible if there is at least a token in p and at least a token in p'. The firing of t, however, does not change the number of tokens in p and p'. To avoid representing the two arcs between p and t and tbe two arcs between p' and t we simply assign to transition t the label $\mathbf{p} \wedge \mathbf{p}'$. This can also be generalized to a self-loop of n arcs using the label $\{\mathbf{p} = \mathbf{n}\} \wedge \{\mathbf{p}' = \mathbf{n}\}$.

We will often need to give a compact representation of a structure in which there are parallel transitions with different labels $\mathbf{F_i}$, each of which may be the $\wedge$ of single labels as discussed above. A simpler representation of this will be a single transition with label $\mathbf{L} : \vee_i \mathbf{F_i}$, as shown in Figure 1.(b).

3. Petri net modeling of qualitative systems

In this section we discuss how from a qualitative model of a continuous time system it is possible to derive a discrete event model using Petri net structures.

Let us consider a continuous time system described by the following set of state equations:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{1}$$

where $\mathbf{x}$ is a vector with p components, $\mathbf{u}$ is a vector with q components, $A = \{a_{i,j}\}$ is a $p \times p$ matrix, and $B = \{b_{i,k}\}$ is a $p \times q$ matrix.

The qualitative model corresponding to (1) is given by the following set of qualitative equations ($i = 1, \cdots, p$):

$$[\dot{x_i}] = \sum_{j=1}^{p} [a_{i,j}][x_j] + \sum_{k=1}^{q} [b_{i,k}][u_k] \tag{2}$$

We will consider two different models for state variables in the quantity spaces $\{-, 0, +\}$ and $\{-n, \cdots, 0, \cdots, +n\}$. For each of these two cases, we will give general construction algorithms to derive a Petri net model representing a given set of qualitative equations.

Once a model has been constructed, the behavior of the net can be studied with various techniques pertaining to Petri nets. In particular, reachability analysis may be used to study the evolution of the system with standard Petri net simulators [9]. Examples and discussions are presented in Section 4.

Model with quantity space $\{-, 0, +\}$

The following algorithm can be used to construct a Petri net model when the quantity space of the variables is $\{-, 0, +\}$.

Algorithm 1 Consider the qualitative equations (2).

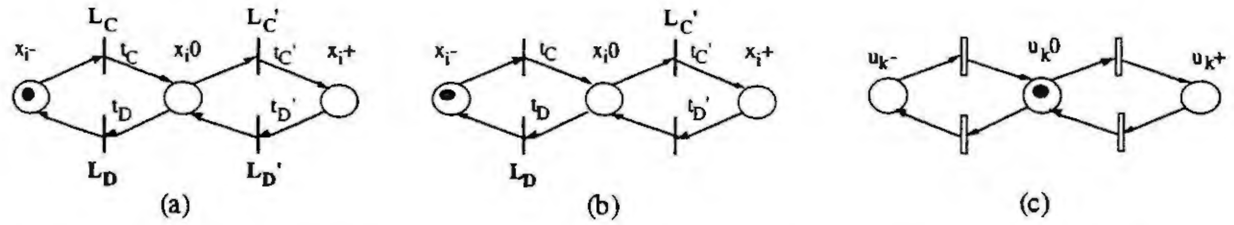


Figure 2: Petri net model with quantity space $\{-,0,+\}$. (a) Subnet for x_i with $[a_{i,i}] \in \{0,+\}$. (b) Subnet for x_i with $[a_{i,i}] = -$. (c) Subnet for input u_k.

1. Associate to each state variable x_i a Petri net with three places $\mathrm{x_i}-$, $\mathrm{x_i0}$ and $\mathrm{x_i}+$, as in Figure 2.(a). Here a token in place $\mathrm{x_i}-$ means that $[x_i] = -$, a token in place $\mathrm{x_i0}$ means that $[x_i] = 0$, and so on. Thus, we may write $[x_i] = \mathrm{sign}\{M(\mathrm{x_i}+) - M(\mathrm{x_i}-)\}$. A physically meaningful initial marking M_0 will be such that $M_0(\mathrm{x_i}-) + M_0(\mathrm{x_i0}) + M_0(\mathrm{x_i}+) = 1$.

2. Associate to each input u_k a Petri net with three places $\mathrm{u_k}-$, $\mathrm{u_k0}$ and $\mathrm{u_k}+$, as in Figure 2.(c). Here a token in place $\mathrm{u_k}-$ means that $[u_k] = -$, and so on. A physically meaningful initial marking M_0 will be such that $M_0(\mathrm{u_k}-) + M_0(\mathrm{u_k0}) + M_0(\mathrm{u_k}+) = 1$.

3. In the net of each u_k introduce four transitions as in Figure 2.(c), whose firing will denote the crossing of a landmark value. As an example, the transition from $\mathrm{u_k}+$ to $\mathrm{u_k0}$ will fire when the qualitative value $[u_k]$ goes from + to 0. The transitions are *controlled transitions*, i.e., they will fire according to external events and are represented as empty boxes.

4. The qualitative value of the state variable x_i will change according to the qualitative value of its derivative. Due to the ambiguity of the qualitative sum, $[\dot{x}_i]$ *may* be positive when there exists at least a positive term in the RHS of eq. (2) , i.e., when there exists at least a state variable x_j such that $[x_j] = [a_{i,j}]$, or an input u_k such that $[u_k] = [b_{i,k}]$. To represent this behavior, add several transitions in parallel from $\mathrm{x_i}-$ to $\mathrm{x_i0}$ and from $\mathrm{x_i0}$ to $\mathrm{x_i}+$, one for each term in the sum at the RHS of eq. (2). A similar reasoning can be applied when variable x_i is decreasing.

   In Figure 2.(a), are represented the parallel of the increasing transitions with two single transitions t_C and t'_C labeled $\mathbf{L_C}$, $\mathbf{L_C}'$, and the parallel of the decreasing transitions with two single transitions t_D and t'_D labeled $\mathbf{L_D}$, $\mathbf{L_D}'$, following the notation defined in Section 2.

   To determine the value of the labels on the transitions we will consider two different cases.

(a) $[a_{i,i}] \in \{0,+\}$. In this case let $\mathbf{L_C} = \mathbf{L_C}' : \left(\bigvee_{j\in J} \mathbf{F_j}\right) \left(\bigvee_{k\in K} \mathbf{G_k}\right)$ and $\mathbf{L_D} = \mathbf{L_D}' : \left(\bigvee_{j\in J} \mathbf{F_j}'\right) \left(\bigvee_{k\in K} \mathbf{G_k}'\right)$ where $J = \{j \mid j \neq i, [a_{i,j}] \neq 0\}$, $K = \{k \mid [b_{i,k}] \neq 0\}$, and

$$\mathbf{F_j} : \begin{cases} \mathrm{x_j}+, \text{ if } [a_{i,j}] = + \\ \mathrm{x_j}-, \text{ if } [a_{i,j}] = - \end{cases} \quad \mathbf{F_j}' : \begin{cases} \mathrm{x_j}+, \text{ if } [a_{i,j}] = - \\ \mathrm{x_j}-, \text{ if } [a_{i,j}] = + \end{cases} \quad \mathbf{G_k} : \begin{cases} \mathrm{u_k}+, \text{ if } [b_{i,k}] = + \\ \mathrm{u_k}-, \text{ if } [b_{i,k}] = - \end{cases} \quad \mathbf{G_k}' : \begin{cases} \mathrm{u_k}+, \text{ if } [b_{i,k}] = - \\ \mathrm{u_k}-, \text{ if } [b_{i,k}] = + \end{cases}$$

With these labels we have introduced a transition for each term in the RHS of eq. (2), except for the term $[a_{i,i}][x_i]$. In fact, when $[a_{i,i}] = 0$, the term $[a_{i,i}][x_i]$ will be missing from the RHS of eq. (2). When $[a_{i,i}] = +$, we should consider several cases. If $[x_i] = 0$, again the term $[a_{i,i}][x_i]$ will not affect the RHS of eq. (2) and thus there will be no corresponding transition in the parallel of transitions represented by t'_C and t_D. If $[x_i] = +$, the term $[a_{i,i}][x_i]$ can never contribute to give a negative value to $[\dot{x}_i]$ and thus there will be no corresponding transition in the parallel t'_D. Finally, if $[x_i] = -$, the term $[a_{i,i}][x_i]$ can never contributes to give a positive value to $[\dot{x}_i]$ hence there will be no corresponding transition in the parallel t_C.

(b) $[a_{i,i}] = -$. In this case the labels $\mathbf{L_C}'$ and $\mathbf{L_D}$ are constructed as before. However, the two parallels of transitions t_C and t'_D consist of two single transitions with no label as in Figure 2.(b). In fact, when $[x_i] = +$ ($[x_i] = -$), because of the ambiguity of the qualitative sum, the term $[a_{i,i}][x_i]$ could give a negative (positive) value to $[\dot{x}_i]$ and thus the transition t'_D (t_C) could fire regardless of the marking of the other subnets.

Model with finer quantity space

We now assume that the quantity space of the variables be partioned in finer intervals, so as to avoid the ambiguity of qualitative sum, as discussed in Section 2. In particular, each state variable x_i and each input u_k takes qualitative values in the set $\{-n, \cdots, 0, \cdots, n\}$. The coefficients $[a_{i,j}]$ and $[b_{i,k}]$ are assumed to be integers (this can be done with a suitable normalization).

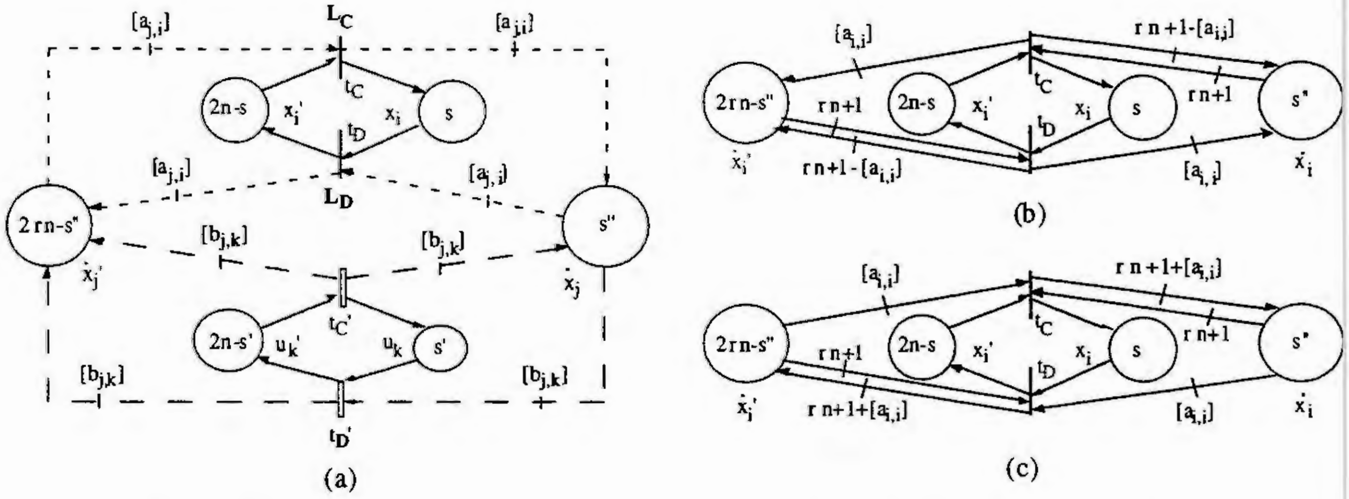


Figure 3: (a) Petri net model with quantity space $\{-n, \cdots, n\}$. (b) Modified net for $[a_{i,i}] < 0$. (c) Modified net for $[a_{i,i}] > 0$.

Algorithm 2 Consider the qualitative equations (2).

1. Associate to each state variable x_i a Petri net with two places x_i and x_i', as in Figure 3.(a). The qualitative value of x_i is related to the marking of the net as follows: $[x_i] = M(\mathrm{x}_i) - n$. A physically meaningful initial marking M_0 will be such that $M_0(\mathrm{x}_i) + M_0(\mathrm{x}'_i) = 2n$. Thus, when there are, say, $n+3$ tokens in place x_i and $n-3$ tokens in place x_i' the qualitative value of x_i is $[x_i] = 3$

2. Associate to each input u_k a Petri net with two places, labeled u_k and u_k', as in Figure 3.(a). The value of $[u_k]$ is related to the marking of this net in the same way discussed for the x_i subnet.

3. Associate to each variable $\dot{x}_j$ a Petri net with two places, labeled $\dot{\mathrm{x}}_j$ and $\dot{\mathrm{x}}_j$', as in Figure 3.(a). Since $[\dot{x}_j]$ is defined by eq. (2), its quantity space is $\{-rn, \cdots, 0, \cdots, rn\}$, where $r = \sum_i \mid [a_{j,i}] \mid + \sum_k \mid [b_{j,k}] \mid$. Thus, the value of $[\dot{x}_j]$ is related to the marking of the net as follows: $[\dot{x}_j] = M(\dot{\mathrm{x}}_j) - rn$.

   Since the initial value of $[\dot{x}_j]$ is a function of the qualitative values of the state variables and inputs, a physically meaningful initial marking M_0 will be such that

   $$M_0(\dot{\mathrm{x}}_j) = \sum_{i \in I} [a_{j,i}] M_0(\mathrm{x}_i) - \sum_{i \in I'} [a_{j,i}] M_0(\mathrm{x}'_i) + \sum_{k \in K} [b_{j,k}] M_0(\mathrm{u}_k) - \sum_{k \in K'} [b_{j,k}] M_0(\mathrm{u}'_k)$$

   where $I = \{i \mid [a_{j,i}] > 0\}$, $I' = \{i \mid [a_{j,i}] < 0\}$, $K = \{k \mid [b_{j,k}] > 0\}$, and $K' = \{k \mid [b_{j,k}] < 0\}$. The initial marking of the complementary place will be $M_0(\dot{\mathrm{x}}'_j) = 2rn - M_0(\dot{\mathrm{x}}_j)$.

4. The qualitative value of the state variable x_i will change according to the qualitative value of its derivative. Thus, two transitions will be introduced in each x_i subnet, as in Figure 3.(a). The increasing (decreasing) transition t_C (t_D) may only fire when $[\dot{x}_i] > 0$ ($[\dot{x}_i] < 0$) moving a token from x_i' to x_i (from x_i to x_i'), thus it will have a label $\mathbf{L_C : \dot{x}_i = rn + 1}$ ($\mathbf{L_D : \dot{x}'_i = rn + 1}$).

   Each time the value $[x_i]$ changes, according to eq. (2) there will be a corresponding change in all the $[\dot{x}_j]$ such that $[a_{j,i}] \neq 0$. Thus, the firing of the transitions in each x_i subnet may also change the token content of the places in some $\dot{x}_j$ subnet. This can be modeled adding arcs of weight $[a_{j,i}]$ between the transitions of x_i and the places of $\dot{x}_j$. As an example, in Figure 3.(a) the dotted arcs correspond to a coefficient $[a_{j,i}] > 0$. The direction of the arcs should be reversed if $[a_{j,i}] < 0$. Finally, these arcs will not be present if $[a_{j,i}] = 0$.

   This construction needs to be partially modified for arcs between the transitions in the x_i subnet and the places in the $\dot{x}_i$ subnet, arcs that will be present if $[a_{i,i}] \neq 0$. In fact, transition t_C associated to x_i may fire only if $[\dot{x}_i] > 0$, i.e., if there are at least $rn + 1$ tokens in place $\dot{\mathrm{x}}_i$. If $[a_{i,i}] < 0$, the firing of t_C will remove $[a_{i,i}]$ tokens from place $\dot{\mathrm{x}}_i$ and add $[a_{i,i}]$ tokens to place $\dot{\mathrm{x}}_i$'. A similar reasoning can be applied to the firing of transition t_D. This behavior is captured in the construction shown in Figure 3.(b), where we have removed the labels in the transitions t_C and t_D because we have explicitely represented the self-loops. If $[a_{i,i}] > 0$, we need to use the construction shown in Figure 3.(c).

5. In each u_k subnet introduce two controlled transitions t'_C and t'_D, as in Figure 3.(a), whose firing will denote the crossing of a landmark value.

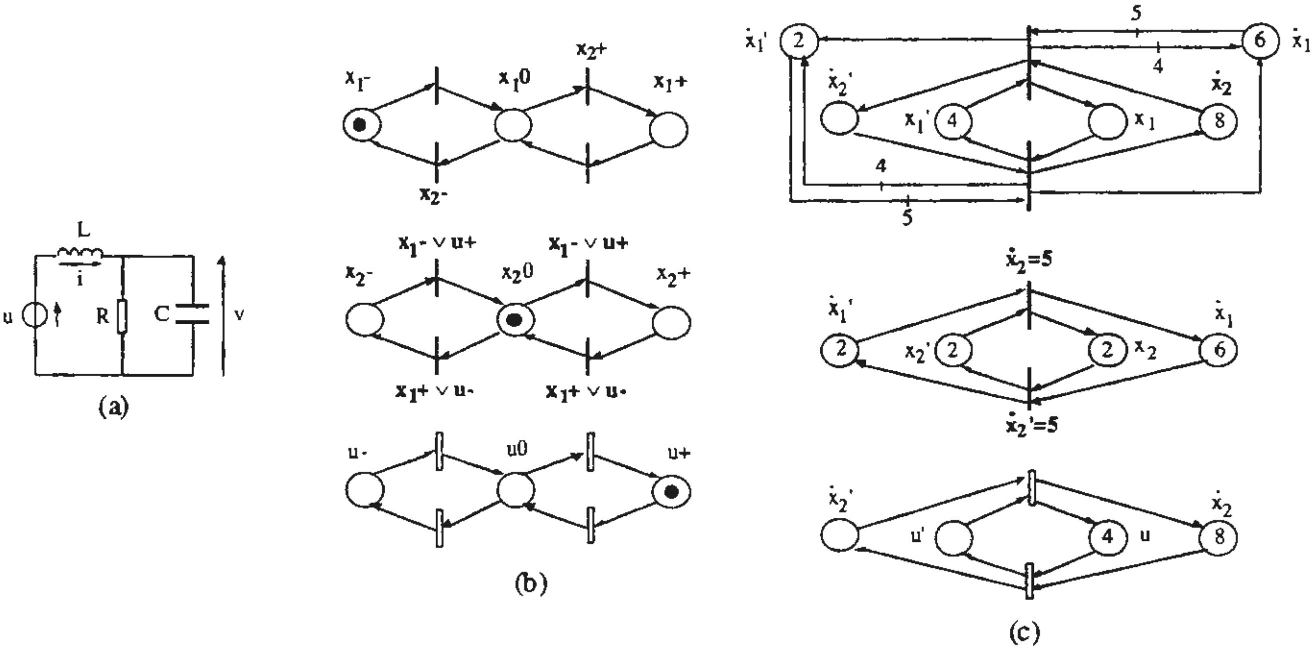


Figure 4: Applicative example. (a) Electrical circuit. (b) Petri net model with quantity space $\{-,0,+\}$. (c) Petri net model with quantity space $\{-2,\cdots,2\}$

Each time the value $[u_k]$ changes, according to eq. (2) there will be a corresponding change in all the $[\dot{x}_j]$ such that $[b_{j,k}] \neq 0$. Thus, the firing of the transitions in each u_k subnet may also change the token content of the places in some $\dot{x}_j$ subnet. This can be modeled adding arcs of weight $[b_{j,k}]$ between the transitions of u_k and the places of $\dot{x}_j$. As an example, in Figure 3.(a) the dashed arcs correspond to a coefficient $[b_{j,k}] > 0$. The direction of the arcs should be reversed if $[b_{j,k}] < 0$. Finally, these arcs will not be present if $[b_{j,k}] = 0$.

The previously described construction may be simplified if eq. (2) contains only one term for a given $\dot{x}_i$. In fact, in this case the qualitative value of $\dot{x}_i$ is equal to the qualitative value of a state variable x_j or of an input u_k (possibly changed of sign). Thus, we need not introduce the $\dot{x}_i$ subnet. Examples of this case will be discussed in Section 4.

4. Example

Consider the second order circuit in Figure 4.(a). The state variables are $x_1 = v$, and $x_2 = i$. Assuming unitary values of R, L, and C the state equations and the corresponding qualitative equations for this system are:

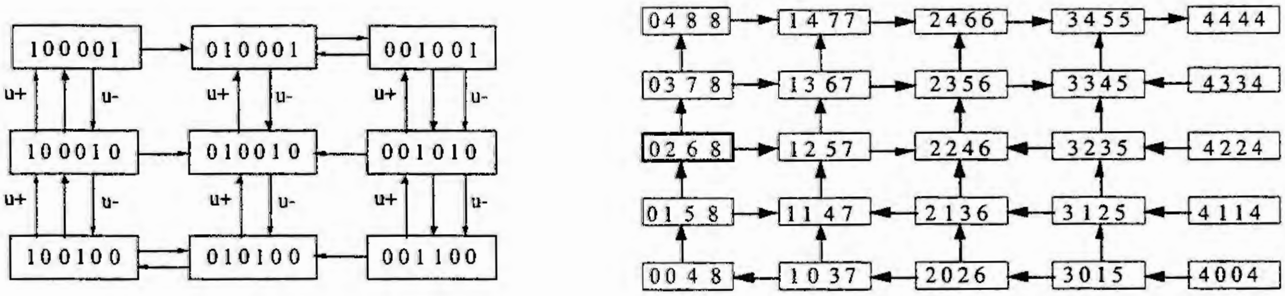
$$\left\{ \begin{array}{lcl} \frac{dv}{dt} & = & -\frac{1}{CR}v + \frac{1}{C}i \\ \frac{di}{dt} & = & -\frac{1}{L}v + \frac{1}{L}u \end{array} \right. \qquad \left\{ \begin{array}{lcl} [\dot{x}_1] & = & -[x_1] + [x_2] \\ [\dot{x}_2] & = & -[x_1] + [u] \end{array} \right.$$

Figure 4.(b) shows the Petri net model obtained with Algorithm 1, assuming a quantity space $\{-,0,+\}$, while Figure 4.(c) shows the Petri net model obtained with Algorithm 2, assuming a quantity space $\{-2,\cdots,2\}$.

In Figure 5.(a) we have given the reachability graph of the net in Figure 4.(b). Note that for easiness of representation we have projected the reachable markings over the state space of the two nets, i.e., any marking shown in the figure is of the form $M = [M(\mathrm{x}_1-)\ M(\mathrm{x}_10)\ M(\mathrm{x}_1+)\ M(\mathrm{x}_2-)\ M(\mathrm{x}_20)\ M(\mathrm{x}_2+)]^T$. In the figure, arcs labeled u+ (u-) may only fire when $[u] = +$ ($[u] = -$).

Assume now the forcing input has a constant value $[u] = 0$. The behavior of the net is given by the graph in Figure 5.(a) where all arcs labeled u+ and u- are removed. By inspection, we see that starting from any initial state it is possible to reach the steady state $\{[x_1] = 0, [x_2] = 0\}$, corresponding to marking $[0\ 1\ 0\ 0\ 1\ 0]^T$. This state is a steady state because no arcs are leaving it.

On the contrary, if the forcing input has a constant value $[u] = +$, the behavior is given by the graph in Figure 5.(a) where all arcs labeled u- are removed. In this case, one can see that no steady state will exist according to this model. This is in contradiction with the actual behavior of the system. This discrepancy is due to the ambiguity of the qualitative model chosen to describe it, and can be avoided using a finer quantity space.



(a)

(c)

Figure 5: (a) Reachability graph of the net in Figure 4.(b). (b) Rechability graph of the net in Figure 4.(c) for $[u] = +2$.

As an example, assuming a quantity space $\{-2, \cdots, 2\}$, with constant input $[u] = +2$, we obtain the graph in Figure 5.(b), where for simplicity we have represented in each node the vectors $[M(\mathrm{x}_1)\ M(\mathrm{x}_2)\ M(\dot{\mathrm{x}}_1)\ M(\dot{\mathrm{x}}_2)]^T$. In this case, we see that starting from any intial state, after a finite number of steps the steady state marking $[4\ 4\ 4\ 4]^T$ (corresponding to the state $\{[x_1] = 2, [x_2] = 2, [\dot{x}_1] = 0, [\dot{x}_2] = 0\}$), is reached.

5. Conclusions

The paper discussed how Petri nets may be used for the qualitative modeling of physical systems.

Given the quantitative description of a physical system behavior, the corresponding qualitative description is derived and is compiled into a Petri net structure. Different Petri net structures may be used to represent the same qualitative behavior depending on the choice of the variable quantity space. Both systems described by homogeneous differential equations and systems with external forcing inputs have been considered.

There are some advantages in using Petri nets to represent the qualitative behavior of a system. Firstly, there is a simple and intuitive correspondence between the marking of the net and the state of the system. Secondly, the dynamic behavior of the system can be studied as sequences of reachable markings of the net, as we have shown in the applicative example.

References

[1] J. deKleer, "How Circuits Work," *Artificial Intelligence*, Vol. 24, pp. 205–280, 1984.

[2] A. Fanni, A. Giua, D.-Y. Lee, "Petri Nets in the Qualitative Modelling of Systems,," *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics* (San Antonio, Texas), pp. 2316–2321, October, 1994.

[3] K.D. Forbus, "Qualitative Process Theory," *Artificial Intelligence*, Vol. 24, pp. 86–168, 1984.

[4] B. Kuipers, "Qualitative Simulation," *Artificial Intelligence*, Vol. 29, pp. 289–338, 1986.

[5] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Pro. IEEE*, Vol. 77(4), pp. 541–580, 1989.

[6] K. Okuda, T. Ushio, "Petri Net Based Qualitative Simulation," *Proc. IASTED Int. Symp. on Expert Systems Theory and Applications* (Los Angeles, California), December, 1990.

[7] K. Okuda, T. Ushio, "Hierarchical Qualitative Simulation for Large Scale Dynamic Systems," *Applications of Artificial Intelligence in Engineering VI*, Rzevski and Adey (eds.), pp. 301-317, Elsevier.

[8] P.J. Ramadge, W.M. Wonham, "The Control of Discrete Event Systems," *Proc. IEEE*, Vol. 77(1), pp. 81–98, 1989.

[9] List of Petri net tools: `http://www.daimi.aau.dk/~petrinet/tools/db/`

DISCRETE EVENT MODELS IN BATCH CONTROL

Michael Tittus and Knut Åkesson
Control Engineering Lab, Chalmers University of Technology
S-412 96 Gothenburg, Sweden
e-mail: mt/ka@control.chalmers.se

Abstract. An automata-based approach for the modeling of batch plants as well as products is presented. The different units of a plant are modeled as bounded Petri nets and products are represented by way of their recipes. With the focus on synchronization and booking issues, we propose general Petri net building-blocks for the construction of these recipes. Both, resource and recipe models support formal supervisor synthesis for dynamic resource allocation according to the Ramadge-Wonham framework.

1 Introduction

Batch processes take an important place in process industries. A batch process involves a sequence of phases that are carried out on a discrete quantity of material within a piece of operating equipment (resource). The control of this kind of systems is discrete in order to achieve transitions between different control modes, and continuous within these modes. As examples for the different kinds of control involved we can mention traditional steady-state set-point control or tracking as typical continuous control tasks, and sequencing and synchronization as two typical discrete control tasks.

A model becomes meaningful first when it fills some purpose, e.g. simulation, control-design, analysis. In this paper we propose models of the plant and its discrete specifications that (1) support the synthesis of discrete control and (2) easily can express the resulting behavior of the controlled plant. For an overview of different modeling techniques for batch processes, we refer the reader to a survey found in [1]. In this study the authors compare high-level Petri nets, temporal logic and Minimax algebra as modeling tools. However, to our knowledge, the synchronization issues associated with material transfers have not been considered.

We introduce Petri net models for both plant resources and recipes (product specifications). Even though plant resources work continuously, it is possible to classify their behavior into discrete states. A transition between states is caused by the occurrence of events signaling the beginning or the end of continuous tasks on one hand and a recipe demanding access to a resource on the other.

In batch processes, a recipe or product specification is a sequence of operations to be performed on certain quantities of raw materials and results in the product. Since the most accurate characterization of a product is by means of its recipe it is natural to model each product by its recipe. A for our purposes suitable model of the product specification not only expresses sequences of operations but also the related synchronization of needed resources in a formal way. The proposed models give special attention to synchronization mechanisms between resources when material flow is involved. This is a particularly interesting problem since, unlike in most manufacturing systems, both, source and target resource have to cooperate to move material.

Thus, we consider recipes consisting of five general kinds of elements - sequence of operations, moving material, different ways to join material, adding material during an operation, and the splitting of material. These elementary task models are formalized with respect to resource booking and the necessary synchronization of involved resources. Furthermore, general and reusable Petri net building blocks representing these elementary tasks are introduced. Using these building blocks a recipe model can easily be put together.

The possible behavior of the controlled process can be expressed by synchronizing the different resource models with the recipe model. By this we mean that transitions in resources and recipe that are equipped with identical labels are forced to fire simultaneously, thus forcing the different Petri net models to execute in synchrony. From this process model a booking model is easily derived, which controls the booking and unbooking of resources. It can be used to algorithmically synthesize a discrete supervisor that coordinates the simultaneous execution of several recipes within the plant [6]. The supervisor synthesis is done according to an extension [2] of the Wonham-Ramadge approach [3].

We start by introducing resource models. After discussing different synchronization and booking issues, a recipe model is proposed. Some conclusions end the paper.

2 Modeling Resources

In this paper we assume the plant to consist of two generic classes of resources (equipment devices), namely *processors* (units) and *transporting devices*. *Processors* are typically tanks, reactors, and other container-like units, fully equipped with control modules and other devices to manipulate a batch. *Transporting devices*, on the other hand, have as their main task to open and close connections between processors causing and preventing material flow. A special case are supply tanks, that is, tanks that contain the raw materials a batch product consists of. We assume them to be infinite in content (that is, sufficient content for the given recipes), and are modeled as on/off valves, which, when booked, denote that one recipe has gained access to the corresponding supply tank. In this paper we exemplify processors with tanks and transporting devices with on/off valves, and use bounded Petri nets as our modeling tool.

Modeling Processors

Our main concern is to keep the Petri net representation as general and flexible as possible. In [6] processor models consisting of three states (places) have been introduced. In the present approach, which focuses more on the booking of resources, only two control places (states) are necessary (see Fig. 1). Each processor has a number of places and a unique event alphabet. Whenever a processor model is created, a set of unique events are assigned to the Petri net template by adding the modeled resource's ID as an index to each event.

Modeling Transporting Devices

Besides the natural states UNBOOKED and BOOKED, two extra states in which the valve is kept closed, BLOCKED_1 and BLOCKED_2, are needed (see Fig. 1). These two extra states are necessary since a valve with two outlets can be blocked by at most two recipes. Each recipe blocking one of the valve's outlets.

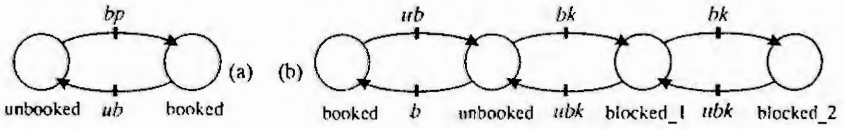


Figure 1: Automata representation of (a) a processor and (b) a valve template

When moving material from one processor to the next a number of valves have to cooperate to open and close certain connections. Thus, a higher-level class of transporters, called *connection line* or just *line*, is created.

Modeling Connecting Lines A line is an abstract object that has purely supervisory functions. For each possible connection between any two processors a line object can automatically be created from information about the plant's topology. A line serves as a kind of mediator, booking and coordinating the different valves needed to open and close a connection. Its booking automata is shown in Fig. 2. At the event bl_{ij}, meaning "book line ij", the line object books or blocks the different valves needed for controlling the connections, as soon as they become available. A valve V_k, for example, is booked, if event b_k can be triggered in both the line automaton and the automaton representing V_k. This is automatically achieved by synchronizing the two automata. That is, events by the same name *have* to happen simultaneously.

When the line is ready and set, signalled by the event lok_{ij} ("line ij ok"), it is closed by default. When opening and closing the line, the corresponding valves are opened or closed, respectively. Event lub_{ij} initiates the unbooking of the line and its associated valves.

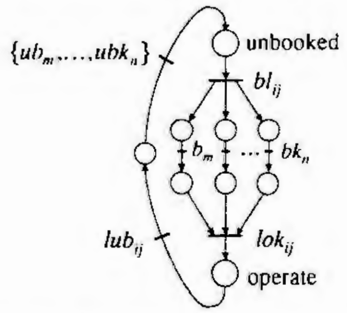
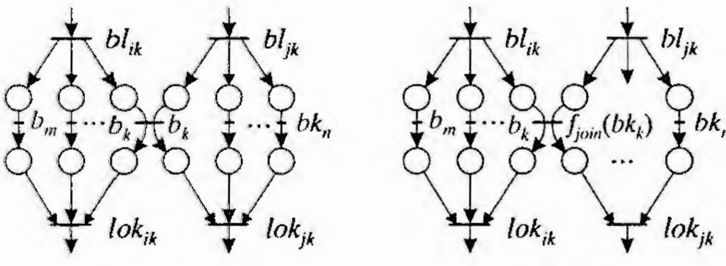


Figure 2: PN representation of a connecting line



case 1

case 2

Figure 3: Synchronous Booking of Lines

3 Modeling the Recipe

Product models or recipes can be specified on different levels of abstraction. Following the SP88, one can distinguish between the plant-independent general recipe, which describes the operations to be applied without reference to any specific equipment to be used, and the master recipe, which, besides the operations, also shows the raw materials path through a specific plant. Recipes on an even lower abstraction level describe the detailed control that has to be applied.

In this paper we will concentrate on models to specify the plant-dependent master recipe. These models then can be used together with the previously introduced plant models to synthesize discrete supervisors that coordinate the execution of different recipes. We will here model the following basic functions:

- *operation:* different phases applied to a part of the batch
- *move:* moving the batch from one unit to another
- *add:* adding material into a unit that already contains part of the batch (e.g. as part of an operation)
- *join:* merging of two parts of a batch from two source units into a third target unit
- *split:* separating of a batch into two disjunct parts

Modeling Operations

An *operation* is a major processing sequence applied to the whole or a well-defined part of the batch. Each operation has to be executed within only one unit and no two operations can be applied to the same part of the batch concurrently. An operation consists of a set of different phases that can be executed in sequence or parallel. They start at a well-defined initial state/set of states and are supposed to terminate in some final (set of) state(s), satisfying a number of constraints on the way.

For our purpose it suffices us to represent each operation as a single place in a PN. Whenever a token is placed in this place, a local, hybrid supervisor belonging to the current unit and synthesized according to the hybrid supervisory control theory (HSCT) introduced in [5] and [4], is taking control. Its purpose is to transfer the system safely from its current initial system-state to some final state. As soon as this final state is reached, the transition exiting from the PN place is enabled.

Joining of Material Flows

In order to better understand the different ways two independent branches of a batch can be joined, and the accompanying mechanisms, we now interpret a join as the mixing of the contents of two tanks P_1 and P_2 into a third tank, say P_3.

The *general join* consists of a pre-synch phase, a synchronization point and concludes with a post-synch phase. In the pre-synch phase, both material flows (from P_1 to P_3 and from P_2 to P_3) are started independently of each other and execute until they each reach some pre-specified synchronization point. During the post-synch phase both material flows are executed *in synchrony* with each other. For this, a separate specification in the form of a defined order in which to join the different flows, or continuous restraints can be added, which control the coordination of the two flows during the join. The hybrid control synthesis introduced in [6] can be used under certain restrictions to generate fitting controllers that guarantee that the restrictions will be satisfied.

One special case of the general join is the *synchronous join*. Here the pre-synch phase is empty, that is, both joining parts of the batch have to be ready and synchronized before they are combined. This implies, that all required resources have been booked by the recipe. Normally the joining is controlled by specifications. No specifications imply that after the initial synchronization both flows are executed independently of each other, as for example a tank that is to be filled simultaneously from two inlets.

The other extreme is an *asynchronous join*. Taking the same interpretation as above, in the asynchronous case neither of the source tanks has to wait for the other in order to empty its content into the target tank. Both branches of the join act independently of each other and are synchronized after the filling is finished. This corresponds to an empty post-synch phase.

In the sequel, only the asynchronous and the synchronous join are of interest. A special application of the asynchronous join is a so-called *add*, which describes the adding of one material flow to another as part of an operation. An example is the adding of a catalyst at a certain point of a reaction.

Booking of Resources

All resources have to be booked by a recipe before they can be utilized by that particular recipe. For the representation of an operation this has no consequence since we already assume the batch to be contained in the unit. All other basic PN building blocks, however, are concerned with transferring material from one unit (*source*) to another (*target*). Before any material can be transferred, the *target unit*, together with the *line* that connects the two units, needs to be booked by the recipe. A line is only booked when both the target and source unit have already been secured by the recipe. This guarantees the line's availability and avoids the occurrence of circular waits on the line/unit level.

Synchronous Booking of Connecting Lines Assume that the contents of two units, P_1 and P_2, are supposed to be merged into unit P_3. Two lines, namely P_1P_3 and P_2P_3, need to be booked by the same recipe. Assume further that both lines need to either book or block the same valve, say V_k. Since the opening and closing of both lines is to be controlled by the same local supervisor, it can be assured that the common valve is opened and closed correctly and this kind of double-booking is allowed.

From the booking point of view we distinguish between three cases:

- *Both lines need to block V_k:* No competition arises, since blocking a valve does not book it.
- *Both lines need to book V_k:* The two line-events denoting the booking of V_k are synchronized, i.e. they are treated as only one event and fired simultaneously (see Fig. 3, case 1). Whenever a valve is booked, it is closed by default.
- *The two lines need to block and book V_k, respectively:* Valve V_k is *booked* by the recipe. This is done by changing the label bk_k to b_k and then treating the two identically labeled events as one event as above (see Fig. 3, case 2).

In the last two cases, the recipe has to resolve the competition for the valve and guarantee that only one line is opened at a time.

More formally: Let $\Sigma_{1,3}$ and $\Sigma_{2,3}$ denote the event alphabets of line automata P_1P_3 and P_2P_3 respectively and let $\Sigma_{\text{join}} = \Sigma_{1,3} \cup \Sigma_{2,3}$. Then we define a re-labeling function f_{join}: $\Sigma_{\text{join}} \to \Sigma_{\text{join}}$ for $\tau \in \Sigma_{\text{join}}$ such that

$$f_{\text{join}}(\tau) = \begin{cases} b_i & \text{if} \quad \tau = bk_i \wedge b_i \in \Sigma_{\text{join}} \\ \tau & \text{if} \quad otherwise \end{cases}$$

As a result of this possible competition for transporting devices, we distinguish between two ways of booking resources in connection with join constructs: *synchronous booking* and *asynchronous booking*. *Synchronous booking* implies that both lines have to be booked at the same time, and can, for example, be used when a recipe needs to book two lines that share some transporting device. Otherwise, *asynchronous booking* can be employed, i.e. both lines are booked independently of each other.

For all synchronous joins, synchronous booking is used (S) while asynchronous joins can be associated with asynchronous (A) line booking, independent of whether or not there is a competition for transporting devices.

General Building Blocks for Material Transfer

Each transfer of material requires at least three resources: a target unit, one or more source units and connecting line(s). As soon as all involved source units are ready for material transfer, the system goes through the following steps: (1) Booking of target unit and corresponding lines, (2) preparation of target unit (preprocessing), (3) the actual material transfer, and (4) the post-processing (e.g. cleaning) and unbooking of all source units.

To be able to efficiently model a general building block, we first need to define some operations on PNs.

Different Synchronization Operations The *full synchronous* composition operator $PN_1 \parallel PN_2$ models the interaction of two concurrently executing PNs, PN_1 and PN_2. This interaction requires simultaneous participation of all the involved nets on mutually labeled transitions.

A finite set of events $\{\tau_1, \ldots, \tau_n\}$ labeling a transition t is called an *event connection*. In this case when t fires all events $\tau_1, \ldots, \tau_n$ are triggered simultaneously. As an example consider three Petri nets PN_1, PN_2, PN_3. Assume that PN_1 contains a transition labeled with the connected event $\{\tau_1, \tau_2\}$ and PN_2 and PN_3 contain an event τ_1 and τ_2, respectively. No other mutual events exist. If these three Petri nets are to be executed using full synchronous composition then the three transitions labeled with τ_1, τ_2 and $\{\tau_1, \tau_2\}$ can only be fired simultaneously.

As a last operation we introduce the *alternative connection* of events, denoted by $\langle\tau_1, \tau_2, \tau_3\rangle$ with τ_i denoting transition labels. The *alternative connection* is interpreted as follows: As soon as τ_1 is ready to fire it is *connected* to either τ_2 or τ_3, depending on which of these is first enabled, and the newly found set of connected events, either $\{\tau_1, \tau_2\}$ or $\{\tau_1, \tau_3\}$, is fired in direct sequence. After that, the remaining transition, τ_3 or τ_2, is fired as soon as it is enabled. It is important to note that the firing of the transition labeled that way cannot start without the firing of τ_1.

This operator is easily generalized to the case where τ_2 and τ_3 can be substituted by sets of *connected* events. Furthermore, if the PN is branching following an alternative connection, labeled paths can be specified to be taken depending on the events fired. We then write $\langle\tau_1, \{\tau_j\}^1, \{\tau_k\}^2\rangle$ for an alternative connection, with the PN interpretation shown in Fig. 4.

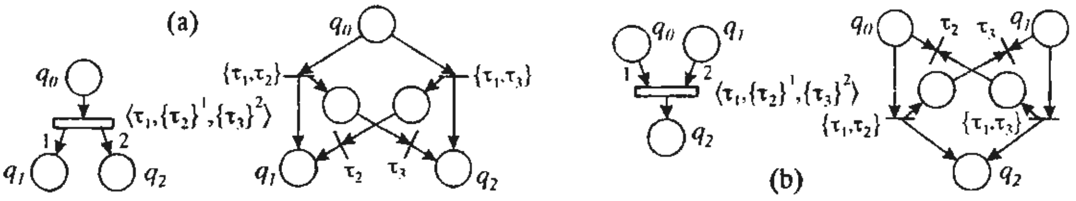


Figure 4: The *alternative connection* operators with labeled paths (cf. σ_1 and σ_2 in Fig. 5) and their PN interpretations

Building Blocks for Material Transfer We start by proposing generic PN building blocks that illustrate booking and synchronization aspects of the different kinds of material transfer. Using these models we can then deduce booking models which are used to synthesize a discrete supervisor for dynamic resource allocation.

Figure 5(a) shows the generic building block used to model the two different join constructs (A, S). All transition labels are generic plant events assuming material transfer from two source units, P_i and P_j to the target unit P_k, using lines P_iP_k and P_jP_k. In the case of an S-join the bottom part of the building block can be collapsed as shown in Fig. 5(b). In case one or both of the source units are supply tanks, it was mentioned before that supply tanks are modeled as on/off valves and are thus booked by the corresponding line. If, for example, the generic P_i is a supply tank, then all generic events from P_i's Petri net are omitted.

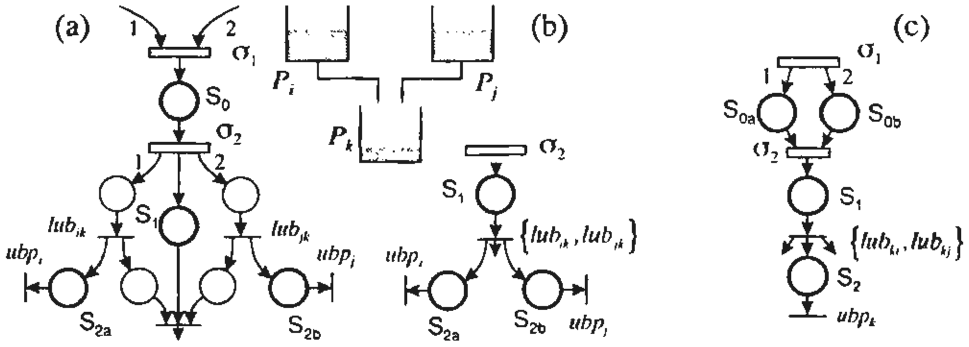


Figure 5: Generic building blocks for material transfer: (a) A-join, (b) collapsed bottom part of S-join, and (c) split

The building block expressing a split is shown in Fig. 5(c)where the batch is split from one source unit, P_k, into two target units, P_i and P_j. Note that both target units are booked independently of each other in the first case. An alternative way would be to connect the events booking the two target units and the corresponding lines. The main disadvantage of this alternative is that both units need to be available simultaneously for the booking to take place; a fact that easily could lead to starvation if there are several recipes competing for the plant's resources. The interpretation of the local controllers is as before. The special case where one part of the batch is extracted into a target unit, while the rest stays in the original unit can be derived by pruning the split block. Analogously, a simple move can be derived from either of the two building blocks by deleting the unnecessary places.

The transitions labeled $\sigma_i, i = 1, 2$ represent important synchronization points where synchronization between different resources is required. This synchronization is achieved by connecting the corresponding events in the plant's resource models.

σ_1 coordinates the booking of target unit and corresponding lines, and guarantees that no line is booked without the target unit being secured. σ_2, on the other hand, ensures that the lines are ready and the target unit preprocessed before material transfer is started.

This building blocks also contains local supervisors (denoted by S), which control activities on a lower hierarchical level:

- S_0 leads the target unit through a number of phases with the purpose of preparing it for operation.
- S_1 controls the actual material transfer by opening and closing the corresponding line(s). S_1 continues to control the target unit(s) even after the source unit(s) have been emptied, so as to allow the material transfer to be part of another operation. The nature of S_1 depends very much on the kind of material transfer modeled (synchronous or asynchronous, join or split).
- S_2 is used to post-process a source unit before releasing it. After S_2 has accomplished its task, the unit is released (unbooked).

The following table gives the interpretation of the σ-labeled transitions for the A, the S and the split case, respectively.

	S join	**A join**	**batch split**
σ_1	$\{bp_k, bl_{ik}, bl_{jk}\}$	$\langle bp_k, \{bl_{ik}\}^1, \{bl_{jk}\}^2\rangle$	$\langle \epsilon, \{bp_i, bl_{ki}\}^1, \{bp_j, bl_{kj}\}^2\rangle$
σ_2	$\{lok_{ik}, lok_{jk}\}$	$\langle \epsilon, \{lok_{ik}\}^1, \{lok_{jk}\}^2\rangle$	$\{lok_{ki}, lok_{kj}\}$

Note that each building block can be divided into an upper *booking part* and a *transfer part* (below σ_2). Thus, when building a recipe different booking strategies can be implemented.

Booking Models From the point of view of a supervisor that coordinates the booking and unbooking of resources, the only relevant information is to know when resources are booked and when they become available again. That is, it is of no interest for a supervisor whether a unit is preprocessed, ready to receive an already booked batch or already operating on a batch. All these cases imply that the unit is booked by some recipe and hence not available.

The booking models for the different material transfers are given in Fig. 6 and are obtained by eliminating all events that do not book or unbook some resource.

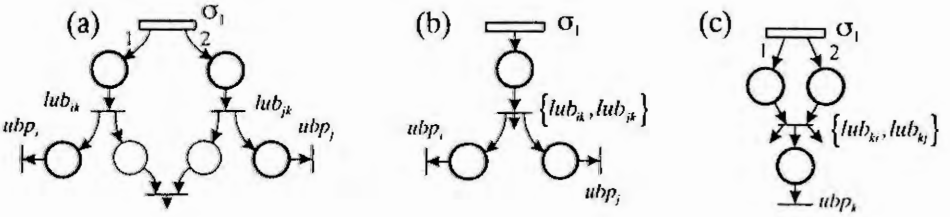


Figure 6: Petri net representation of booking models

4 Conclusions

Generic discrete-event models suitable for supervisor synthesis have been presented. Starting with these representations, a maximally permissive supervisor can be synthesized that coordinates the execution of simultaneous recipes by allocating resources. Since complexity is the major setback of the supervisor synthesis, two kinds of building blocks have been introduced: a somewhat more elaborate to focus on synchronization and booking issues and a more compact one only focussing on resource allocation.

Acknowledgement: The authors thank Prof.s Bo Egardt and Bengt Lennartson for valuable discussions and insights. Financial support from the Swedish Board for Industrial and Technical Development (NUTEK) under grant no. P4851-1 and the Swedish Research Council for Engineering Sciences (TFR) under grant no. 92-185 is gratefully acknowledged.

References

[1] E.P. Patsidou E.C. Yamalidou and J.C. Kantor. Modeling discrete-event dynamical systems for chemical process control — A survey of several new techniques. *Computers chem. Engng.*, 14(3):281–299, 1990.

[2] M. Fabian and B. Lennartson. A class of non-deterministic specifications for supervisory control. In *Proc. of ECC'95*, Rome, Italy, 1995.

[3] P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control Optim.*, 25(1):206–230, January 1987.

[4] M. Tittus. *Control Synthesis for Batch Processes.* PhD thesis, Control Eng. Lab, Chalmers Univ. of Techn., Göteborg, Sweden, 1995.

[5] M. Tittus and B. Egardt. Control-law synthesis for linear hybrid systems. In *Proc of 33rd CDC*, pages 961–966, Orlando, FL, USA, 1994.

[6] M. Tittus, M. Fabian, and B. Lennartson. Controlling and coordinating recipes in batch applications. In *Proc. 34th CDC*, pages 2484–2489, New Orleans, LA, 1995.

USING GENERAL PURPOSE DISCRETE SIMULATORS FOR MICROSOPIC MODELLING AND SIMULATION OF TRAFFIC SYSTEMS

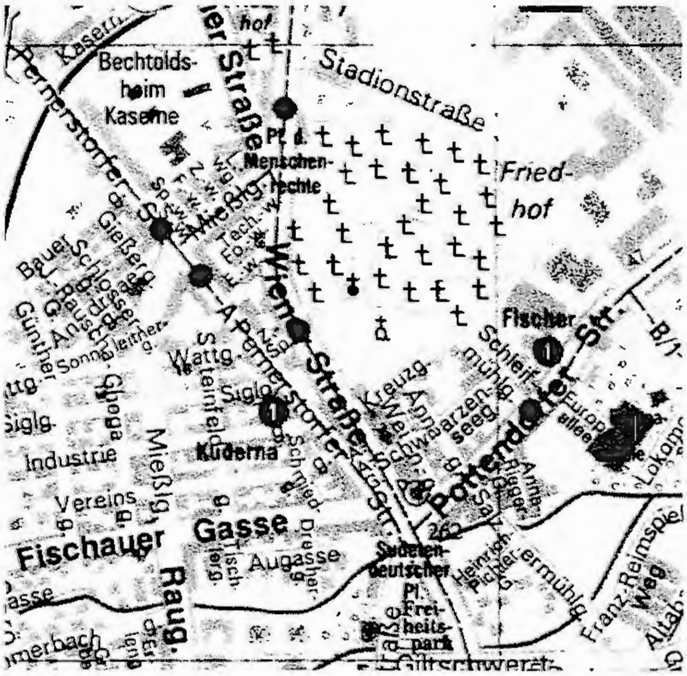
C. Kiss[1], M. Klug[2], F. Breitenecker[3]
Dept. for Simulation Techniques, Technical University of Vienna
Wiedner Hauptstraße 8 - 10, A-1040 Vienna, Austria
[1] jeanluc@osiris.tuwien.ac.at; [2] mklug@osiris.tuwien.ac.at; [3] Felix.Breitenecker@tuwien.ac.at

Abstract. The aim of this paper is to discuss the advantages but also the problems of using general purpose discrete simulators for traffic simulation. This will be investigated by two representatives of different classes of simulators: i. e. GPSS/H - a textual modelling language, and Micro Saint - a modelling language with a graphical user interface. This two languages should demonstrate their features for modelling and simulation of a traffic system of an Austrian city.
As case study a system of four junctions and a roundabout included in an open traffic system were chosen. As conclusion there is a summary of the advantages and disadvantages of the chosen simulation language.

Traffic system

The junctions of interest are located in Wiener Neustadt. The traffic engineers have implemented a system of traffic manipulation which tries to make the public transportation more effective and convenient. The busses are favoured at the junctions. This is realised by a radio signal which is sent from the arriving busses and the traffic light cycle is adjusted in order to grant the busses as soon as possible the passing of the junction.

The following picture describes the situation:



Four main streets are leading into the traffic system: Fischauergasse, Pottendorfer Straße, Wiener Straße, Grazer Straße (continuing to Wiener Straße). You can go from Fischauergasse to Wiener Straße by passing Mießlgasse and from Wiener Straße to Pottendorfer Straße by passing Stadiongasse. At the northern end of Mießlgasse you have a roundabout built there, all the other junctions are controlled by traffic lights.

The data and all other information to build a fitting model have been collected by the town government of Wiener Neustadt.

Short description of Micro Saint

Micro Saint is a process based, discrete simulation language with a graphical user interface. It is available on many software platforms, its system requirements are very low. The models are stored in an ASCII - format which eases transportation or even editing.

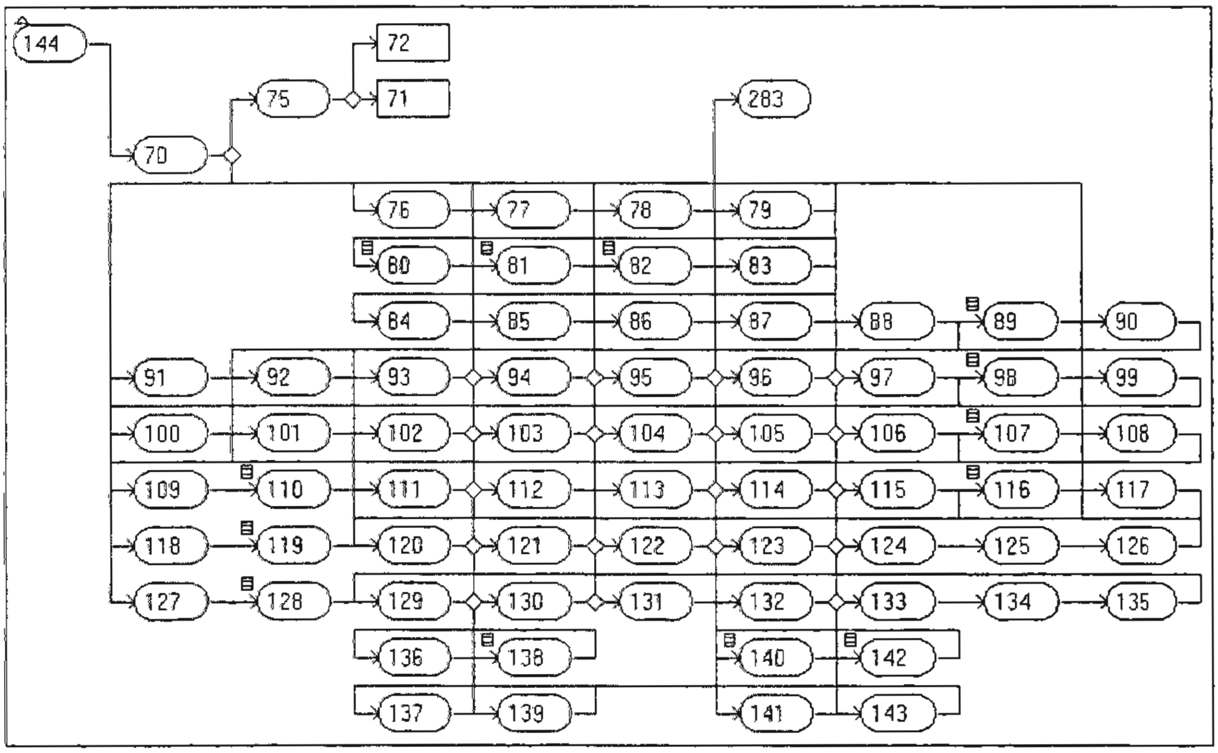
Using Micro Saint is easy to learn. Its structure is very useful for small simulation problems which should be solved within a short time. For huge models it is necessary to deepen the knowledge of Micro Saint and its features like user defined functions and the use of variables. Micro Saint also provides the possibility to use programming structures like „if" statements and loops. Micro Saint also provides functions for tracing data and for creating an animation.

At the end of 1996 there was published a new version of Micro Saint which is adopted for the common operation systems. There have been also repaired some bugs of the old version and some new features were implemented. In general Micro Saint is a very useful tool for simulation for everyone's purpose.

Modelling of Traffic Systems in Micro Saint

The simulation language Micro Saint consists of four main elements: the entity, the task, the path and the decision. The entity is the object which goes it's way along the path changing the way or splitting at decisions through a network of decisions. Therefore the entities are representing the various vehicles, the tasks represent parts of the street. The decisions are used to model the decision of a vehicle about the way to go.

The following screen shot of a part of the Micro Saint model shall demonstrate how the real system can be modelled in this simulator. The oval items are tasks which are representing parts of the street with a length of about 7m. The rectangles show several subnets, which are containing tasks as well, but they are used for controlling the traffic lights and pedestrians. The lines represent the paths that are used by the entities (vehicles). The small rectangles beside the tasks are showing the queues, where the vehicles are waiting for green light. The small rhombs are the symbols for the decisions. Whenever a path is splitting such a decision guides the entity into the right direction.



Micro Saint - network diagram of the model of the investigated junction.

Some implemented features of the model

Various types of vehicles are implemented by means of numerical attributes of entities. It is possible to define different values for the length, the average speed, the standard deviation of the speed, the mean acceleration, the standard deviation of the acceleration. The acceleration after the stop at the traffic light is also implemented as well as the changing the speed with regard on the speed of the vehicle in front.

Furthermore the traffic concept of Wiener Neustadt is implemented as far as the busses in the model are able to announce themselves at the next junction, and then the traffic light control of the junction reacts and adjusts its cycle to provide the passage for the bus as soon as possible.

Some features that are not implemented

Some properties of real life traffic are not implemented. The reason for this is to find in some bugs of the simulation language. For example it is not implemented, that the vehicles reduce their speed under the speed of the vehicle in front. It was tried to relieve this system by using queues, but there is a mistake in the administration of the queues which leads to deadlocks.

Another feature that has not been implemented was the Psycho-Physic-Distance-Model (PPDM). It would have been possible to have this feature implemented, but it would have made the model more complex and would have led to unacceptable runtimes.

Advantages and disadvantages of Micro Saint for modelling and simulation of traffic systems

The main advantage of Micro Saint is as well its main disadvantage: The very simple and easy to learn structure. But especially for modelling traffic systems it is often necessary to formulate complex system logic such as the Psycho-Physic-Distance-Model (PPDM), where the simple structures are insufficient.

A very positive aspect of modelling traffic systems in Micro Saint derives from the fact that it is not necessary to create an animation. If the structure of the model is chosen with regards to appearance of the real system the network diagram acts also as an animation. A further advantage is the possibility to control most features with variables.

Some negative characteristics can be found in the lack of alphanumeric variables and that it is not possible to get information about entities that are in a task when this task is not active. The user defined functions are quite useful but the possibility of using parameters makes it very difficult to calculate even simple formulas.

The conclusion of this investigations is that Micro Saint is only suitable for small traffic systems, but it is very useful to understand the important characteristics of modelling and simulation of traffic systems.

Short description of GPSS/H

GPSS/H is a text-based simulation language with many "blocks" and commands for modelling and controlling a simulation, GPSS/H is a compiler based one and not an interpretative program, therefore GPSS/H is very fast in doing simulation, but there are no possibilities to change the model description during the simulation itself. The blocks describe stations, where the "entities" flowing through the system are handled in some way. A sequence of blocks builds up the model. The commands allow various experiments, including iterated simulation runs with statistical evaluations.

GPSS/H is a very powerful "programming language" with easy to learn basic elements. To create a more complex model it is necessary to study details of the language, which takes a couple of time. GPSS/H offers a wide range of possibilities to define functions or subroutines. Interfaces to C and FORTRAN programs are completing the features of this simulation language.

A complete output file shows a wide range of results including all queues used, resources, etc. This file is also stored in ASCII format. To compute more results, there may be defined other result files to be imported into any calculating / statistical program. Such user defined files are also necessary for doing animation with Proof Animation®.

GPSS/H is available for the most computer operating systems including Windows 95 and Windows NT, and runs very stable.

Modelling of traffic systems in GPSS/H

To create a model for a traffic system, it is essential to think how to model the street between two junctions. This section of a street has a certain length, and a car occupies a certain space in this section according to the length of the car and to the distance to the car in front. There are additional characterisations for each car, (velocity, reaction time, etc.). The vehicles are represented as "entities" and the above mentioned characterising parameters mark these entities as attributes.

There is a principle problem: The order of the cars passing the street should not change, while each car may drive at its own speed. A section of a street is modelled by a resource (station) with a capacity corresponding to its length. Therefore at the begin the length of this part of a street has to be fixed exactly. If the car (entity) enters such a street section, it is splitted: the first copy occupies its space in the storage, the other copy is stored into a FIFO user queue to control the order of the cars coming along. To get these two parallel flows of entities merged, the entities have to stay in a "MATCH" block waiting for each other. This concept can be implemented in GPSS/H in a very easy way.

Additionally, at the end of a street section a facility (resource with capacity 1) is implemented, controlling the temporary distance (reaction time) of two cars. This is important, because the differences in length, e.g. between a bike and a truck, are also effecting the time the entities take to pass the street at this ending point. This modelling strategy guarantees, that the forementioned problem of individual velocity and order of cars can be solved efficiently.

After modelling this complex submodel of a street section the overall model can be built with these submodels, whereby single resources are used to link them. Also flow of traffic may be locked and opened there, in order to model the traffic signals. Extra queues have to be defined there to simulate the dangerous parts of a junction or a roundabout, and to control the higher priority of the public transport.

The whole model has a capacity of more than 2800 vehicles in a hour. The simulation of a whole day takes about 10 to 12 minutes on a 486DX4-133.

Advantages and disadvantages of GPSS/H for modelling and simulation of traffic systems

GPSS/H is a textual simulation language, also with "basic" features of a programming language. Consequently, nearly everything can be modelled (programmed). Furthermore, GPSS/H is a compiler based language and therefore extremely fast in running a simulation, and GPSS/H also offers a very good random number generator.

Creating an output file, GPSS/H offers a standard output file with plenty of data stored. This file is very hard to read especially for a novice in simulation.

While the graphical model of Micro Saint can be seen as a kind of animation, there is nothing similar included into GPSS/H on its own. It is necessary to create an additional output file for the animation package *Proof Animation.*

In GPSS/H, modelling and programming of almost everything can be done, but the environment is very inconvenient and "old-fashioned", but GPSS/H is very fast. In GPSS/H also other approaches for modelling of traffic systems may be implemented, but the modeller must be an expert user.

References

1. Hubschneider; H. Mikroskopisches Simulationssystem für Individualverkehr und öffentlichen Personennahverkehr, Dissertation 1982
2. Steierwals, G. & Künne, H.-D. Stadtverkehrsplanung, Springer - Verlag Berlin Heidelberg 1994
3. ARGESIM ARGE Simulation News Diskrete Simulation, Moderne Grundlagen und ausgewählte Anwednungen, Seminarunterlage S32, Eigenverlag 1995
4. ASIM Arbeitsgemeinschaft Simulation in der Gesellschaft für Informatik, Mitteilungen aus den Arbeitskreisen Heft Nr. 38, Eigenverlag in Braunschweig 1993
5. ASIM Arbeitsgemeinschaft Simulation in der Gesellschaft für Informatik, Mitteilungen aus den Arbeitskreisen Heft Nr. 41, Eigenverlag in Braunschweig 1993
6. Knoflacher, H. & Macount, T., Ökologie und Straßenverkehr, Umweltbundesamt 1989
7. Breitenecker, F., Diskrete Simulationsyteme, Skriptum TU Wien
8. Troch, I., Modellbildung und Simulation, Skriptum TU Wien
9. Sammer, G., Detailplanung von Verkehrslichtsignalanlagen in Wiener Neustadt, Bericht
10. Schriber, Thomas J. [1991]: An Introduction to Simulation using GPSS/H. Wiley, New York, NY
11. Schriber, Thomas J. [1974]: Simulation using GPSS. Wiley, New York, NY
12. Banks, Jerry; Carson, John S. II; Sy John Ngo [1989]: Getting started with GPSS/H. Wolverine Software Corporation, Annandale, VA
13. Henriksen, James O.; Crain, Robert C. [1989]: GPSS/H - Reference Manual. 3rd Ed.; Wolverine Software Corporation, Annandale, VA
14. Wolverine Software Corporation [1995]: GPSS/H Professional System Guide. Ergänzungen zum Handbuch, Annandale, VA

FROM PETRI-NET SIMULATIONS TO EFFECTIVE INSENTIVE SYSTEMS

Marcel Hutter and Edouard Schmid
Hutter & Partner Dendrit Risk Management
Hintergasse 1, CH-8180 Bülach, Tel/Fax: +0041-1-861 07 70

Abstract. Our aim is to convince managers to incorporate simulation results into the financial accounting, especially when dealing with fortuitous processes. A total risk management framework is developed for the example of combined gas/steam power plants.

The complex decision making process in the field of combined gas/steam power plant construction with its reliability-, availability-, and maintainability (RAM) risk cannot be handled adequately without simulation. We start on the technical level by defining a Petri-net representation of a combined gas/steam power plant. As a first result we get the distributions for the RAM relevant variables. We then take into account the different specifications in the sales contracts and derive the corresponding financial distributions. With the help of these distributions, we are in a position to decide on risk adjusted journal entries, thus are able to generate the stable and meaningful contract related income or expense.

Only now we are ready for a fair performance measurement and correspondingly are able to build effective insentive systems for all involved acteurs of the decision process. A harmonized approach, integrating the views of RAM engineers, techniciens, salesmen, insurance experts, traders, accountants, financial controllers, treasurers, top management, and shareholders is envisaged.

Introduction.

Many processes in banking, insurance, but also in the industrial business are clearly fortuitous. Of course, in all these fields many models have already been developed to represent the corresponding random processes. What we would like to see is a consequent further processing of the model results on the financial and accounting side. Unfortunately, this has only been done so far in the insurance business, and to a certain extent in banking, but definitely not for industrial business.

We recommend in all these fields to estimate the distributions for the relevant financial variables. These distributions should than also be the basis for every journal entry.

In our view, the most meaningful accounting approach whenever chance plays an important role, is to distinguish between **expected** and **unexpected** contributions. The basis for a journal entry which affects the profit and loss accounts should be the expectation value of the financial variable. As the deviations from the expectation are random and consequently cannot be attributed meaningfully to any of the acteurs involved (except to the top management and the shareholders) they should be managed by means of one risk pool clearly declared in the balance sheet. Therefore, the responsibility of managing this very risk pool can certainly not be attributed to the salesman. It is clearly the top management and the shareholders that should have a coherent concept of how to handle risks within their enterprise.

A first attempt in this direction is the recently implemented credit risk management system ACRA (Actuarial Credit Risk Accounting) [3] by the Swiss Banking Corporation, one of the three largest Swiss banks. The core of the ACRA model is also based on distributions and on the distingtion of expected and unexpected losses of both capital and interest. However, the accounting is slightly different than what we suggest.

In this paper we put the focus on an example from the energy industry. The sale contract for a combined gas/steam power plant is not so far away from any banking or insurance contracts. It is a question of the related underlying characteristics. A power plant contract usually guarantees the buyer a certain level of annual energy output. As the actual energy output is of course a random variable, we treat this sale contract exactly as explained above. The accounting concept is the **'expectation variance'** principle.

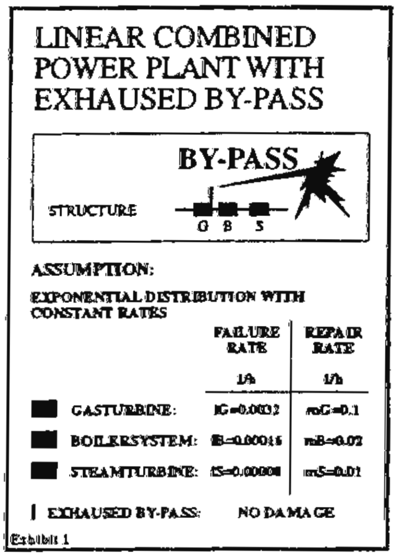
We start on the technical level by defining an adequate Petri-net representation of a combined gas/steam power plant. As a first result, after simulating within the Petri-net model, we get distribution curves for the RAM relevant variables. We then take into account the guarantee, penalty, refund and price specifications defined in the contract between the buyer and the seller. The combination of these contract specifications and the distributions for the technical variables lead finally to the distributions of the financial variables. They must be the basis for all further management considerations. Book keeping practices must be tightly linked to these curves; they may not "live a separate life" but must be totally harmonized. Thus, with the help of these financial distributions, we are in a position to decide on risk adjusted journal entries.

A total risk management framework is developed analyzing three different sales contracts and discussing in detail the following above mentioned views: RAM engineer, techniciens, salesmen, and financial controllers, accountants respectively.

The View of the RAM engineer

Without adequate RAM analysis, the fundamental technical questions concerning complex gas/steam power plants cannot be solved appropriately. RAM studies may be accomplished in many different ways. The most demanding and powerful but also timeconsuming way is the building of a model that is based on simulation tools. Our simulation model is built and discussed using the Petri-net approach.

The Petri-net model as defined by the RAM engineer is nothing else than an other representation of a Marcov process [2]. The analyzed combined gas/steam power plant consists of one gas-turbine (G) with built-in exhaused by-pass, one steam-turbine (S) and is linked through one boiler-system (B). Every element is dependent on the other. All assumptions concerning their failure and repair rates are shown in Exhibit 1.


LINEAR COMBINED POWER PLANT WITH EXHAUSED BY-PASS

ASSUMPTION:

EXPONENTIAL DISTRIBUTION WITH CONSTANT RATES

	FAILURE RATE 1/h	REPAIR RATE 1/h
■ GASTURBINE:	lG=0.0032	mG=0.1
■ BOILERSYSTEM:	lB=0.00015	mB=0.02
■ STEAMTURBINE:	lS=0.00008	mS=0.01

\| EXHAUSED BY-PASS: NO DAMAGE

Exhibit 1

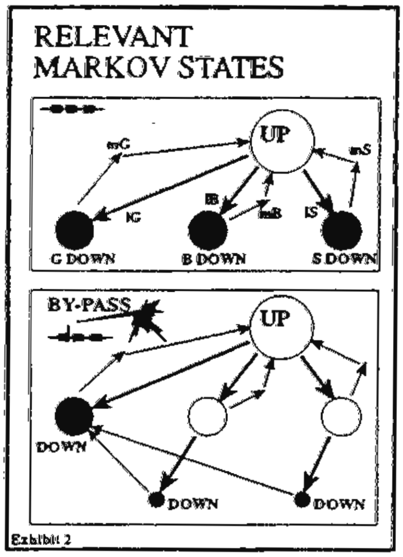

RELEVANT MARKOV STATES

Exhibit 2

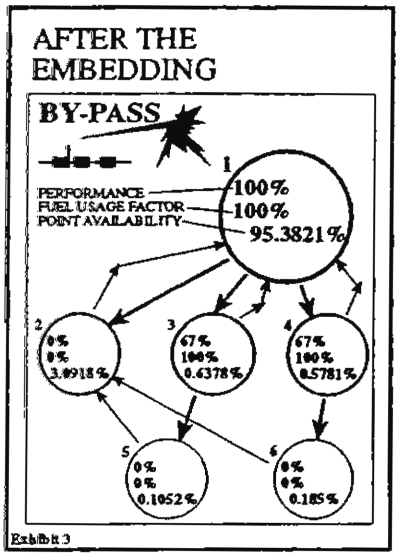

AFTER THE EMBEDDING

Exhibit 3

The RAM engineer knows that the exhaused by-pass never causes damage. Moreover, he knows that a system with built-in exhaused by-pass behaves slighdy different compared to a system without by-pass (Exhibit 2). The by-pass has the effect that the total system observed will not totally break down when the boiler-system or the steam-turbine falls out. However, it will fall into a state with a lower performance level due to not being linked to the combined effect anymore. Therefore, he defines a Markov model that incorporates the fact that if either the boiler-system or the steam-turbine breakes down, than the total system will still produce energy, however, on a lower level. He assumes for those two states a performance of 67%.

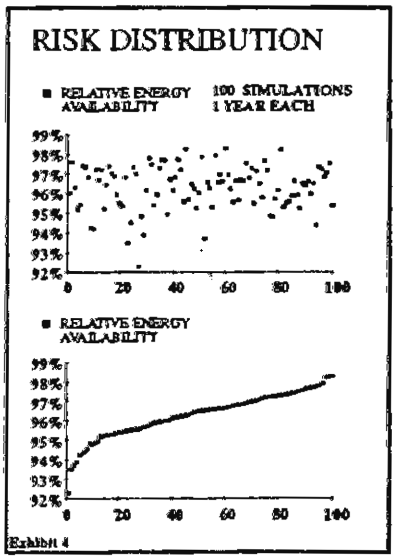

RISK DISTRIBUTION

Exhibit 4

The advantage of analyzing within the Markov states which have been finetuned with respect to both the reliability block diagram and the reality gets now obvious. On the one hand, it is possible to define repair strategies by linking the states differently, on the other hand, new and important information can be put into each state. The embedding of the data is shown in Exhibit 3. Three variables have been embedded, the performance, the fuel usage factor and the point availability. First, the RAM engineer belives to know that the system will be repaired by one repair crew (with outfallsequence strategy). After testing the corresponding failure-, and repair rates with Kolmogoroff-Smirnow [1] the RAM engineer finds that the failure and repair times are well represented by an exponential distribution with constant failure-, and constant repair rates. Second, analyzing Markov processes enables one to embed those important figures which are needed to transform all technical variables into managerial information.

In this example, we are only interested in the energy availability per year. That's why we take the energy availability of the total system at the end of each year as the relevant RAM variable. As a matter of fact, we are not interested in the point availability, which discribes only the output (stationary state) of an endless simulation (and which could be calculated numerically solving differentials). We are much more interested in a distribution curve for this very RAM variable. Therefore, it is not astonishing that we simulate within the Petri-net. Yearly background runs enable us to generate such an energy availability distribution curve. Obviously, every time when the state is changed, the related time within the corresponding state will be multiplied by the state performance. The result when proceeding as described is shown in Exhibit 4. The focus is put on the simulated yearly energy availabilities. The plotted graphs show exactly the same information of the underlying risk distribution with respect to the RAM variable. However, the outputs in the lower graph are ordered, thus represent our RAM distribution curve. Before finetuning the model to the technicien's view, the expectation of the energy availability is 96.2%.

The View of the Technicien

Discussing the technical behaviour with the techniciens working at the site, the RAM engineer is confronted to three important points which he has neglected so far. First, he learns that in the worst case where two blocks are out, the reparation will always be done with two repair crews working independently. The assumed outfallsequence strategy of only one repair crew was a wrong assumption (Exhibit 5). The model refinement results in a higher energy availability (96.4%).

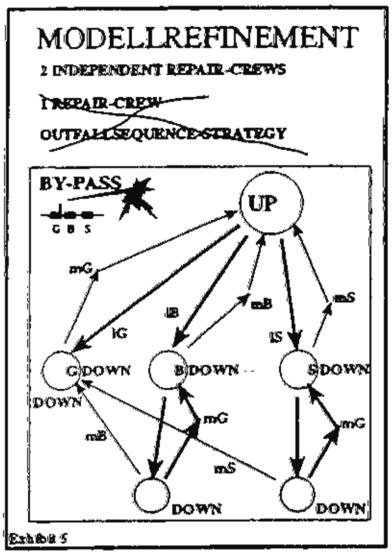


Exhibit 5

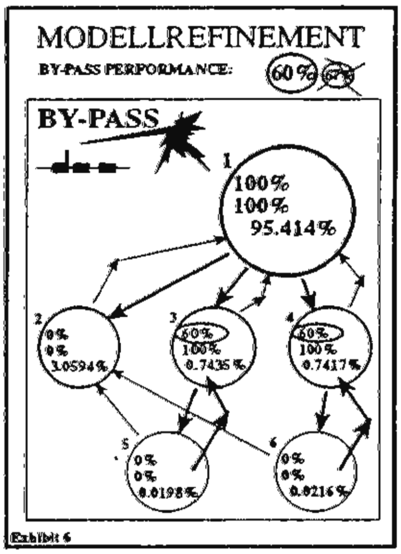


Exhibit 6

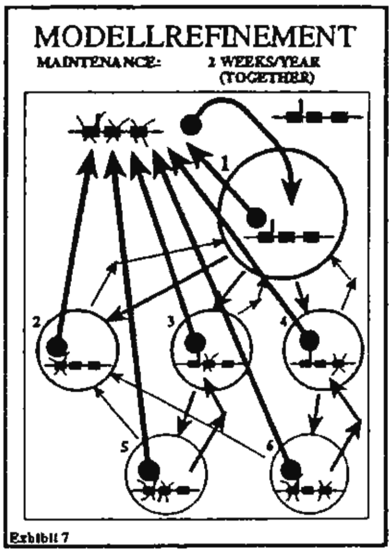


Exhibit 7

Second, the techniciens tell him that within the two low level performance states, not 67% but only 60% is generated due to bad site conditions (humidity, air pressure, etc.). Because of these model refinements the expected energy availability falls to 96.3% (Exhibit 6). Third, they tell him that besides the unplanned repair events, 2 weeks of planned maintenance per year (which is 3.8%) can not be neglected. Within this period, the total system is totally down and does not produce any energy at all. Therefore, the expected energy availability shrinks to (96.3%*(100%-3.8%)) = 92.6%. It might be argued that only under the condition of best maintenance practices, the assumption of exponentialy distributed failure-, and repair rates is fair and might be applied for our model. We therefore immediately adjust slightly our Petri-net model to what the techniciens told us to be observed in reality (even if constant rates were not reasonable, we would not have any difficulty in finetuning the Petri-net model).

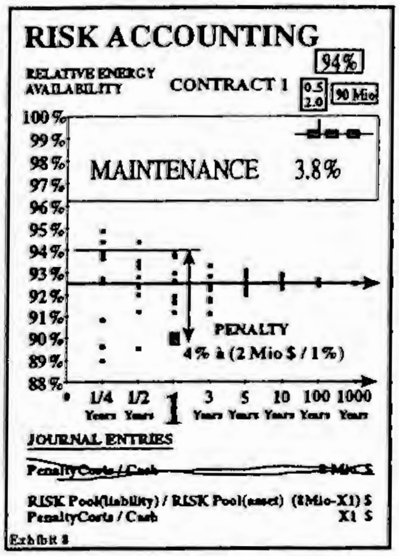
Isolated RAM analysis is of no use (eventhough the refinements of the techniciens have already flown into the model). RAM data should be economized. In consequence, the generated RAM data needs to be integrated in further processings. RAM data should be processed into information. That's when the view of the salesmen contributes to the next logical step of the overall model design. The underlying distribution curve of the adjusted (refinements of the techniciens) energy availability variable is not plotted anymore. However, all further exhibits are built on the correct RAM distribution curve before any financial contract related transformation takes place.

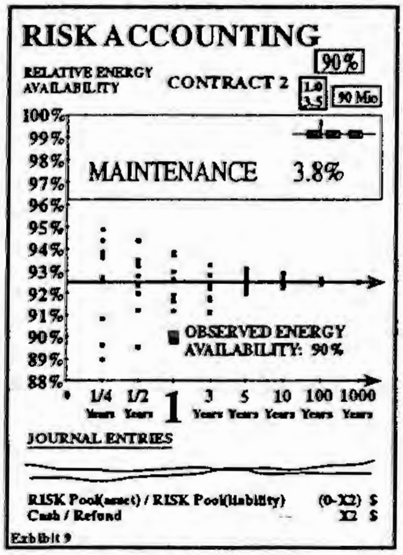
The View of the Salesman

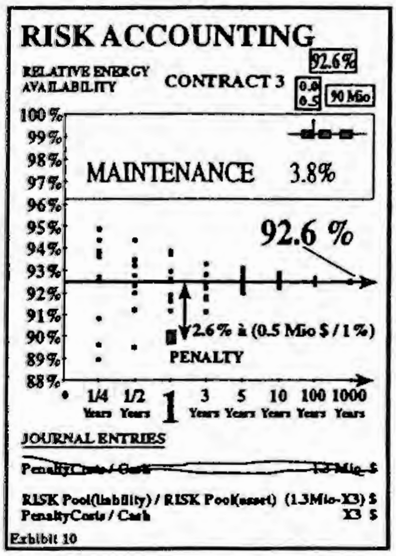
Having full knowledge about the behaviour of the RAM relevant variables, we now link the technical distribution curve to three different typical contracts. We therefore take into account the guarantee, penalty, refund and price specifications defined in the contract between the buyer and the seller. The combination of these contract specifications and the distributions for the technical variables lead finally to the distributions of the financial variables. We take them as basis for all further management considerations. The three contracts show the following specifications:

	Sales Price $	Guarantee %	Penalty /1% /	Refund /1%
Contract 1:	90 Mio $	94%	2 Mio $ /	0.5 Mio $
Contract 2:	90 Mio $	90 %	3.5 Mio $ /	1.0 Mio $
Contract 3:	90 Mio $	92.6%	0.5 Mio $ /	-

According to contract 1, the buyer and the seller have aggreed upon a guaranteed level of energy availability at the end of the year of 94%. Furthermore, a penalty of 2 Mio $ per 1% of lower energy availability will be charged and must be paid by the seller. In the case of producing more than 94% energy availability, a refund of 0.5 Mio $ will be transfered back from the buyer to the seller.







The other two contracts can be interpreted in analogy. Contract 3 is insofar special as there are no refunds paid at all and both the expectation of the energy availability and the guaranteed corresponding figure is the same. In this example, the sales price will not be investigated because it does not influence the yearly cash flow volatility with respect to our underlying RAM distribution.

We assume that at the end of the year all three plants produce the same amount of energy. Let's further assume that the observed energy availability is by random 90%. Remember, that the three plants are exactly the same as far as their structure and their failure-, and repair rates are concerned, thus the underlying drivers in form of our RAM distribution curve is exactly the same, too. The only difference is that the applied financial contracts between the buyer and the seller have other specifications. According to contract 1 a penalty of 4% à 2 Mio $ = 8 Mio $ has to be paid at the end of the year. Do we have to punish the seller of contract 1? The **NOT** risk adjusted journal entry would be Penalty Costs / Cash 8 Mio $. By random, the guaranteed (set in advance) and the observed (at the end of the year) energy availability of plant 2 (contract 2) are alike. No cash will flow out of the organization. What about the salesman of contract 2? Is he really in a neutral position? In such situations, a not risk adjusted accounting would not apply any journal entries. Within a fully risk adjusted accounting system, we do have to generate 2 journal entries, one of them is profit & loss effective, the other is not.

The View of the Financial Controller/Accountant

Again, what's the use of isolated financial distribution curves without adequately linking them to the financial accounting? Non. Only through the fully embedding of this data into the financial book keeping, the aim of creating a stable and harmonized accounting system for the future (with respect to all innovative derivative contracts to come, as well) is to be accomplished.

There is only one way to address and solve all above mentioned questions. What is urgently needed is a transformation of the RAM curve into three financial distribution curves. The overall objective is to calculate the financial expectation value of each contract, separately. Furthermore, accounting practices must be tightly linked to these curves; they may not "live a separate life" but must be totally harmonized. In doing so, a fully risk adjusted accounting system can be built which enables one to coherently create effective insentive systems. After starting another three background runs the financial distribution curves can be plotted. Only now we are really in a position to decide on fully risk adjusted journal entries. The consequent separation between unexpected (random) and expected (non random) contributions is the key. The basis for each and every profit & loss effective journal entry has to be the expectation value of the corresponding financial variable.

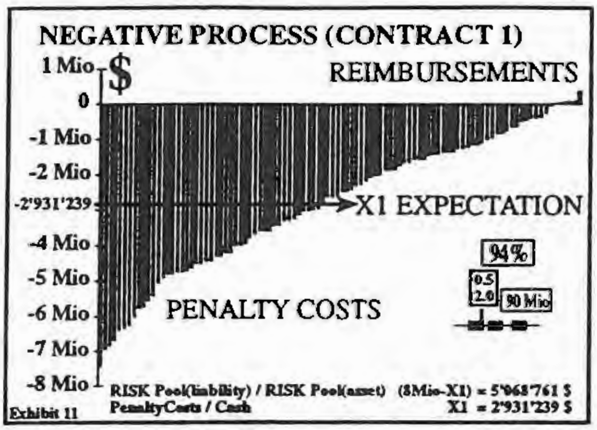


Exhibit 11

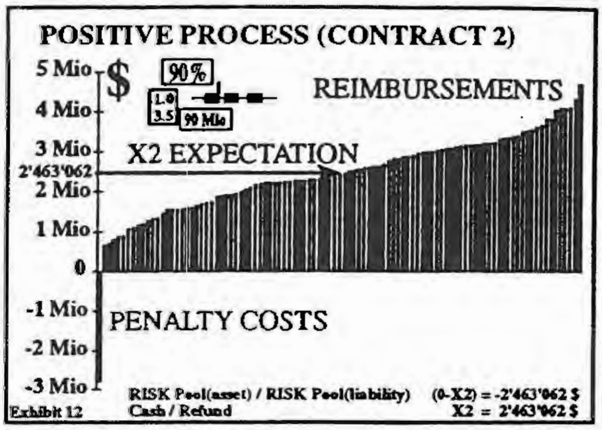


Exhibit 12

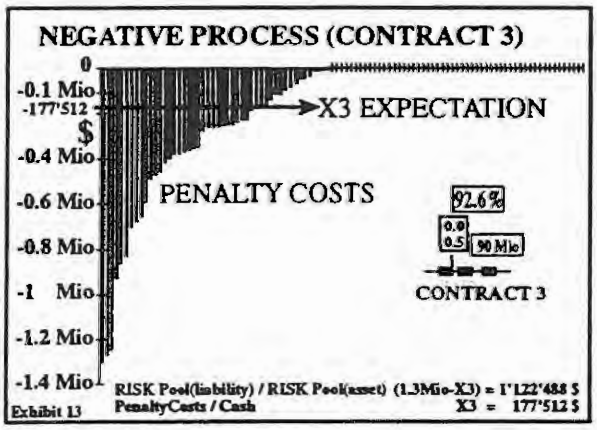


Exhibit 13

For contract 1 (Exhibit 11), we have 2'931'239 $ which is to be interpreted as a negative result, thus is unfavorable. Contract 2 generates 2'463'062 $ which is favorable (Exhibit 12), and contract 3 holds 177'512 $ unfavorable (Exhibit 13). All random deviations from these expectation values have to be managed by means of one risk pool clearly declared in the balance sheet. The random contributions are never subject to be booked into the income statement. Thus, the involved accounts for the journal entries are balance sheet accounts, solely. Only through these practices, it can be guaranteed that the risks can be managed there, where they have to be: in the balance sheet. As a consequence, not the profit and loss will be subject to tremendous fluctuations, but the balance sheet. Within the balance sheet, we do have to indicate a special risk pool on both the asset side and the liability side, where all unexpected contributions will be accounted for. Contract 1, 2, and 3 contribute as follows: -5'068'761 $, -2'463'062 $, -1'122'488 $, respectively. The total unexpected amount of the cash flow volatility is summed up to -8'654'311 $. This figure is to be interpreted as a risk pool shrinkage. One of the key points is the fact that the risk pool is never influenced by either negative or positive shifts. The expectation of the change of the risk pool is at all times 0.

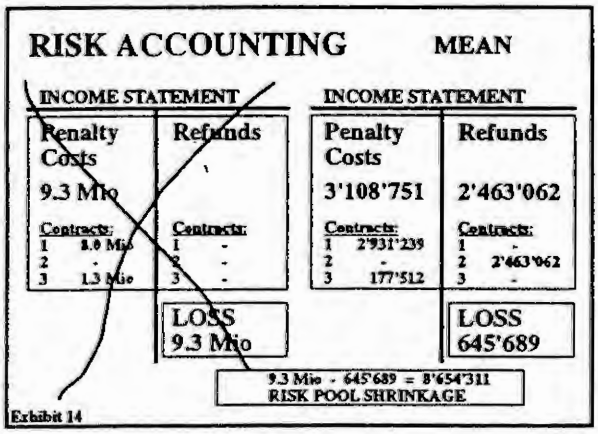


Exhibit 14

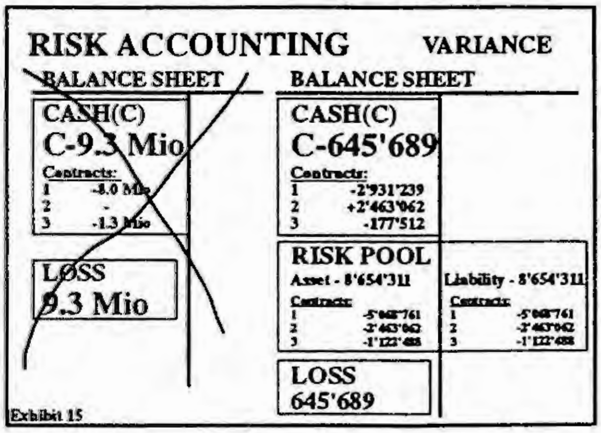


Exhibit 15

At this point, we stop. Eventhough much more risk analysis could now be drived in this framework. The logical determination of the starting risk pool (taking into account some further assumptions from the top management) would be the next step. Questions concerning bankruptcy due to faulty dimensioned risk pools could also be addressed and logically solved.

Summary

The complex decision making process in the field of gas/steam power plant construction with its reliability-availability-maintainability (RAM) risk has been analysed, always refining the model and adjusting it until oeconomically sensful statements could have been derived from.

We showed an incorporating framework generating risk adjusted journal entries having at hand all contract related information and RAM distribution curves. Only through adequate simulation analysis and the fully embedding of the underlying drivers the financial figures on the accounting level can be modelled adequately. The concept was built within the **'expectation variance'** world, separating expected and unexpected financial values, consistently. Without exception, only expected values have been put to the income statement, whereas the unexpected values have been managed by means of the risk pool philosophy clearly declared in the balance sheet.

The most demanding accounting-, and controlling practices with immense innovative risk analysis potential result by applying the 'expectation variance' concept, which puts the focus permanently on the volatility of cash flows by cutting first each single cash flow into two pieces, in an 'expected' cash flow part and in a 'variance driven (unexpected)' cash flow part.

It should be guaranteed, when applying the 'expectation variance' principle, that all relevant information is constantly being harmonized dynamically. Both the expectation values and the random components underly therefore a constant dynamic update. It would not be sufficient to declare only the random contributions as a subject of dynamic change.

Furthermore, this principle can be applied for any contracts harmonizing the risk management in banking, insurance, and industry. All risky products in insurance, banking and industrial business have to be compared to their expectation. Only through the 'expectation variance' principle broken down to the cash flow level a totally new 'total risk management' framework integrating all kinds of risky contracts can be derived from. E.g. interest rate products can easely be combined with storm or earthquake contracts and their derivatives. The ease of accounting such constructs with respect to a fair stability policy can also be demonstrated with the help of the 'expectation variance' principle as overall book keeping rule.

It can not be overemphazised that with this 'expectation variance' principle the intrinsic diversification potentials to be generated on the level of the whole institution are tremendous due to the total global economization of all risks within an organization.

References

1. Birolini, A., Qualität und Zuverlässigkeit technischer Systeme - Theorie, Praxis, Management, Zweite Auflage, Springer-Verlag 1988, S. 218

2. Lampert, D., The Effect of the Structure of Combined Gas/Steam Turbine Plants upon their Availability, Brown Boveri, Publication No. CH-T040 143E

3. Neue Zürcher Zeitung, Acra statt Abracadabra - Das neue Risikosystem des Schweizerischen Bankvereins, Neue Zürcher Zeitung, Nr. 218, Donnerstag, 19. September, 1996, S. 29

PETRI NET SIMULATION OF A DISTRIBUTED AUTOMATION SYSTEM BASED ON PROFIBUS

Georg Marschall
University of the Federal Armed Forces
D-22039 Hamburg, Germany

Abstract. This paper introduces a high level Petri Net model of distributed Automation facilities on the basis of a Profibus-FMS monomaster system, which enables analysis of the application under user aspects by simulation. The application is based on a typical PLC-architecture and Profibus-implementation. The functionality of the system is discussed and modelled in a hierachical net structure, which allows an easy modification of the model by exchanging or extending single subnets. The model is verified by two different aspects, which are indicating the efficiency of automation facilities.

Introduction

The Modelling of Automation Systems can be divided into three main parts: The control algorithm, the plant, that has to be controlled by the algorithm, and the automation facility used for the controlling. A decentral automation facility as shown in fig. 1 consists in general of different Controllers, e.g. PLCs, PCs, ..., which execute the Control Algorithm, and different Control Units, like I/O-modules, Encoders, MMIs, ... for the instrumentation of the plant. All these devices are connected by a communication system, called fieldbus, like Profibus, CAN, FIP, ... depending on the special demands of the automation task. With this, the automation facility represents the controltechnical infrastructure of the automation system.

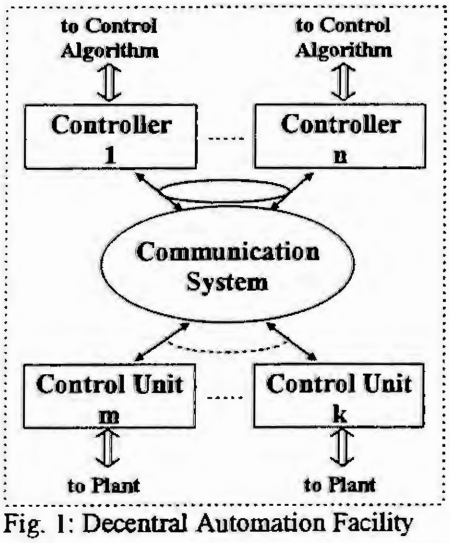


Fig. 1: Decentral Automation Facility

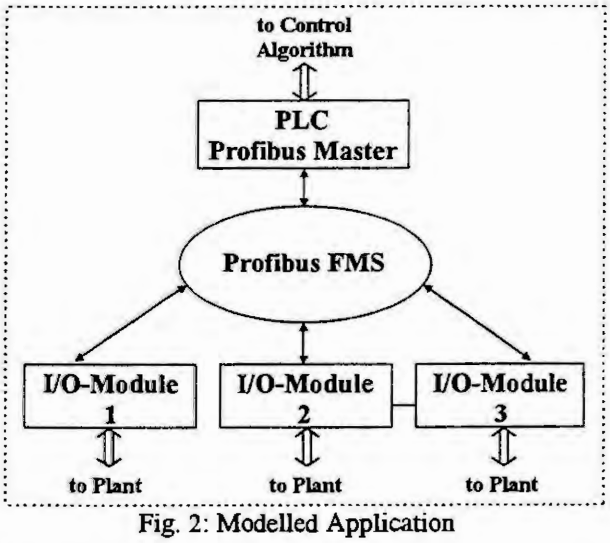


Fig. 2: Modelled Application

For the modelling of automation facilities it is useful to distinguish between the hardware architecture of the control devices on the one hand and the fieldbus system on the other hand. Both problems can be described with high level Petri Nets, like Coloured Petri Nets [13] or Predicate/Transition Nets [10]. Corresponding models of hardware architectures, like multiple processor systems [16], [17], [6], and models of communication systems [2], [3], [5], [9] or only layers [8], [11], are existing. However, for the judgement of the system efficiency it is necessary to choose a closed modelling approach for the whole automation facility.

The general modelling procedure should be shown by a small application that is illustrated in fig. 2. The automation facility may consist of a Profibus-FMS monomaster system with one PLC as master and three I/O-modules as slaves. The components 'Control Algorithm' and 'Plant' of the automation system are reduced to minimum necessary functions, as neither the controller nor the physical process are the main task of this research. Profibus is well established fieldbus system since several years and described in [1] and [7].

Therefor a class of extended Petri Nets developed by Dähler [4] is used. The extended Petri Nets belong to high level Petri Nets with individual tokens, which are described in an object oriented way, so that different attributes can be designed to every token. These attributes can be used as a condition for the activation of a transition and can be manipulated during the firing of transitions by a transition action code (SmallTalk-80). A hierachical net structure allows the development of functionalities in subnets as a transition refinement. The firing of transitions can be delayed in order to get quantitative reliable simulation results. A detailed function description of the simulator or of SmallTalk-80 respectively is given in [12] and [15]. Thus parallels to the controltechnical interpreted Petri Nets with timed transitions of [14] are getting distinct.

Development of the simulation model

While the simple I/O-modules have no own programming capacity and act only as slaves, they can be easily modelled as delayed transitions for every telegram type corresponding to the parameter 'Station Delay Responder' (T_{SDR}). The important and more complex component of this application is of course the PLC and Profibus master. A typical architecture not only limited to this application is a two processor system shown in fig. 3. While the main processor unit has several tasks, like control program, monitoring, ..., a single processor is only responsible for the communication protocol. The necessary data exchange between control program and communication interface is done by one of the main processor tasks, called 'Application Layer Interface' (ALI). This ALI is the user of the communication interface, here the Profibus interface. In this modelling it's simplified supposed, that the ALI is the third of three tasks of the main processor.

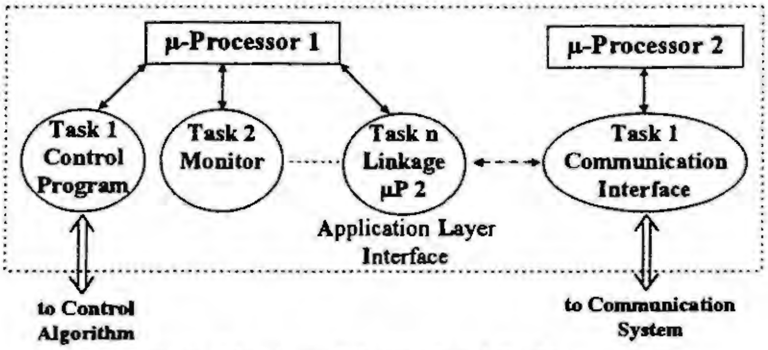


Fig. 3: Architecture of the PLC and Profibus master

For the modelling this controller functionality is described in a layered structure shown in fig. 4 and fig. 5, as the ISO/OSI-model does for communication interfaces. On the top level of this hierachy the multi task environment of the PLC main processor appears, acting as a Write-job generator for the ALI-task. The multi task environment defines the ALI cycle time and the ALI operating time. During its operating time the ALI produces automatically Read-jobs for all slaves. Write-jobs are produced in dependence of the program. These Read- or Write-jobs are handed over by a mailbox to the top-layer of the communication interface, the OSI-layer 7, FMS/LLI. In this application only two jobs can be put down in the mailbox. The FMS/LLI generates to every job a Read- or Write-request, which implies a relevant layer 7 operating time.

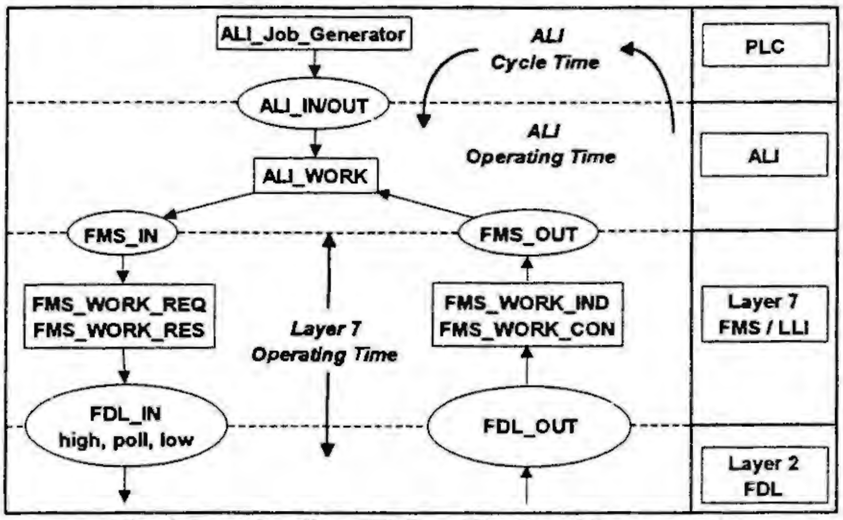


Fig. 4: Functionality of PLC and Profibus Master, part 1

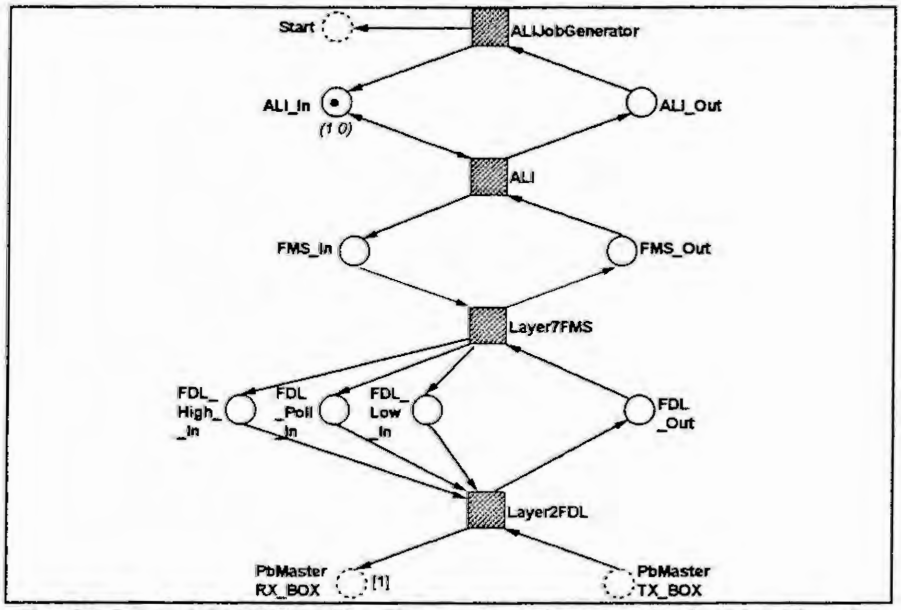


Fig. 6: Model of the PLC and Profibus master oriented at the function hierachy

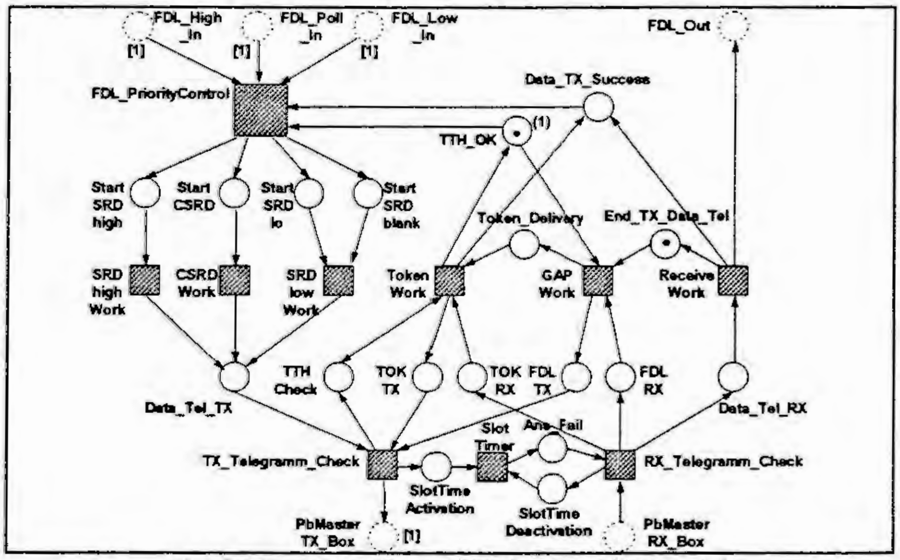


Fig. 7: The subnet 'Layer2FDL' as Petri Net model

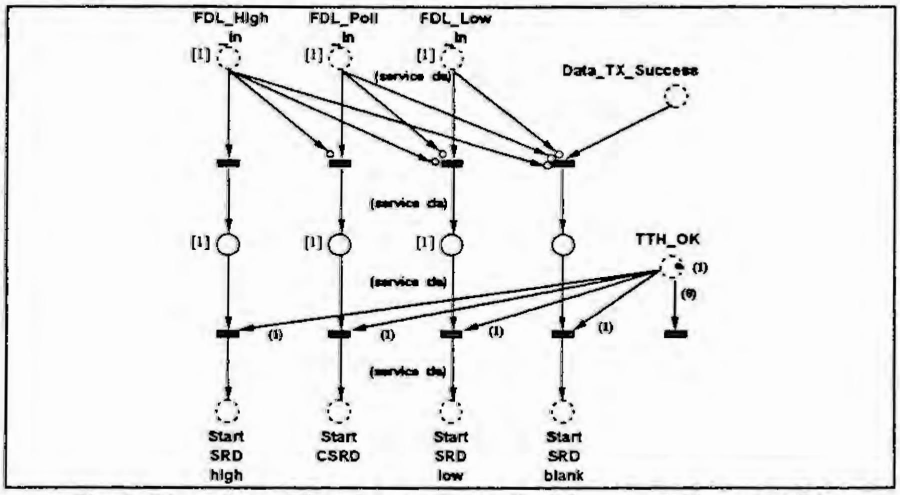


Fig. 8: The subnet 'FDL_PriorityControl' of the subnet 'Layer2FDL'

Verification of the simulation model

The verification of a simulation model for automation facilities should be lead by the needs of the users. An important parameter from their point of view for the efficiency of Automation systems is the response time of this system. It can be described by the execution time of a Write-service inside the automation facility, which means the time from activating a decentral remote output point by the PLC program to the changing of the physical level of this output point. This Write service execution time depends on the network configuration, the PLC-architecture, which means the ALI-parameters, the communication interface, especially the layer 7, the communication parameters and the actual condition of the communication interface. As in general for the user the variable parameters are the communication parameters, especially the 'Station Delay Initiator' - time and the Slottime, one kind of verification is the recording of the Execution time during varying the parameter T_{SDI} and T_{SL}. The slottime only affects on the execution time, if the addresses of the slaves do not succeed, as in this case the master always has to wait for the slottime, if a blank address is checked during the gap-update. Hence, in this paper only a verification of a succeeding address contribution by recording the execution time over an idletime interval should be shown in fig. 9. The dashed lines are measured in the application, the solid ones are simulated. Maximum, minimum and mean value of each 1000 Write services are shown.

Another important kind of verification is the analysis of the simulated communication protocol. This includes the correct telegram sequencing and the telegram distribution, which means the part of every telegram type in a certain time interval or telegram volume. This telegram distribution indicates e.g. the actual transmission capacity of the interface, as every blank telegram could have been replaced by a user-service, which means that the ALI is not working to capacity. The telegram distribution depends on the communication parameters, especially on the T_{SDI} parameter. It should be expected, that the less the value of the idletime, the more the part of the blank telegrams grow and the read telegrams go down, as a shorter T_{SDI} enables a faster finishing of a request/response procedure, while the ALI-cycle time keeps constant. Therefor, the fig. 10 shows the recording of the distribution during the same idletime interval as in fig. 9, again dashed lines measured in the application, solid lines simulated. The expected behaviour turns out in simulation and application in the same way.

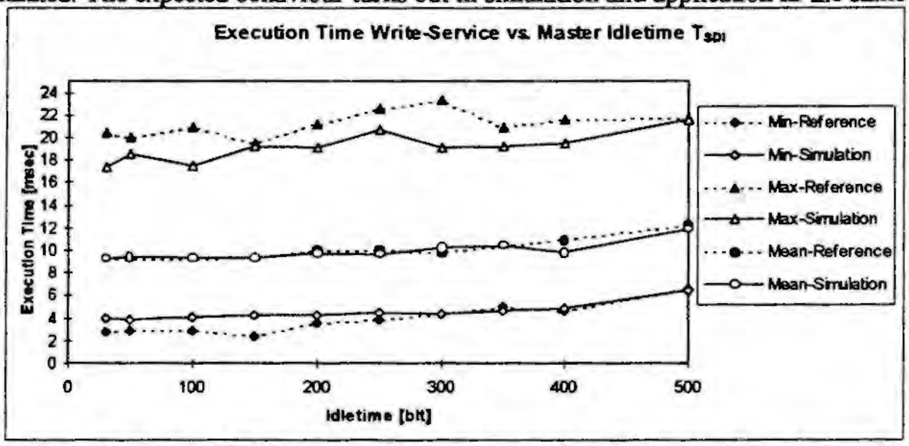


Fig. 9: Execution time of Write services in dependence of T_{SDI} for simulation and application

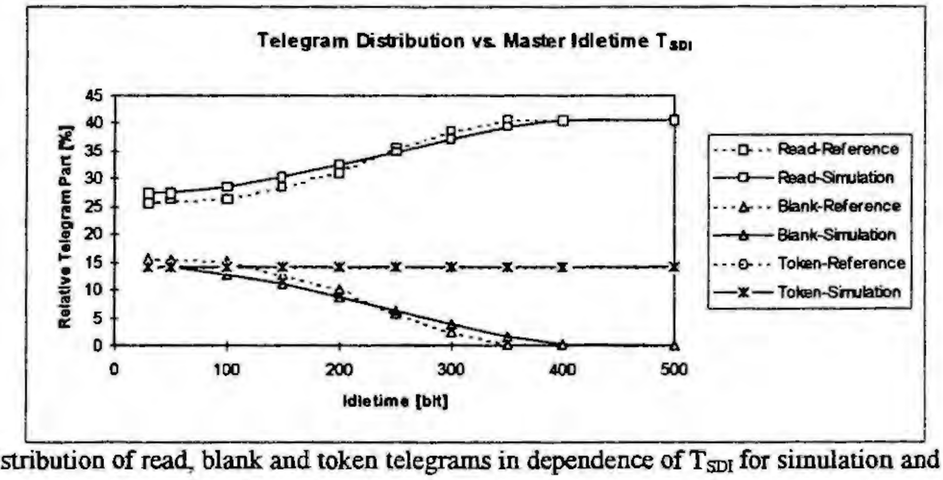


Fig. 10: Distribution of read, blank and token telegrams in dependence of T_{SDI} for simulation and application

Conclusion and Outlook

This Petri net model is able to simulate Profibus-FMS mono master applications of maximum 32 slaves in any address sequence considering not only the fieldbus system, but the whole automation facility. The assumed two processor system as Profibus master is a typical PLC-architecture not limited to this application. However, for different controller architectures it is of course necessary to modify the model. Due to the hierachical net structure such a modification is limited to the subnets ALI and ALIJobGenerator, as the model of the communication interface can be kept unchanged. Different hardware implementations of the communication interface mainly have to be considered in the parameter 'layer 7 operating time' of the model. The realized Profibus model only considers a Profibus subset, e.g. no SDA-connections or only Read- and Write-services. Again the hierachical net structure allows in this case an easy extension of the existing model, as it is oriented at single function blocks. Thus the Slottime could have been considered in an easy manner by supplementing a new submodel to the existing subnet Layer2FDL.

At all this simulation shows a good approximation of the application. The differences are caused mainly by simplifications of the multi tasking environment and the actually complex building of layer 7. However the simulation model only works with parameters, which are normally known by the user. Hence it offers the user and the manufacturer possibilities to analyse or improve the effectiveness of planned or existing automation systems only by simulation, while this normally is done by extensive tests and measurements of application engineers, which are in general more time-consuming and expensive as a simulation. As these results have met the approval of different users, the actual work is concentrated on a higher versatility of this model. That includes multi master networks and further communication systems, like Profibus-DP and CAN.

Acknowledgment. I would like to thank Dr. Mathias Jotter and his colleagues from the German manufacturer KUHNKE GmbH, who supported this work by generous providing of several control devices for this project.

References

1. Bender, K., PROFIBUS: the fieldbus for industrial automation, Hanser, München 1993
2. Chiola, Donatelli, Solda', Construction and Validation of a Petri Net Model of a Layered Protocol Architecture, In: Proc. TENCON 89, 4. IEEE Region 10 International Conference, 1989, p. 226-233
3. Chiola, Gaeta, Sereno, A Simulation Model of a Double Ring Protocoll based on Timed Well-Formed Coloured Petri Nets, In: Proc. MASCOTS 93, Int. Workshop on Modelling, Analysis and Simulation of Comput. and Telecommunication Syst. San Diego 1993, p. 259-264
4. Dähler, J., Ein Werkzeug für den Entwurf verteilter Systeme auf der Basis erweiterter Petri-Netze, Diss. an der ETH Zürich 1989
5. Diaz, M., Petri Net based Models in the Specification and Verification of Protocols, In: Petri Nets: Applications and Relationships to Other Models of Concurrency, (Eds.: Brauer, Reisig, Rozenberg) Lecture Notes in Computer Science 255, Springer 1986
6. Dicesare, F. et.al., Practice of Petri Nets in Manufacturing, Chapman & Hall, London, 1993
7. DIN 19245 Teil 1und 2 Profibus, Deutsches Institut für Normung e.V., 1991
8. Di Stefano, Mirabella, Evaluating the Field Bus Data Link Layer by a Petri Net-Based Simulation, IEEE Transactions on Industrial Electronics, Vol. 38 (4), 1991, p. 288-297
9. Funke, Machbarkeitsanalyse von Feldbusanwendungen, VDI-Fortschrittberichte 10 (222), VDI, 1992
10. Genrich, H. J., Predicate/Transition Nets, In: High-level Petri Nets, (Eds.: K. Jensen, G. Rozenberg), Springer, Berlin, 1991, p. 3-43
11. Hong, S. H. and Lee, S. G., Performance Analysis of the Data Link Layer in the IEC/ISA Fieldbus by Simulation Model, In: Proc. 5th IEEE Conference on Emerging Technologies and Factory Automation, ETFA, Kauai 1996, p. 593-601
12. IBE, PACE Benutzerhandbuch, Version 2.3, Glonn 1996
13. Jensen, K., Coloured Petri Nets: A High Level Language for System Design and Analysis, In: High-level Petri Nets, (Eds.: K. Jensen, G. Rozenberg), Springer, Berlin, 1991, p. 44-119
14. König, Quäck, Petri-Netze in der Steuerungs- und Digitaltechnik, Oldenbourg, 1988
15. LaLonde, W. R., Pugh, J. R., Inside SmallTalk, I, II, Prentice Hall, Englewood Cliffs, N.J., 1990
16. Peterson, J. L., Petri Net Theory And The Modeling Of Systems, Prentice Hall, Englewood Cliffs, N.J., 1981
17. Schnieder, E., Prozeßinformatik, Vieweg, Braunschweig, 1993

MODELLING AND SIMULATION OF SUPERVISORY PROCESS CONTROL SYSTEMS

G. Mušič and D. Matko
Faculty of Electrical Engineering, University of Ljubljana,
Tržaška 25, 1000 Ljubljana, Slovenia

Abstract. The control and supervision of a simple two reactor batch system is described in the paper. Continuous and discrete event model of the batch reactor are developed and a coordinating supervisor for the two reactors is designed. The overall system is simulated by the continuous simulation tool Matlab-Simulink which is enhanced for simulation of the sequential control logic represented by Sequential Function Chart.

Introduction

Modern industrial process control systems are most often implemented in a form of a multilevel distributed and hierarchical computer control system. The control hierarchy starts with various input and output devices at the lowest level and continues with the next, so called process control level where simple control functions are performed by loop controllers and programmable logic controllers. Process control level is followed by the supervisory level which consists of more sophisticated controllers or general purpose computers with graphical user interfaces and corresponding real-time communication links connected to the process control level.

Basic functions of the supervisory level are collecting of process data, display and archiving, communication with the operators and communication with the higher levels of the control hierarchy [7]. These are not direct control functions but supporting functions which are needed for efficient and comfortable process operation.

Beside these, more complex supervisory control functions can reside on this level. These include coordination of the low level control activities, scheduling of subprocess operations, various emergency scenarios, start-up and shut-down sequences. The implementation of such functions can significantly increase reliability, availability and flexibility of production process.

There is a lack of a generally accepted procedure for design of such supervisory systems. They are most often designed ad-hoc, using ingenuity of control engineers to solve the specific problem. If we look at the process control level, the situation is somewbat different. There exists a paradigm of the control design cycle which starts with modelling, continues by simulation, design of controller and verification by simulation before the control system is actually implemented in the real process. We believe that modelling and simulation can significantly contribute to more clear and comparable solutions in the field of supervisory control systems as well. For example, simulation can be applied to a non-formal verification of the designed systems. Formal verification of supervisory systems can be very difficult since no general formal analytical tools are currently available. Simulation can be effectively used to validate the performance of these systems, even if it can not replace the formal techniques.

The modelling of a supervisory process control system is a specific task. By the nature it is a pure discrete event system, which changes its discrete state as a reaction to external discrete events and performs external actions according to its state. But the object on which these actions are performed is generally a continuous or mixed continuous/discrete system. Part of discrete events which enter the supervisory system are generated by the evolution of the continuous state. We are therefore dealing with a hybrid overall system hich consists of a continuous part, a discrete part and an interface between them.

The supervisory system itself can be modelled as a discrete transition system such as finite automata or Petri net [3]. For the finite automata framework there exists a well established theory of supervisory control by Ramadge and Wonham [8]. In this theory the term supervisor is used in the sense of a discrete event controller. There have not been many attempts to apply this framework to the design of supervisory process control systems. The Petri net synthesis of supervisory controller has not been that extensively covered in literature although some contributions can be found [4, 11, 13]. On the other hand, Petri net models seem to be more easily understood by non-experts and yield a possihility of more intuitive design based on the knowledge about the process and not only purely on the rigid theoretical framework.

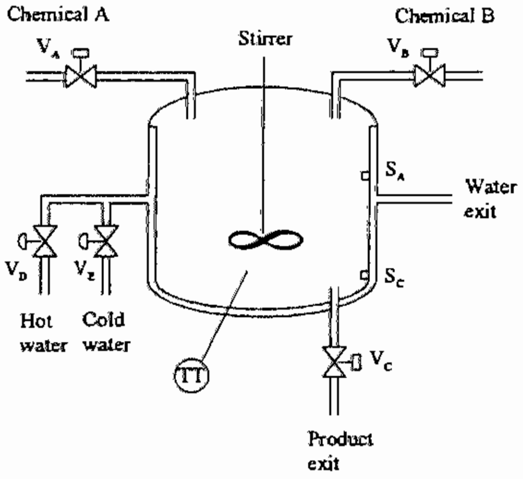


Figure 1: Batch reactor

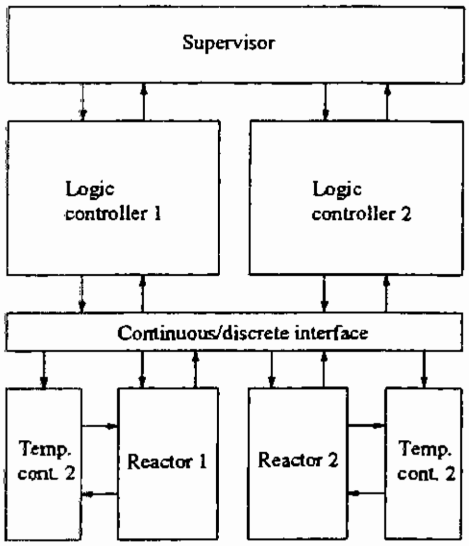


Figure 2: Model structure

The coordination is a task of a supervisory system. Similarly, in the phase of heating up the mixture to the reaction temperature a large amount of energy is required, while only a small flow of hot water is required afterwards to keep the mixture at the constant temperature. Therefore it is desired to coordinate the heating up phase between the two reactors as well.

Modelling

The described system is modelled as a combined continuous/discrete event system. The overall structure of the model is shown in Fig. 2.

The continuous part of the single reactor model consists of two subprocesses, describing the liquid level in the reactor and the temperature of the mixture in the reactor.

Liquid level is described by the Eqns. (1) and (2) which correspond to the filling and discharge phase, respectively.

$$A\frac{dh(t)}{dt} = (K_A\phi_A(t) + K_B\phi_B(t)) \tag{1}$$

$$A\frac{dh(t)}{dt} = -K_C\sqrt{2gh(t)} \tag{2}$$

In the given equations, $h(t)$ denotes liquid level, A is the transverse section of the reactor vessel, K_A, K_B and K_C are valve constants and ϕ_A and ϕ_B are the volume flows at the corresponding inlets.

Temperature is described by the following equation

$$m_{mix}c_{pmix}\frac{d\vartheta(t)}{dt} = \rho_w\phi_D(t)c_{pw}(\vartheta_D - \vartheta(t)) - \rho_w\phi_D(t)c_{pw}(\vartheta_E - \vartheta(t)) \tag{3}$$

where $\vartheta(t)$ is the temperature of the mixture, m_{mix} and c_{pmix} are the mass and the specific heat of the mixture, ρ_w and c_{pw} are the density and specific heat of the water, $\phi_D(t)$, ϑ_D, $\phi_E(t)$, ϑ_E are the water flows and temperatures of the incoming water at the hot water inlet and cold water inlet.

The temperature controller is a simple PI controller which is tuned to perform a response with no overshoot. The input to the controller is the temperature $\vartheta(t)$ while its output corresponds to ϕ_D when the output is positive and the output corresponds to ϕ_E when it is negative.

Upper part of the overall model in Fig. 2 represents the process as seen from the viewpoint of the logic controllers and the supervisory system and is therefore a discrete event system. It is modelled by Petri nets. The Petri net model of the reactor actually corresponds to the logic controller specification and can also be treated as a model of the logic controller.

The Petri net model of the logic controller is drawn on the basis of functional specification given in the previous section. The model of a controller for one reactor is shown in Fig. 3. The places and transitions are related to process sensors and actuators as seen from Tab. 1.

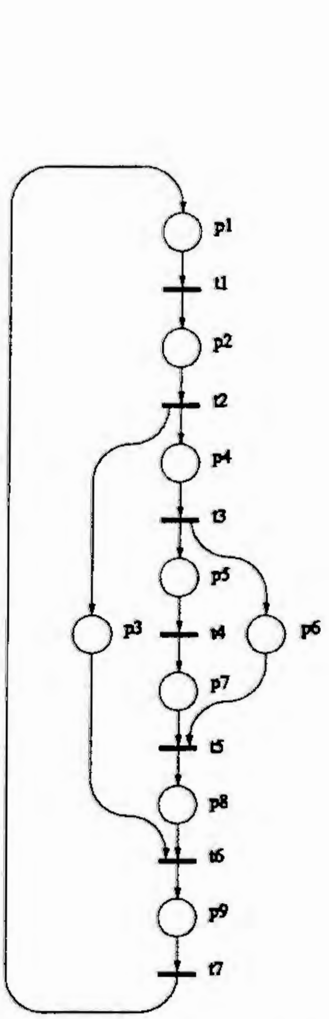


Figure 3: Petri net model of a batch reactor

Table 1: Control interpretation of places and transitions

Places	
p1	Initialisation
p2	V_A open
p3	Stirrer operating
p4	V_B open, Filling timer running
p5	Heating up to the setpoint
p6	Temperature controller enabled
p7	Reaction timer running
p8	V_E fully open
p9	V_C open

Transitions	
t1	Start of cycle
t2	S_A closed
t3	Filling timer run out
t4	Setpoint temperature reached
t5	Reaction timer run out
t6	Output temperature reached
t7	S_C open

The model of the supervisor is derived from the Petri net model by the method of place invariants [11] which results in the Petri net of the supervised system shown in Fig. 4. The supervisor consists of two places ($pc1$ and $pc2$) and the corresponding arcs are shown with dotted lines. The supervisor actually represents the classical mutual exclusion mechanism classified as parallel mutual exclusion in [13]. Note that for the proper operation of the overall system the interpretation of places $p2$, $p4$, $p11$ and $p13$ should be modified. For example, place $p11$ should be interpreted as: open V_A if S_A not closed. In the opposite case, the valve V_A of the reactor 2 could remain open even if the desired level has been reached when the coordinator enforces waiting for reactor 1 to finish with filling chemical B. Same could be achieved by splitting the place $p11$ into two places and inserting a new transition between them.

Between the continuous and discrete part of the overall model in Fig. 2 resides the interface which performs two tasks: it translates the binary controller outputs to the continuous manipulated variables of the process and it generates the state events, discrete events whose generation depends on the continuous state of the system.

Simulation

The overall system model can only be simulated if both the continuous and discrete event part are taken into account. In this section we describe the approach which is based on the existing general and widely spread simulation tool Matlab-Simulink.

In the simulation system the overall system structure presented in Fig. 2 is followed. The continuous process part is simulated by the existing continuous simulation block library. Both continuous subprocesses, level and temperature are simulated here and the temperature controller is included in the simulation scheme as well. The discrete control logic could be simulated by existing logic blocks in Simulink but this would lead to large and complex simulation schemes which are hard to follow.

We used a Sequential Function Chart (SFC) instead also referred as Grafcet. SFC is a part of the IEC standard IEC 1131.3 which defines languages for programming of the programmable logic controllers. It inherited many of its features from the theory of Petri nets [3]. More precisely, interpreted Petri net can be defined such that the input-output behaviour is the same as the input-output behaviour of the SFC. This enables a SFC to be directly redrawn from a Petri net model and the classical properties of Petri nets, such as marking invariants, can be applied also to SFC.

In order to implement a SFC graphic language in Simulink we defined a library of five new simulation blocks: SFC init, SFC step, SFC transition, SFC merge and SFC split. Because graphical representations such as Petri nets and Sequential Function Charts implicitly assume some feedback influence between consecutive nodes (e.g. when the step becomes active, the prevoius block is reset)it is difficult to convert such representations in a block scheme where the only influence among block is the signal flow through the directed connections between blocks. We had to implement a separate simulation mechanism for the SFC part of the simulation scheme which runs parallel to the standard ODE solver.

In order to achieve this, each SFC block is registered to a set of global variables during initialisation phase of the simulation. A special data structure is built which contains all the topological information about SFC, state of the SFC (status of the steps) and status of the transitions. A routine which calculates the new state of the chart is called once during each integration step.

The communication between continuous and discrete part consists of two parts. First part is the generation of the events which trigger the transition enabling conditions. State events are generated by comparison of the signals of interest to the specified threshold values. The 'Hit Crossing' block is used to decrease the integration step in the vicinity of the switching point and so increase the precision of the switching point location. The communication between the comparator blocks and the SFC transition blocks is performed via global variables. The result of each comparisons is assigned to a global variable and expressions built up of these variables and standard Matlab operators are assigned to transitions. During transition condition evaluation the expressions are evaluated as standard Matlab expressions and the result is interpreted as a Boolean value which enables or disables the particular transition.

The second part of the interface is the processing of the discrete actions. This is again performed through global variables which are assigned to each step of the SFC and are used in the continuous part of the simulation scheme as the input signals. These can operate switches or can be appropriately scaled and used as step shaped inputs.

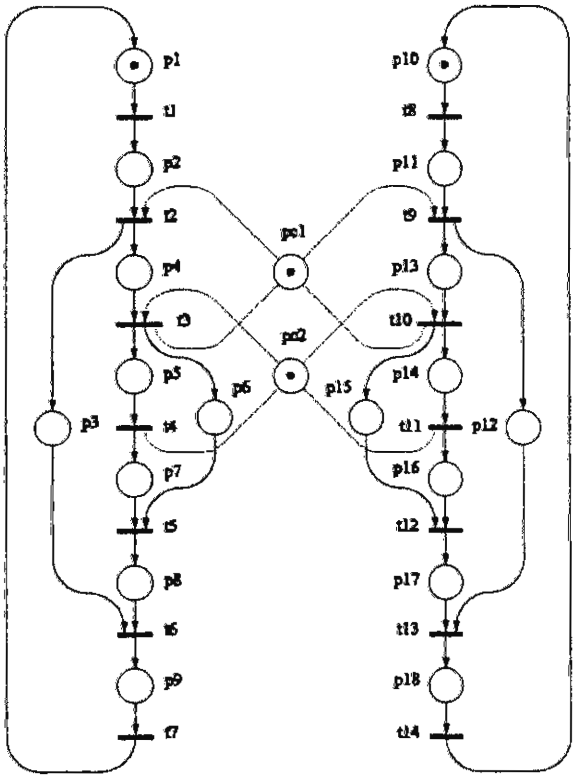


Figure 4: Petri net model of the overall system

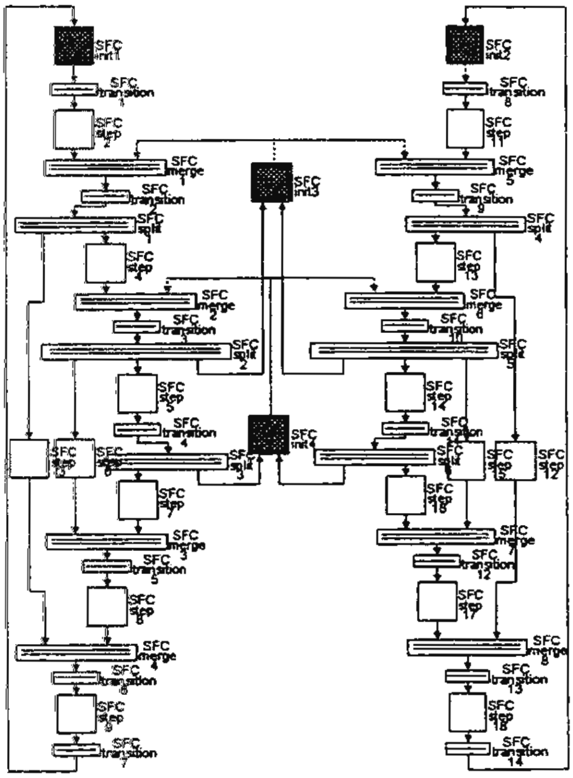


Figure 5: Simulation scheme of the discrete model

The designed simulation environment has been tested on the overall system model described in the previous section. The resulting simulation scheme of discrete part of the system is shown in Fig. 5. Different background color of certain steps indicate that these steps are active at the moment.

Conclusions

Continuous and discrete event modelling frameworks have been successfully applied to the modelling and simulation of batch system. It has been shown how the Petri nets can be employed in designing a supervisor and this is a straightforward approach to the real implementation in the programmable logic controller.

The Matlab-Simulink simulation environment was extended in a way to enable the simulation of sequential control logic. Sequential Function Chart library was designed and tested by the simulation of the modelled system. The integration methods used in Simulink do not provide any means of detecting the state events during simulation run. The only way to improve the accuracy of the simulation in the proximity of switching points is to use the 'Hit Crossing' block which forces the decrease of the integration step. Despite this the results of our simulation are satisfactorily. Namely, our purpose was to validate the functional correctness of the overall system and this can be done by the existing simulation method. The problem remains, however, in the computational inefficiency of such a simulation.

The open architecture of the Matlab-Simulink enables the use of self-written integration methods on the existing Simulink models and this could be one of the directions for further work.

References

1. Barton, P. I. and Pantelides, C. C., Modelling of combined discrete/continuous processes. AIChE, 40, 6 (1994), 996-979.

2. Cellier, F. E., Elmquist, H., Otter, M. and Taylor, J. H., Guidelines for Modelling and Simulation of Hybrid Systems. In: Proc. IFAC 12th Triennial World Congress, Vol. 8, Sydney, 1993, 391-397.

3. David, R. and Alla, H., Petri Nets for Modeling of Dynamic Systems - A Survey. Automatica, 30, 2 (1994), 175-202.

4. Holloway, L. E. and B. H. Krogh, Synthesis of feedback logic for a class of controlled Petri nets. IEEE Trans. Autom. Control, AC-35, 5 (1990), 514-523.

5. Jafari, M., Supervisory Control Specification and Synthesis. In: Petri Nets in Flexible and Agile Automation, (Ed.: Zhou, M.), Kluwer Academic Publishers, 1995, 337-368.

6. D. Matko, R. Karba, B. Zupančič, Simulation and Modelling of Continuous Systems, A Case Study Approach. Prentice Hall International, 1992.

7. Polke, M., Process Control Engineering. VCH Verlagsgesellschaft, Weinheim, 1994.

8. Ramadge, P. J. G. and Wonham, W. M., The Control of Discrete Event Systems. Proc. IEEE, 77, 1, (1989), 81-97.

9. SIMULINK, A program for Simulating Dynamic Systems, User's Guide. The Mathworks Inc., 1992.

10. Taylor, J. H. and D. Kebede, Modeling and simulation of hybrid systems in Matlab. In: Proc. IFAC 13th Triennial World Congress, Vol. J, San Francisco, 1996, 275-280.

11. Yamalidou, K., J. Moody, M. Lemmon and P. Antsaklis, Feedback Control of Petri Nets Based on Place Invariants. Automatica, 32, 1 (1996), 15-28.

12. Wölhalf, K., Fritz, M., Schultz, C. and Engell, S., BaSiP - Batch Process Simulation with Dynamically Reconfigured Process Dynamics. Computers Chem. Engng., 20, Suppl. (1996), S1281-S1286.

13. Zhou, M. C., F. DiCesare and D. L. Rudolph, Design and Implementation of a Petri Net Based Supervisor for a Flexible Manufacturing System. Automatica, 28, 6 (1992), 1199-1208.

A DECENTRALISED ALGORITHM TO DETERMINE INVARIANTS IN PETRI NETS

M. Boutayeb[1], A. Bourjij[1] and M. Darouach[1]

[1] University of Henri Poincaré - Nancy I - CRAN CNRS URA 821

186, rue de Lorraine, 54400 FRANCE. Email : boutayeb@iut-longwy.u-nancy.fr

Abstract

In this contribution, we propose a simple decentralised algorithm for computing invariants in large-scale interconnected Petri Nets. The main feature of the proposed technique lies in elaborating each subsystem's decision by using only the local incidence matrix and by the aid of an adjustment procedure ensures the global solutions taking the constraints interconnection into account. Consequently computational requirements, which are evaluated in terms of performed elementary operations, are reduced considerably in comparison with the global approach. To show performances of the proposed technique, a numerical example is provided in the last section.

1. Introduction

Petri Net has been one of the most frequently used tools for modelling, analysis and applications since its origins, about thirty years ago. It has been particularly used for representing computer systems to describe concurrency, conflicts synchronisation of processes etc. Furthermore, as a graphical tool Petri Net is well adapted to supervise dynamic systems in real time and thus to improve performances, such as the reliability and productivity of processes. It is not intended to give a total overview and summary of the theory and applications of Petri Nets, for more details the reader is referred to [4]-[5] and the references mentioned inside.

In this paper, we propose a decentralised algorithm for computing invariants in large-scale interconnected Petri Nets, the latter may be obtained from systems that are composed of geographically distributed systems. The method we put forward, has a great advantage is that computational requirements, to obtain the invariants, are reduced in comparison with the global approach. This property is particularly advantageous when large-scale time varying Petri Nets, or equivalently time varying incidence matrices, are considered. Indeed, many dynamic processes, such as distributed computer systems, are described by a time varying incidence matrix, which may be due to changes in the communication structure, or to changes in system configuration. Thus structural properties must be determined continually in order to enhance the performances of the process.

The main feature of the proposed architecture lies in elaborating each subsystem's decision by using only the local incidence matrix and with an adjustment procedure ensures the global solutions taking the constraints interconnection into account. In the proposed architecture, we may use any method to determine minimal support invariants of the subsystems.

In he last section, we discuss some aspects of the decentralised algorithm implementation and the computational requirements. One advantage of the above structure is the possibility to implement the proposed algorithm on a multi-processor environment where each subsystem is treated by a local processor. What is more, all local solutions are transmitted to a co-ordination processor to deduce the global solution.

Furthermore, we give an idea of the computational savings in the global and decentralised implementations of the proposed method. This is done in terms of performed elementary operations i.e. the total number of additions and multiplication, which give a good measure of these requirements. To show performances of the proposed technique, a numerical example is provided.

2. Problem formulation

Hereafter some basic definitions must be given. A Petri Net is a 4-tuple (P, T, F, M_0) where :

$P = \{P_1, ..., P_n\}$ is a finite set of places,

$T = \{T_1, ..., T_m\}$ is a finite set of transitions,

F is a binary relation which is represented by directed arcs between the places and transitions.

$M_0 : P \to N$ the initial marking of the places.

The mathematical model of a Petri Net may be described by the following equation :

$$M = M_0 + Ax \qquad 1$$

with $A = A^+ - A^-, A \in Z^{n.m}$ is the incidence matrix,

$A^+ = \left[a_{ij}^+\right]$ is an n.m integer matrix where a_{ij}^+ is the weight of the arc from transition T_j to place P_i.

$A^- = \left[a_{ij}^-\right]$ is an n.m integer matrix where a_{ij}^- is the weight of the arc from place P_i to transition T_j

and x is the firing count vector.

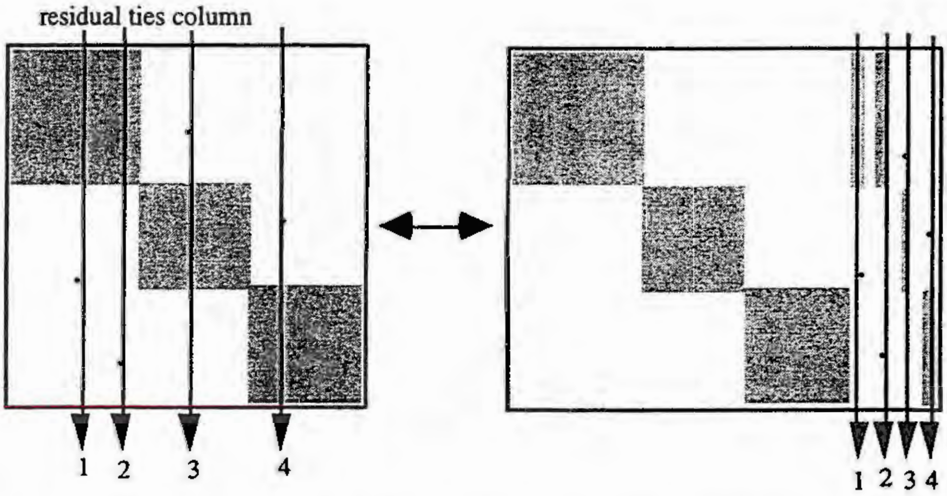
In this note, we address the problem of invariants determination in large scale interconnected Petri Nets. The associated mathematical model is of the form :

$$\begin{pmatrix} M_1 \\ . \\ . \\ M_p \end{pmatrix} = \begin{pmatrix} M_{01} \\ . \\ . \\ M_{0p} \end{pmatrix} + \begin{pmatrix} A_1 & 0 & . & 0 & A_{c1} \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & 0 & A_p & A_{cp} \end{pmatrix} \begin{pmatrix} x_1 \\ . \\ x_p \\ x_c \end{pmatrix} \qquad 2$$

where the connection between the subsystems (M_i, M_{0i}, A_i, x_i), i=1 ... p, is assured by (A_{ci}, x_c) with $A_i \in Z^{n_i.m_i}$ and $A_{ci} \in Z^{n_i.m_c}$. M_i and M_{0i} represent markings of the net related to the i^{th} sub-system. We are then concerned with vectors of positive integers so that :

$$\begin{pmatrix} A_1 & 0 & . & 0 & A_{c1} \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & 0 & A_p & A_{cp} \end{pmatrix} \begin{pmatrix} x_1 \\ . \\ x_p \\ x_c \end{pmatrix} = 0 \qquad 3$$

The obtained solutions are called T-invariants of the considered system. Vectors for which (3) is satisfied lead back to initial marking. Also P-invariants are defined as integer solutions of the transposed homogeneous equation. Naturally, there are several dynamic systems which are described by interconnected Petri Nets, but for the case where the global incidence matrix is general; several decomposition techniques exist like in [2] which transform the matrix A into a number of interconnected subsystems with local incidence matrices such as in equation (2). Coupling matrices A_{ci} ensure connection between M_i and M_P through x_C. In the following scheme, we give a simple decomposition technique for huge incidence matrix :



The obtained matrix is then composed with three independent local incidence matrices with coupling vectors 1, 2, 3 and 4. This technique will be applied to a numerical example where the Petri Net represents a communication protocol [1].

2. Main result

In this section we investigate the structure of the global incidence matrix to derive a new technique for determining invariants in interconnected Petri Nets in a decentralised way. Indeed, computing invariants by the global method may require too many computations and may even give inaccurate numerical results. Here we propose a simple and decentralised algorithm using separately and independently the local incidence matrices.
At first, let C_g denotes the invariants obtained by a global resolution :

$$C_g = \begin{pmatrix} x_{11} & & x_{1g} \\ \cdot & & \cdot \\ \cdot & \cdot\ \cdot\ \cdot & \cdot \\ x_{p1} & & x_{pg} \\ x_{c1} & & x_{cg} \end{pmatrix} = \begin{pmatrix} X_1 \\ \cdot \\ \cdot \\ X_p \\ X_c \end{pmatrix} \in N^{m.g} \qquad 4$$

where $X_k = \begin{pmatrix} x_{k1} & \cdot\ \cdot & x_{kg} \end{pmatrix} \in N^{m_k.g}$, $X_c = \begin{pmatrix} x_{c1} & \cdot\ \cdot & x_{cg} \end{pmatrix} \in N^{m_c.g}$ and $\begin{pmatrix} x_{1i} \\ \cdot \\ \cdot \\ x_{pi} \\ x_{ci} \end{pmatrix} \in N^m$ constitutes the i^{th},

i = 1, ..., g and k = 1, ..., p; vector solution of C_g which verifies:

$$\begin{pmatrix} A_1 & 0 & . & 0 & A_{c1} \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & 0 & A_p & A_{cp} \end{pmatrix} \begin{pmatrix} X_1 \\ . \\ . \\ X_p \\ X_c \end{pmatrix} = 0_{n.g} \qquad 5$$

with $m = m_1 + .. m_p + m_c$, rank $C_g = g$ and $x_{ji} \in N^{m_j}$ for $j = 1, ..., p$ and $i = 1, ..., g$.

From the structure of A, (4) is equivalent to:

$$\begin{pmatrix} A_k & A_{ck} \end{pmatrix} C_{gk} = 0 \quad \text{for } k = 1, ..p \qquad 6$$

with $$C_{gk} = \begin{pmatrix} X_k \\ X_c \end{pmatrix}$$

We notice that X_c in C_{gk} is a common solution matrix for all the subsystems. In fact, this represents the coupling constraints. The decentralised algorithm consists then in determining, at first, invariants of each k^{th} subsystem independently from the others and secondly by an adjustment procedure which takes the coupling constraints into account, we deduce the global solution.

We obtain then, for the kth subsystem, a local solution noted as follows :

$$C_k^d = \begin{pmatrix} X_k^d \\ X_{ck}^d \end{pmatrix} = \begin{pmatrix} x_{k1}^d & . & . & x_{kd_k}^d \\ x_{ck1}^d & & & x_{ckd_k}^d \end{pmatrix} \in N^{(m_j + m_c).d_k} \qquad \text{for } k = 1, .., p \qquad 7$$

with $$\begin{pmatrix} A_k & A_{ck} \end{pmatrix} C_k^d = 0 \quad \text{and} \quad \text{rank}(C_k^d) = d_k \qquad \text{for } k = 1, ..p \qquad 8$$

however, contrary to X_c which is a common solution matrix for all the subsystems so as equation 6 is satisfied, X_{ck}^d verifies only one constraint (8).

Thus, we obtain two solutions for the same local incidence matrix $\begin{pmatrix} A_k & A_{ck} \end{pmatrix}$ i.e. equations (6) and (8) k = 1, .., p. The first one is C_{gk} obtained by a global resolution where X_c is a common solution for all the subsystems. The second one is C_k^d obtained by a local resolution without taking coupling constraints into account. Therefore we have necessarily :

$$S_g \subseteq S_k^d \qquad \text{for } k = 1 ... p \qquad 9$$

where S_g and S_k^d are subspaces spanned by X_c and X_{ck}^d respectively.

The global solution S_g is then obtained by :

$$S_g = S_1^d \cap S_2^d ... \cap S_p^d \qquad 10$$

Once the matrix solution X_c is obtained from (10), we deduce $X_1, ..., X_p$ from (8) as :

$$X_k = - A_k^+ A_{ck} X_c \qquad 11$$

where A_k^+ is the pseudo-inverse matrix of A_k.

4. Study of computational requirements

In this paragraph, we discuss some aspects of the decentralised algorithm implementation and the computational requirements. In the proposed configuration, we have shown that each subsystem may be treated independently from the others when the global solution is obtained by a simple adjustment procedure. One advantage of the above structure is the possibility to implement the proposed algorithm on a multi-processor environment where each subsystem is treated by a local processor. After, all local solutions S_k^d are transmitted to a co-ordination processor to deduce X_c, X_k and then the updating matrix C_g.

In the following, we give an idea of the computational savings in the global and decentralised approaches. This is done in terms of elementary operations performed i.e. the total number of additions and multiplication, which give a good measure of these requirements.

For simplicity we consider that all the subsystems have the same dimensions, i. e. $n_i = m_j = m_c = m$, rank($\begin{bmatrix} A_i & A_{ci} \end{bmatrix}$) = m = q for i, j = 1, ..., p.

For multiplication we have:

a - Global method: $(p.q)^2.(p.q+r) + p.q$

b - Decentralised algorithm: $p.[2.q^3 + q^2.(r+m) + 2.q + q.m.r]$

For additions we have:

a - Global method: $p.q.(p.q - 1).[p.q+r-1]$

b - Decentralised algorithm: $p.[2.q.(q - 1).(q-1+m+r) + q.r.(m-1)]$

Clear that superiority of the decentralised method (linear in p) upon the global one (non linear p^3) is confirmed, particularly when the number of subsystems p increases significantly.

5. Numerical examples

The numerical example considered is a communication protocol as in [1]. The associated incidence matrix is :

$$A = \begin{pmatrix} -1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$

If we move columns 1, 2, 3, 6 and 7 at the end of A, we obtain an equivalent matrix with two interconnected sub-systems :

$$A = \begin{pmatrix} A_1 & 0 & A_{c1} \\ 0 & A_2 & A_{c2} \end{pmatrix}$$

with $A_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ -1 & -1 \\ 0 & -1 \\ -1 & 0 \end{pmatrix}$, $A_{c1} = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$ $A_2 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ -1 & -1 \\ 0 & -1 \\ -1 & 0 \end{pmatrix}$, $A_{c2} = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$

computation of the decentralised solutions leads to :

$$X_1^d = \begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \end{pmatrix},\; X_2^d = \begin{pmatrix} 0\,0\,1 \\ 0\,1\,0 \end{pmatrix} \text{ and } X_{c1}^d = X_{c2}^d = \begin{pmatrix} 1\,1\,1 \\ 1\,1\,0 \\ 0\,0\,1 \\ 0\,1\,1 \\ 1\,0\,0 \end{pmatrix}$$

therefore, from (10), we obtain : $X_c = \begin{pmatrix} 1\,1\,1 \\ 1\,1\,0 \\ 0\,0\,1 \\ 0\,1\,1 \\ 1\,0\,0 \end{pmatrix}$

In this example, we notice that since $X_{c1}^d = X_{c2}^d$, the solutions X_1 and X_2 are straightforward (without computation of (11)) i.e. $X_1 = X_1^d$ and $X_2 = X_2^d$.

6. Conclusion

In this paper, a decentralised method to compute invariants in interconnected Petri Nets is developed. The proposed architecture consists first to determine independently the invariants of each subsystem, this may be achieved by any method. In the second step, we deduce the common basis of X_{c1}^d, ... and X_{cp}^d and after we compute X_k by using equation (11). At the end of the paper we have discussed some aspects of the proposed decentralised algorithm implementation and the computational requirements. We have shown that the architecture presented reduce significantly the computational requirements, especially for a high number of interconnected subsystems and for a multi-processor environment.

References

1. Garg, K., An approach to performance specification of communication protocols using timed Petri Nets. IEEE Trans. on Soft. Engineering Vol. 11, N° 10, 1985, 1216-1224.
2. Lemonias, H. and Binder, Z., Decomposition approach for the job-shop scheduling problem. International Conference on CAD/CAM Robotics and Factories of the future, Norfolk Virginia U.S.A., 1990.
3. Martinez, J. and Silva, M., A simple and fast algorithm to obtain all invariants of a generalized Petri Net. 2nd European workshop on application and theory of Petri Net, Bod-honneff, 1981.
4 Murata, T., Petri Nets: properties, analysis and applications. Proceedings of IEEE Vol. 77, N° 4, 1989, 541-580.
5. Murata, M., Shenker, B. and Shatz, S. M., Detection of Ada deadlocks using Petri Net invariants" IEEE Trans. on Soft. Engineering Vol. 15, N° 3, 1989, 314-326.
6. Silva, M. and Colom, J; M., On the computation of structural synchronic invariants in P/T nets. Lecture Notes in Computer Science Vol. 340, 386-417, Springer-Verleg, 1988.

IDENTIFICATION OF DAMAGE IN REINFORCED CONCRETE STRUCTURES

Thomas Jahn
Universität Gesamthochschule Kassel
Kurt-Wolters-Str. 3, 34109 Kassel

Abstract

Structural damage in reinforced concrete structures usually causes a local reduction in stiffness. Often the defects are not visible on surface of the structures. Experimental vibration measurement makes it possible to judge the structure state. The degree of damage can be given by the definition of damage parameters. This article describes a procedure, which enables to identify the parameters by measured modal test data (eigenfrequencies and eigenvectors). The estimated starting parameters will be corrected by minimisation of differences between measured and corresponding analytical data.

1 Introduction

The objective of this work is to show that the measurement and identification of eigenfrequencies and eigenvectors can be used to draw conclusions about damage in reinforced concrete structures.

Damage in reinforced concrete structures can be, for example, a partial or complete break in the reinforcement. The causes of these defects are errors in construction planning, material defects or corrosion. The corrosion of steel is directly connected with cracks in the concrete.

As a result of loads, temperature or shrinkage, cracks reach such a size that carbon dioxide, water and oxygen penetrate the reinforcement. The area of protective alkaline milieu will be destroyed, and the reinforcement begins to rust.

The formation of cracks in reinforced concrete is unavoidable, so it is necessary to judge and limit the cracks' size with respect to the structure's appearance, the corrosion of the reinforcement and the permeability. As a rule, narrow cracks (≤ 0.3 mm) have no influence on the corrosion of the reinforcement. The term „damaging crack" can only be used when the cracks are so large that they restrict the usability or the carrying load of the concrete structure.

Cracks usually cause a local reduction in stiffness. This lowered stiffness changes the dynamic behaviour of concrete structures. This paper describes a procedure which solves the inverse problem to this observation: the use of dynamic test data (eigenfrequencies and eigenvectors) to judge the extent of crack damage. This procedure belongs to the group of indirect parameter correction methods.

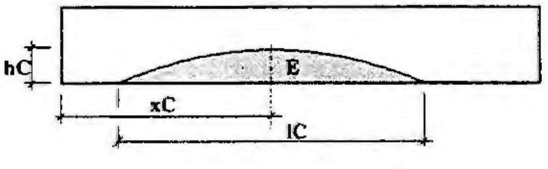
2 Identification of Parameters of the Crack Area

The area of damage is an area in which the assumedly homogeneous structure has a defect. This defect is caused by changing material or geometrical properties.

The starting point of the test is a reinforced concrete beam in a cracked state. The cracks are the defects of the assumedly homogeneous macrostructure.

Under the assumption that the crack area is dominated by bending cracks, it is approximated by a quadratic parabola. Then the area can be described by geometrical and material parameters.

The beam structure will be mapped into a finite element model, where the damage is modelled by the following parameters (Fig.(1)):



hC ... height of the crack area,
lC ... length of the crack area,
xC ... position of the crack area[1] and
E ... modulus of elasticity[2].

Fig.1: Parameters of the crack area

[1] position of the parabola peak,

[2] vertical to the cracks' direction.

2.1 Mathematical Basis for the Correction of the Parameters

The vector of the residuals R_w serves as the basis for correcting the parameters. In R_w the weighted errors from measured and calculated parameters are grouped together [1]:

$$\mathbf{R}_w = \hat{\mathbf{W}} \cdot \mathbf{R} = \hat{\mathbf{W}} \cdot (\mathbf{V} - \mathbf{V}^C(\mathbf{P})), \tag{1}$$

(**R** ... vector of residuals, **V** ... vector of measured values from the test model, $\mathbf{V}^C(\mathbf{P})$ ... vector of corresponding calculated values, **P** ... vector of the damage parameters, $\hat{\mathbf{W}}$ ... weighting matrix).

The parameters are estimated by the least square method:

$$\mathbf{J} = \mathbf{R}_w^T \cdot \mathbf{R}_w \qquad \rightarrow \qquad \text{Min}. \tag{2}$$

A minimization of J with respect to the residuals provides equations for calculating parameters. $V^C(P)$ in equation (1) is a nonlinear function of the damage parameters P. It is linearised by a Taylor series truncated after the linear term according to:

$$\mathbf{V}^C(\mathbf{P}) = \mathbf{V}_a + \mathbf{G} \cdot \Delta\mathbf{P}, \tag{3}$$

where $\quad \mathbf{V}_a = \mathbf{V}^C\Big|_{\mathbf{P}=\mathbf{P}_a}$ and $\mathbf{G} = \dfrac{\partial \mathbf{V}^C}{\partial \mathbf{P}}\Bigg|_{\mathbf{P}=\mathbf{P}_a}$,

(G ... sensitivity matrix, $\Delta\mathbf{P}$ ... parameter changes (=$\mathbf{P}$-$\mathbf{P}_a$), $\mathbf{P}_a$ ... linearisation point).

Then is: $\quad \mathbf{R}_W = \hat{\mathbf{W}} \cdot (\mathbf{V} - (\mathbf{V}_a + \mathbf{G} \cdot \Delta\mathbf{P}))$. (4)

The necessary conditions for minimizing equation (2) is:

$$\frac{\partial \mathbf{J}}{\partial \mathbf{P}} = \frac{\partial \mathbf{R}_w^T}{\partial \mathbf{P}} \cdot \mathbf{R}_w + \mathbf{R}_w^T \cdot \frac{\partial \mathbf{R}_w}{\partial \mathbf{P}} = 2 \cdot \frac{\partial \mathbf{R}_w^T}{\partial \mathbf{P}} \cdot \mathbf{R}_w = \mathbf{0}. \tag{5}$$

Substitution of equation (1) into equation (5) yields: $\quad -2 \cdot \dfrac{\partial \mathbf{V}^K(\mathbf{P})^T}{\partial \mathbf{P}} \cdot \mathbf{W} \cdot (\mathbf{V} - \mathbf{V}^K(\mathbf{P})) = \mathbf{0}$ (6)

where $\mathbf{W} = \hat{\mathbf{W}}^T \cdot \hat{\mathbf{W}}$.

The parameter variances ΔP are obtained from the equations (3) and (4):

$$\Delta\mathbf{P} = (\mathbf{G}^T \cdot \mathbf{W} \cdot \mathbf{G})^{-1} \cdot \mathbf{G}^T \cdot \mathbf{W} \cdot (\mathbf{V} - \mathbf{V}_a). \tag{7}$$

This equation is solved iteratively:

$$\mathbf{P}_{i+1} = \mathbf{P}_i + \Delta\mathbf{P}_i. \tag{8}$$

Using the differences between measured and calculated eigenvalues (λ, λ^C) and eigenvectors (**X**, $\mathbf{X}^C$), the residual vector in equation (1) is:

$$\mathbf{R} = (\lambda_1 - \lambda_1^C, \ldots, \lambda_n - \lambda_n^C, \mathbf{X}_1 - \mathbf{X}_1^C, \ldots, \mathbf{X}_n - \mathbf{X}_n^C). \tag{9}$$

The solution of the eigenvalue problem of the corrected analytical model produces the dynamic parameters (λ^C, $\mathbf{X}^C$):

$$(\mathbf{K}^C - \lambda_i^C \cdot \mathbf{M}^C) \cdot \mathbf{X}_i^C = \mathbf{0}, \tag{10}$$

(λ_i^C ... corrected eigenvalue, i=1,...,n, $\mathbf{X}_i^C$ ... corrected eigenvector, $\mathbf{M}^C$ ... corrected mass matrix, $\mathbf{K}^C$ ... corrected stiffness matrix).

Assuming that the mass matrix is constant (M=M^C) and independent of the parameter changes, then only a correction of the stiffness matrix is necessary. Corresponding to the equation (6) the partial derivatives of the residual vector with respect to the parameters P are:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{P}} = \left(\frac{\partial \lambda_1^C}{\partial \mathbf{P}}, \ldots, \frac{\partial \lambda_n^C}{\partial \mathbf{P}}, \frac{\partial \mathbf{X}_i^C}{\partial \mathbf{P}}, \ldots, \frac{\partial \mathbf{X}_n^C}{\partial \mathbf{P}} \right). \tag{11}$$

The partial derivatives of the eigenvalues λ^C and of the eigenvectors $\mathbf{X}^C$ (compare [2]) are:

$$\frac{\partial \lambda_i^C}{\partial \mathbf{P}} = \mathbf{X}_i^{CT} \cdot \frac{\partial \mathbf{K}^C}{\partial \mathbf{P}} \cdot \mathbf{X}_i^C, \qquad \frac{\partial \mathbf{X}_i^C}{\partial \mathbf{P}} = \sum_{\substack{i=1 \\ i \neq j}}^{n} \frac{\mathbf{X}_i^{CT} \cdot \dfrac{\partial \mathbf{K}^C}{\partial \mathbf{P}} \cdot \mathbf{X}_j^C}{\lambda_j^C - \lambda_i^C} \mathbf{X}_i^C. \tag{12),(13}$$

2.2 Assignment of Dynamic Test Data

An important requirement is to be able to compare experimental results with corresponding results obtained for the finite element model. If a unique assignment is not possible, additional information has to be obtained from the eigenvectors. The so-called MAC-value (Modal Assurance Criteria) in equation (14) [1] allows a judgement of the comparability of the modal data:

$$\mathrm{MAC} = \frac{(\mathbf{X}^{CT} \cdot \mathbf{X})^2}{(\mathbf{X}^{CT} \cdot \mathbf{X}^C) \cdot (\mathbf{X}^T \cdot \mathbf{X})} \cdot 100[\%], \tag{14}$$

($\mathbf{X}^C$ ... eigenvector of the analytical model, $\mathbf{X}$ ... eigenvector of the test model).

The value of the MAC is between 0 and 100%. A value of 100% means that the modes correspond exactly. As a rule the accordance of the compared vectors is acceptable, if MAC $\geq$ 80%.

3 Finite Element Model

The theoretical basis for the FE-Model is described in the literature [3]. Only the following important properties of the FE-Model used will be described here.

An isoparametric plan stress finite element (4-node) is used for modelling the concrete structure. A smeared crack model is used to represent the cracks in the beam [4]. Eilbracht and Link [5] identify geometrical parameters of the crack area by a discrete crack formulation.

The reinforcement is modelled by 2-node-bar elements. The bond between concrete and steel is assumed to be perfect (no relative displacements between concrete and steel).

The cracked reinforced concrete will be modelled by two material properties. As the forces and the displacements are very small, it is assumed that the material is linear elastic.

In the areas where the beam is uncracked, isotropic material behaviour is assumed in (15). The crack areas, orthotropic material behaviour is assumed in equation (16).

$$\begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy}\end{bmatrix} = \frac{E}{1-\nu^2}\begin{bmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix} \cdot \begin{bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \gamma_{xy}\end{bmatrix}, \qquad \begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy}\end{bmatrix} = \begin{bmatrix}E_{11} & E_{12} & 0\\ E_{21} & E_{22} & 0\\ 0 & 0 & E_{33}\end{bmatrix} \cdot \begin{bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \gamma_{xy}\end{bmatrix}, \tag{15),(16}$$

(σ_{xx}, ε_{xx} ... normal stresses and strains in x-direction, σ_{yy}, ε_{yy} ... normal stresses and strains in y-direction, σ_{xy}, ε_{xy} ... shear stresses and strains in x- and y-direction).

For steel linear elastic material behaviour is assumed:

$$\sigma_{xx} = E_s \cdot \varepsilon_{xx}, \tag{17}$$

(E_s ... elasticity modulus of steel, σ_{xx}, ε_{xx} ... stresses and strains in x-direction).

4 Examples of the Parameter Correction

The correction of parameters of a crack area is described by a mathematical algorithm [6] in the programming language [7].

The first example represents an adaptation of parameters using simulated test data. The FE-Model has been corrected by using the first six eigenfrequencies and eigenvectors of the test model. Table 1 gives the assumed simulated parameters of the crack area of the FE-Model and the estimated starting values.

Table 1

Parameter of crack area	**simulated**	**start values**
length of the crack area lC [m]	1.30	1.00
height of the crack area hC [m]	0.12	0.10
position of the crack area xC [m]	1.23	1.00
E-Modulus E [kN/mm²] (vertical to the crack direction)	20	25

The differences between the eigenfrequencies and the MAC-values prior to the 1st step and following the 12th (last) step are shown in Table 2. Fig. 2 presents the differences between frequencies, the MAC-values and the parameter correction process (with reference to the start values). The reason for the accurate determination of the defined crack parameters is that the modal data of the test model were free of errors.

Table 2: Differences between eigenfequencies and MAC-values

Iteration step	Number of Mode	Eigenfrequencies [Hz]		Differences [%]	MAC-values [%]
		test model	corrected model		
0	1	91.74	100.0	+9.00	99.95
	2	262.4	272.4	+3.81	99.88
	3	502.3	519.2	+3.36	99.80
	4	797.2	830.4	+4.16	99.57
	5	1139	1175	+3.16	99.70
	6	1512	1562	+3.31	99.50
12	1	91.74	91.74	0.00	100.00
	2	262.4	262.4	0.00	100.00
	3	502.3	502.3	0.00	100.00
	4	797.2	797.2	0.00	100.00
	5	1139	1139	0.00	100.00
	6	1512	1512	0.00	100.00

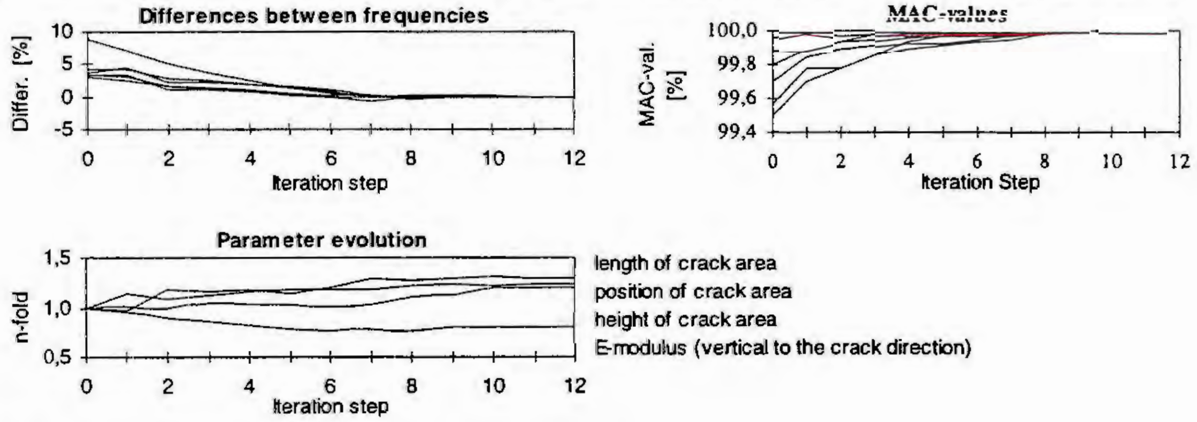


Fig. 2: Parameter correction process (simulated test data)

The second example demonstrates the parameter correction of a crack area on a rea,l reinforced concrete test beam. The crack area has been produced by a point load affected at a distance from the middle of the beam.

The dynamic behaviour of the undamped beam has been identified in the uncracked and cracked state using the computer program [8], [9]. In the dynamic tests both sides of the beam have been freely supported. The eigenfrequencies of the uncracked structure have been used to determine the average dynamical elasticity modulus of the concrete beam. The elasticity modulus (E=35.5 N/mm^2) has been used for the uncracked area of the beam during the identification of the parameters. The parameters of the crack area have been corrected by using the first five modes of the test model. Table 3 gives the estimated start and the corrected parameters (29th iteration step).

Table 3

Parameter of crack area	start values	corrected values
length of the crack area lC [m]	1.00	1.54
height of the crack area hC [m]	0.12	0.12
position of the crack area xC [m]	1.00	1.74
E-Modulus E [kN/mm²] (vertical to the crack direction)	30.0	17.1

The differences of the eigenfrequencies of the test and of the analytical models prior to the 1st step and following the last (29th) step are shown in Table 4.

A necessary condition for the identified parameters is the agreement of all modal responses of the test model with the results of the analytical model. The shaded modal responses of the test model (Table 4) have not been used to correct the parameters. These values agree very well with the calculated values. This test judges the quality of the identified parameters.

Fig. 3 shows the parameter variations with respect to the start values, the differences of frequencies and the MAC-values in every iteration step.

Fig. 4 represents the cracks in the tested beam and the adapted quadratic parabola (drawn according to scale) as a result of the parameter correction process.

Table 4: Differences between eigenfequencies and MAC-values

Iteration step	Number of Mode	Eigenfrequencies [Hz]		Differences [%]	MAC-values [%]
		test model	corrected model		
0	1	88.34	100.9	+14.22	99.46
	2	245.3	272.3	+11.01	99.02
	3	465.6	516.6	+10.95	98.47
	4	767.3	820.7	+6.96	97.53
	5	1085	1166	+7.47	97.98
	6	1434	1548	+7.95	97.17
	7	1813	1955	+7.83	95.65
29	1	88.34	88.46	+0.14	99.94
	2	245.3	249.4	+1.67	99.64
	3	465.6	473.3	+1.65	99.40
	4	767.3	755.9	-1.49	99.05
	5	1085	1090	+0.46	99.39
	6	1434	1452	+1.26	99.48
	7	1813	1845	+1.77	99.15

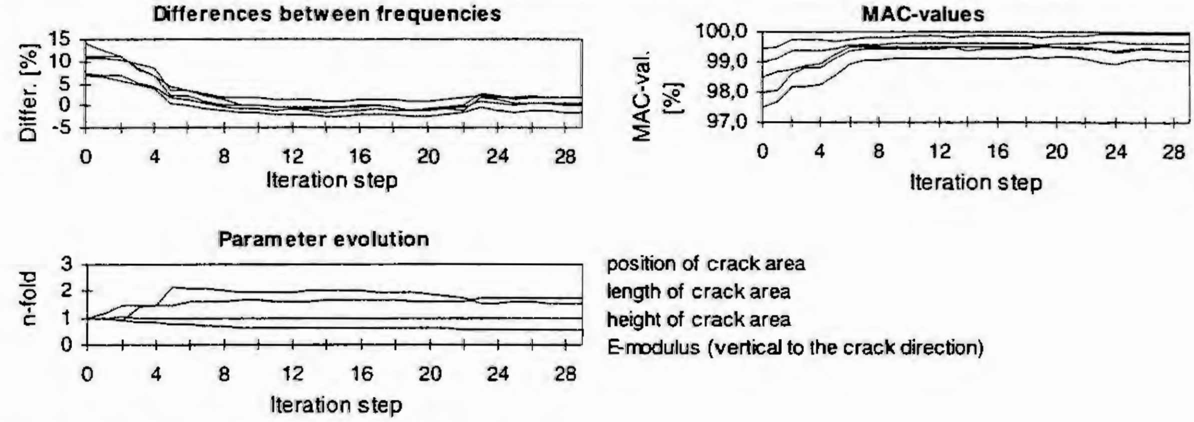


Fig. 3: Parameter correction process (real test data)

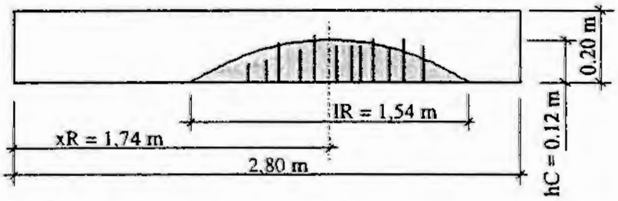


Fig. 4: Crack pattern and identified crack area

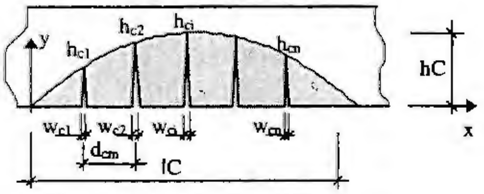
5 Analysis of the crack width $w_{c\,\max}$ in the parabolic crack area

It is possible to calculate approximately the maximum crack width $w_{c\,\max}$ by using the identified length of the crack area lC and the assumptions that the cracks' direction is vertical to the longitudinal axis of the beam, the distances between cracks d_{cm} are equal and the angles of all cracks α_i are identical [6]. This yields the expression:

$$w_{c\max} = \varepsilon_{c,lC} \cdot lC \cdot \frac{\left(\frac{nC+N}{2}\cdot d_{cm}\right)^2 - \frac{nC+N}{2}\cdot lC \cdot d_{cm}}{\sum\limits_{i=1}^{nC}((i \cdot d_{cm})^2 - i \cdot lC \cdot d_{cm})}, \qquad (18)$$

($\varepsilon_{c,lC}$ ... $\frac{\Delta lC}{lC}$ fictive strain over the length of the crack area, nC ... $\frac{lC}{d_{cm}} - 1$ number of cracks in the crack area, N ... $\frac{1-(-1)^{nC}}{2}$ coefficient by the consideration of an even or odd number of cracks).

For the tested beam the crack width (Table 5) has been calculated by using the identified length of the crack area and the fictive strain in the edge fibre calculated by FEM.



lC ... length of the crack area,
hC ... height of the crack area,
w_{ci} ... width of the i-th crack,
h_{ci} ... height of the i-th crack,
d_{cm} ... average distances of cracks.

Fig. 5: Assumed parabolic crack area

The analytical maximum crack width w_k has been determined by equation (19) [10]. The value β=1.7 gives the relation between the maximum and the average width of the cracks. This value is the result of statistical tests on the distance between the cracks under a certain load.

$$w_k = d_{cm} \cdot \varepsilon_{sm} \cdot \beta, \tag{19}$$

(d_{cm} ... average distance of cracks when crack formation is completed, ε_{sm} ... average strain in reinforcement, β ... relation between the calculated and the average value of the crack width).

Table 5: Comparison between the maximum crack width calculated by equations (18) and (19)

lC [m]	ΔlC [m]	nC	$w_{c\,max}$ [mm] equ. (18)	w_k [mm] equ. (19)
1.54	$1.86 \cdot 10^{-3}$	12	0.21	0.20

The comparison of the maximum crack widths shows a good agreement, because the relation between the maximum crack width (equ. (18)) and the average crack width converge to 1.5 increasing the number of cracks [6].

6 Conclusions and recommendations

Both examples of the parameter correction process demonstrate that it is possible to approximate the localisation and the determination of crack area dimensions by a quadratic parabola. A disadvantage of this method is that parameters of a parabolic crack area will always be adapted obtained, even if the crack area has another form. The results of this correction model do not necessarily describe the real shape of the area.

It has been also shown, that it is possible to calculate the maximum crack width using the identified parameters. The comparison with the calculated crack width by using a conventional method [10] confirms the results.

A recommended future research plan is to apply these experiments carried out under laboratory conditions on individual structure elements to more complete structures. Based on the identified damage it was possible to judge the carrying load and the life expectancy of the structure. In this research the following insolved problems still have to be examined:

- structures with multiple crack areas and other support conditions,
- the possibility to identify cracks produced by shear loads with the modal analysis methods and
- the estimation of the reliability of the identified results by means of statistical tests.

References:

1. Friswell, M.J., Mottershead, J.E., Finite Element Model Updating in Structural Dynamics, Kluwer Academic Publishers, Dordrecht 1995.
2. Natke, H.G., Einführung in die Theorie und Praxis der Zeitreihen und Modalanalyse, 3. Aufl., Vieweg Verlag, Braunschweig 1992.
3. Bathe, K.-J., Finite Element Procedures in Engineering Analysis, Prentice Hall, 1982.
4. Rashid, Y. R., Analysis of Prestressed Concrete Pressure Vessels, Nuclear Engineering and Design, Vol. 7 (1968).
5. Eilbracht, G., Link, M., Identification of Crack Parameters in Concrete Beams using Modal Test Data, International Symposium Non-Destructive Testing in Civil Engineering, Berlin 1995.
6. Jahn, T., Ein Beitrag zur Identifikation von Schädigungsparametern an Stahlbetonbauteilen, Dissertation, Universität Gesamthochschule Kassel 1996.
7. MATLAB, Math Works Inc., Interactive mathematical program.
8. ISSPA, Program system for the identification of modal parameters, FG Leichtbau, Universität Gesamthochschule Kassel.
9. Link, M., Qian, G., Identification of dynamic models using base excitation and measured reaction forces, Revue Française de Mécanique, n° 1994-1.
10. Eurocode 2, Teil 1: Planung von Stahlbeton- und Spannbetontragwerken, Beuth Verlag GmbH, Berlin, Juni 1992.

MODELLING OF A TWO LINK FLEXIBLE ROBOT USING THE MULTIBODY SYSTEM TOOLKIT M□BILE

W. Bernzen[1], **B. Riege**[1] **and S. Hartmann**[2]
University of Duisburg, Faculty of Mechanical Engineering, D-47048 Duisburg, Germany
[1] Department of Measurement and Control
[2] Department of Mechatronics
Email: {bernzen, riege}@uni-duisburg.de

Abstract. During the last decade robots with flexible links became a popular research object for control engineers. This is because of their sophisticated properties referring to feedback control, e. g. non-minimum phase behaviour in end effector control. Massive problems already occur trying to obtain an accurate analytic model for multilink flexible robots. This paper presents an effective way for numerical modelling of multilink flexible robots using the multibody program system M□BILE. The experimental model fitting to a laboratory test bed of a two link flexible robot is documented.

1 Introduction

Generally all mechanical systems are subject to deformation under loading and hence are compliant. If one speaks of rigid link robots, speed and load of these systems are as low as the assumption of rigid mechanical links is justified and compliance can be neglected. Increasing speed and/or payload demands for taking elasticity effects into consideration during control design. In [5] some basic investigations referring to the potential payoff in compliant arm control are given.

During the last decade robots with flexible links became a popular research object for control engineers [4]. In contrast to robots or manipulators with rigid links flexible or elastic links result in an extremely complex model especially for the multilink case. In [3], a schematic arbitrary formulation for the lagrangian dynamics of serial chains of flexible links and joints (multilink flexible robots) is given. However it has to be said that it is impossible to derive such a model without the aid of symbolic software packages like e. g. MAPLE®. Whereas for a single link flexible robot it is possible to obtain a linear model [6, 8] massive problems occur trying to obtain an accurate analytic model for multilink flexible robots as the extent of the equations grows extremely with each additional link.

This paper deals with the modelling and model fitting of a two link flexible robot driven by DC motors using the multibody program system M□BILE [10] which represents an easier way for modelling flexible robots. It has to be mentioned that this by no means results in an analytic model rather in a numerical model for simulation. The robot under consideration is a real laboratory test bed (fig. 1) and is illustrated in the second section. The third section explains the main mathematical aspects of the modelling using M□BILE specialized to beam modelling in the fourth section. Finally the experimental model fitting process with some results is presented.

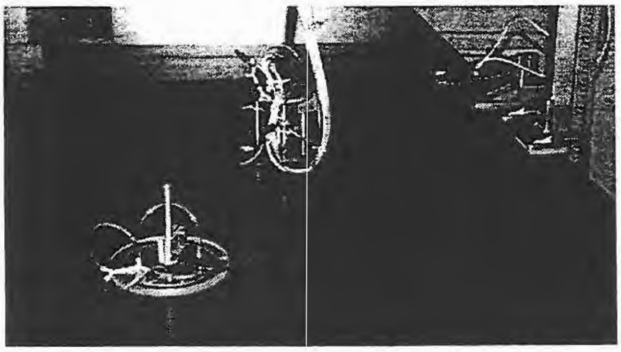


Figure 1: Laboratory test bed

2 Laboratory test bed of a two link flexible robot

The laboratory test bed of the two link flexible robot is depicted in fig. 1. The first joint on the right end is attached to a table with a smooth surface. The tip and the second joint are sliding on the table carried by two air bearings. Fig. 3 shows the air bearing carrying the end effector. DC motors with gear boxes actuate the joints and are controlled by a PC via D/A converters and amplifiers. To reach comparable results between the laboratory test bed and the model the integral character of the system has to be avoided by the use of proportional joint angle controllers for each joint. The links are manufactured of leaf spring steel to equip them with high elasticity.

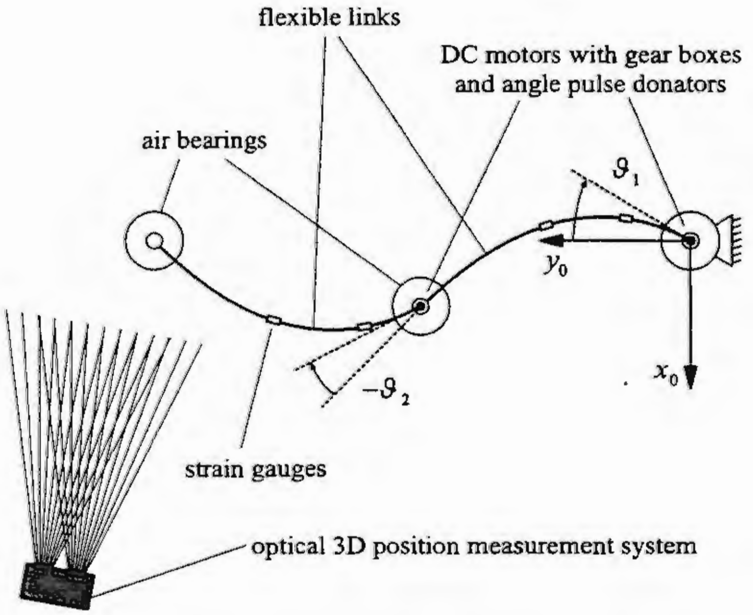

flexible links
DC motors with gear boxes and angle pulse donators
air bearings
ϑ_1
y_0
x_0
$-\vartheta_2$
strain gauges
optical 3D position measurement system

Figure 2: Schematic view of the laboratory testbed

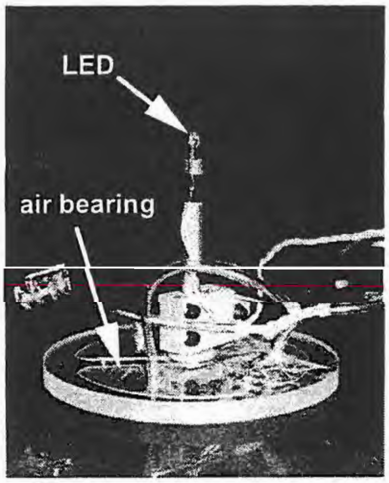


Figure 3: Air bearing of the end effector with LED

The schematic view in fig. 2 gives a general idea of the installed measurement devices and coordinates. Joint angles are measured by a PC TTL counter board counting incremental pulses from a pulse donator mounted on each DC motor. Strain gauges measuring the strain of the flexible links at some discrete locations are attached to several points of the links (two for each link). Their signals are amplified and transferred to the PC by an A/D converter. For measuring the end effector position in the x-y-plane an optical 3D-measurement system is installed. An infrared sensitive camera detects the light of two infrared LEDs attached to the second joint and the end effector. Fig. 3 shows the LED of the end effector. The measurement system transfers the x and y coordinates of the LEDs relatively to the reference frame (x_0, y_0) to the PC via the parallel port with an accuracy better than 1mm.

3 The multibody system toolkit MOBILE

The model is implemented with the help of the object-oriented toolset MOBILE where systems can be built up by assembling transmission elements which can be specialized to the so-called *kinetostatic transmission elements* in the case of mechanical components [10]. The underlying principle will be discussed in the following.

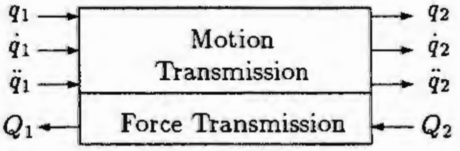


Figure 4: Kinetostatic transmission element

Each component of a system is described as an *object* of abstract mapping types. They transmit motion (q, $\dot{q}$, $\ddot{q}$) forward and forces (Q) backward (Fig. 4) between *state objects* (e. g. frames or state variables). The description of a complete dynamic system can be considered as a tree graph of arbitrary transmission elements where the kinematical information is transmitted from root to top and the force information including external, internal and inertial forces from top to root. By selecting special types of motion (position, velocity, and acceleration) and forces (e. g. inertial forces, coriolis forces) to transmit, the dynamic equations of motion can be built up subsequently. Therefore it is necessary to perform well-defined transmission steps with different acceleration inputs. For more details refer to [10].

4 Modelling of the flexible links

This section presents the basics for modelling the flexible links and the DC motors within the toolkit MOBILE. The flexible links of the manipulating system undergo large deformations in reality. Therefore a geometrically nonlinear beam element is used where the rigid body motion is superposed by the planar beam deflection. The latter is described naturally by a constant curvature κ such that the shear and extension strain vanish at the ends of the beam (Fig. 5). In the case of elastic bending κ and an applied bending torque T_b have the well-known relationship

$$T_b = \kappa E I \tag{1}$$

where E denotes Young's modulus and I the areal moment of inertia. The latter can be specified to a rectangular cross section by $I = \frac{1}{12}bh^3$ where b is the breadth and h the thickness of the beam. Furthermore the quadratic displacement field in fig. 6 using *Bernstein's* polynomials (e. g. [7]) approximates the displacement with given vectors $\underline{p}_0$, $\underline{p}_1$ and $\underline{p}_2$ of three nodepoints P_0, P_1 and P_2 related to the beam element through

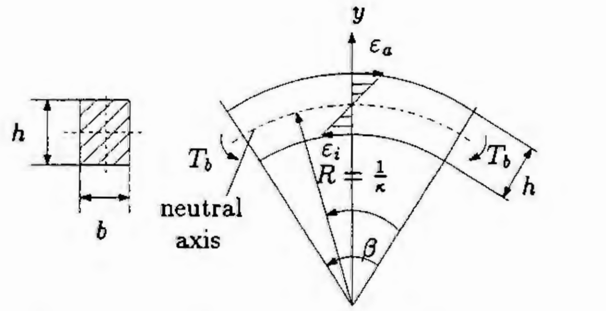


Figure 5: Bending behaviour of the beam element

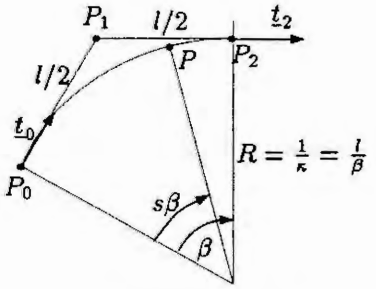


Figure 6: Quadratic displacement field

$$\underline{p}(s) = \sum_{i=1}^{2} \left[\binom{2}{i} s^i \, (1-s)^{2-i} \underline{p}_i \right] \; ; \; 0 \leq s \leq 1 \, . \tag{2}$$

Important geometric properties of this approach are the exact approximation of the positions $\underline{p}_0$, $\underline{p}_2$ and the tangent directions $\underline{t}_0$, $\underline{t}_2$ at both ends of the element which are expressed by the following relationships:

$$\begin{array}{l} \underline{p}_0 \; ; \; \underline{p}_1 = \underline{p}_0 + \frac{l}{2}\underline{t}_0 \; ; \; \underline{p}_2 = \underline{p}_1 + \frac{l}{2}\underline{t}_2 \\ \underline{p}(0) = \underline{p}_0 \; ; \; \underline{p}(1) = \underline{p}_2 \; ; \; \underline{t}_0 \, || \, \overline{P_0P_1} \; ; \; \underline{t}_2 \, || \, \overline{P_1P_2} \quad . \end{array} \tag{3}$$

As mentioned in section 3 the motion and force transmission has to be defined. With the variables in fig. 6 and the bending axis $\underline{u}$ ($\underline{u} \perp \underline{t}_0$; $\underline{u} \perp \underline{t}_2$) and the angular velocities $\underline{\omega}_0$ and $\underline{\omega}_2$ at P_0 and P_2 the kinematic transmission functions from "0" to "2" are

$$\left.\begin{array}{rcl} \underline{p}_2 & = & \underline{p}_0 + \frac{l}{2}(\underline{t}_0 + \underline{t}_2) \\ \underline{\omega}_2 & = & \underline{\omega}_0 + \beta\underline{u} \\ \dot{\underline{p}}_2 & = & \dot{\underline{p}}_0 + \frac{l}{2}(\dot{\underline{t}}_0 + \dot{\underline{t}}_2) \\ \dot{\underline{\omega}}_2 & = & \dot{\underline{\omega}}_0 + \ddot{\beta}\underline{u} + \dot{\beta}\dot{\underline{u}} \\ \ddot{\underline{p}}_2 & = & \ddot{\underline{p}}_0 + \frac{l}{2}(\ddot{\underline{t}}_0 + \ddot{\underline{t}}_2) \end{array}\right\} \text{ with } \left\{\begin{array}{rcl} \dot{\underline{t}}_0 & = & \underline{\omega}_0 \times \underline{t}_0 \\ \dot{\underline{t}}_2 & = & \underline{\omega}_2 \times \underline{t}_2 \\ \ddot{\underline{t}}_0 & = & \dot{\underline{\omega}}_0 \times \underline{t}_0 + \underline{\omega}_0 \times \dot{\underline{t}}_0 \\ \ddot{\underline{t}}_2 & = & \dot{\underline{\omega}}_2 \times \underline{t}_2 + \underline{\omega}_2 \times \dot{\underline{t}}_2 \\ \dot{\underline{u}} & = & \underline{\omega}_0 \times \underline{u} = \underline{\omega}_2 \times \underline{u} \; ; \; (\beta\underline{u} || \underline{u}) \quad . \end{array}\right. \tag{4}$$

The backward transmission function from "2" to "0" including external forces $\underline{F}$ and torques $\underline{T}$, inertial forces $\underline{F}_{inertial}$ and torques $\underline{T}_{inertial}$ and the internal force $Q_{internal}$ can be written as

$$\begin{aligned} \underline{F}_0 &= \underline{F}_2 && - \underline{F}_{inertial} \quad , \\ \underline{T}_0 &= \underline{T}_2 + \tfrac{l}{2}(\underline{t}_0 + \underline{t}_2) \times \underline{F}_2 && - \underline{T}_{inertial} \quad , \\ Q_{internal} &= -\frac{EI}{l}\beta = -\kappa EI \quad . \end{aligned} \tag{5}$$

Applying Simpson's rule for integrating the translational part of inertial properties and the trapezoidal rule for the rotational part over the length l of the beam element the lumped-mass description in fig. 7 is achieved (beam mass m, inertia tensor $\underline{\underline{\Theta}}$). With the acceleration terms at three points ($\underline{p}(s)$ with $s = 0, 0.5, 1.0$) the inertial forces of (5) can be computed. For more details refer to [9].

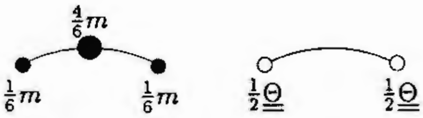

$\frac{4}{6}m$
$\frac{1}{6}m$
$\frac{1}{6}m$
$\frac{1}{2}\underline{\underline{\Theta}}$
$\frac{1}{2}\underline{\underline{\Theta}}$

Figure 7: Lumped masses due to displacement approximation

An examination of the convergence of the deflection while refining the mesh respectively increasing the number of beam elements shows that five elements are a good discretization for the system considered in this paper. The relative difference to results with an infinite number of elements is less than 1/1000. This results in the model depicted in fig. 8. In the case of pure bending the deflection would be calculated exactly for any discretization.

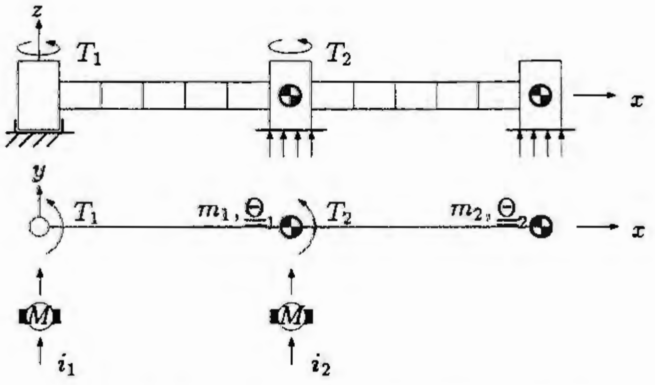

z
T_1
T_2
x
y
T_1
$m_1, \underline{\underline{\Theta}}_1$
T_2
$m_2, \underline{\underline{\Theta}}_2$
x
M
M
i_1
i_2

Figure 8: Simulation model of the manipulator

For simplification the DC motors equipped with proportional joint angle controllers are modelled approximately by a linear differential equation for the torque T

$$T = K_{\mathrm{Motor}} K_{\mathrm{P}} \left(\vartheta_{\mathrm{Set}} - \vartheta\right) - K_{\dot{\vartheta}} \dot{\vartheta} \quad , \tag{6}$$

that can be realized within MOBILE using a simple force element.

5 Model fitting and results

To fit the simulation model to the real process the physical and geometrical properties were taken from the construction drawings in a first step. After that several parameters were adjusted during experiments in the laboratory. The original test bed and the model are equipped with a proportional controller for the joint angles with the proportional gains $K_{P1} = K_{P2} = 5$. An example for experimental parameter adjustment is the joint friction of the DC motors. It is modelled according to fig. 9.

The following figures show a comparison between simulation results and test bed measurements for different quantities. The robot initial position in all plots is 0° for each joint which means that both

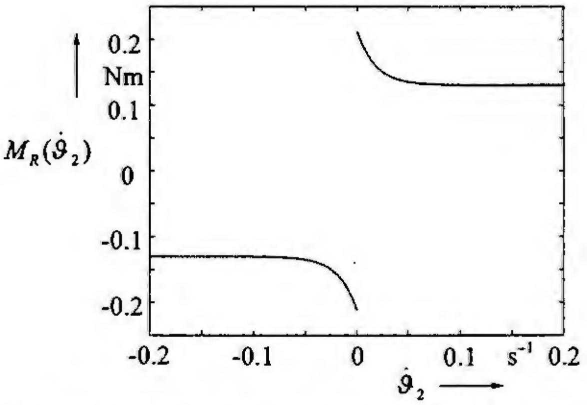


Figure 9: Friction torque for the second joint

links are in-line. The input is a step in the angle set value ϑ_{set} for the first joint of 20° and for the second joint of 30°.

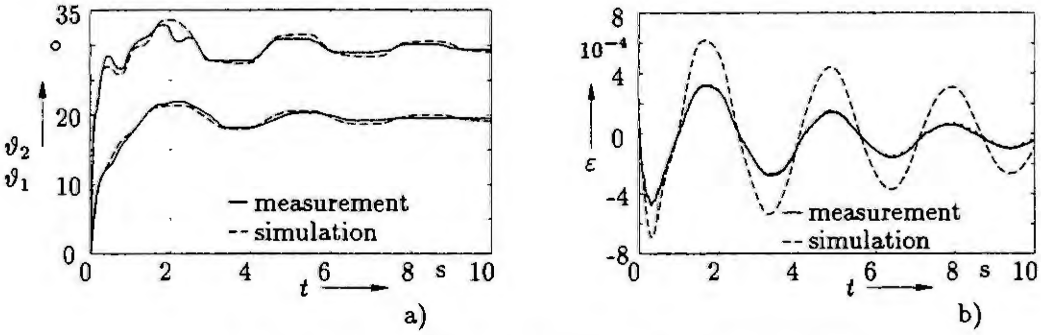


Figure 10: Joint angles (a) and strain in the first link (b)

The two joint angles resulting from the above mentioned inputs are depicted in fig. 10a. Especially for the first joint angle there is a very good correspondence between tbe test bed measurement and the simulation. Also tbe influence of the joint friction effects is obvious in both plots.

Fig. 10b shows the strain measured in the middle of the first link by strain gauges in comparison with the corresponding strain obtained by tbe model. Here is also a good correspondence between measurement and simulation except for the amplitude of the strain. The same effect is visible in fig. 11 where the x and y coordinates of the end effector measured by tbe optical position sensing system compared with the results obtained with MOBILE are depicted.

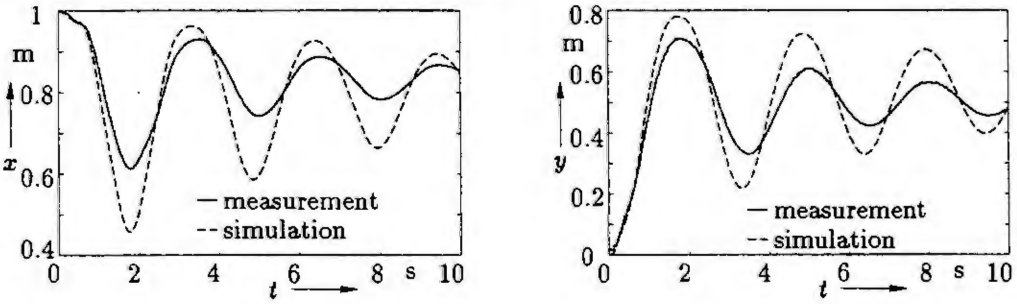


Figure 11: Absolute position of the end effector

The amplitude difference is due to the friction on tbe table on which tbe manipulator is moving carried by the air bearings that actually should be frictionless. As the test bed is not moving entirely without friction as it is supposed in the model this difference occurs. Tbis will be implemented in the model soon. But nevertheless the model qualitatively shows the same dynamic behaviour as the test bed.

6 Conclusion

Flexible robots became a challenging research object for control engineers during the last years. One basic problem in this field is the exact modelling of those flexible structures especially for the multilink case. Analytic modelling always results in extremely complex models. This paper presents the modelling and model fitting of a two serial link flexible robot driven by DC motors using the multibody system toolkit MOBILE. A numerical model is obtained and fitted to the test bed experimentally. The result is a rather exact and easy to handle simulation model which is very effective with respect to calculation time for simulation.

This model will be used for future investigations. It is suitable to test control concepts without using laboratory equipment. The model is easy to extend for modelling more complex flexible structures like, e. g., hydraulically driven flexible robots with closed kinematic loops. The models will be investigated further with respect to their structural properties [11]. They can be used to apply identification techniques to obtain an approximation based on the MOBILE-simulation [1, 2], without using the real system.

Acknowledgement. The authors would like to thank the student Dirk Nissing who carried out all the tedious measurements and simulations.

7 References

[1] W. Bernzen and G. Büdding. Modellbildungsmöglichkeiten elastischer Handhabungssysteme. Research Report 14/95, MSRT, University of Duisburg, 1995.

[2] W. Bernzen and H. Schwarz. Nonlinear approximation of nonlinear systems via linear identification and model combination. *Proc. of the 4th IEEE Mediterranian Symposium on New Directions in Control and Automation* Maleme, Crete, Greece, 59-64, 1996.

[3] W. J. Book. Recursive lagrangian dynamics of flexible manipulator arms. *The International Journal of Robotics Research*, 3:87–101, 1984.

[4] W. J. Book. Controlled motion in an elastic world. *ASME Journal of Dynamic Systems, Measurement and Control. 50th Anniversary Issue*, 115:252–261, 1993.

[5] W. J. Book and M. Majette. Controller design for flexible, distributed parameter mechanical arms via combined state space and frequency domain techniques. *ASME Journal of Dynamic Systems, Measurement and Control*, 105:245–254, 1983.

[6] C. Canudas de Wit, B. Siciliano, and G. Bastin. *Theory of Robot Control.* Springer, New York, 1996.

[7] P. Deuflhard and Andreas Hohmann. *Numerical Analysis: 1. A First Course in Scientific Computation.* de Gruyter, 1995.

[8] A. R. Fraser and R. W. Daniel. *Perturbation Techniques for flexible Manipulators.* Kluwer Academic Publishers, Boston, 1991.

[9] S. Hartmann, M. Anantharaman, and M. Hiller. Nichtlineare Balkenelemente für kinetostatische Beschreibungen der Dynamik von Mehrkörpersystemen. Submitted to *Zeitschrift für angewandte Mathematik und Mechanik*, 1997.

[10] A. Kecskeméthy. *Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen.* VDI Fortschritt-Berichte. Reihe 20. Vol. 88. VDI, Düsseldorf, 1993.

[11] B. Riege, W. Bernzen, and H. Schwarz. Controllability and observability of a one-link flexible robot. In *CSC'96 (Circuits, Systems and Computers)*, Hellenic Naval Academy, Athens, Greece, 1996.

FITTING MATHEMATICAL MODELS OF A PLANAR SERVO-PNEUMATIC TEST FACILITY TO LABORATORY EXPERIMENTS BY USING EXACT LINEARIZATION TECHNIQUES

F. Hecker[1] **and H. Hahn**[2]

[1] *Robert Bosch GmbH, K1-NB/ENB, Postfach 30 02 40, D-70442 Stuttgart*

[2] *Control Engineering and System Theory Group, Department of Mechanical Engineering (FB15),*
University of Kassel, D-34109 Kassel, Moenchebergstraße 7, Germany,
Phone +49 561-804 32 60, Fax +49 561-804 77 68, e-Mail: hahn@hrz.uni-kassel.de

Abstract. The task described in this paper is to fit mathematical models of a planar servo–pneumatic test facility to laboratory experiments. This has been done by using exact linearization techniques as evaluation criterion for judging the quality of the agreement between process model and process.

1 Introduction

Multi–axis test facilities are extensively used in industry and space-craft engineering for performing dynamic tests of critical components of machines ([1], [2]). Among those, hydraulic test facilities are used for testing heavy loads (in earthquake tests and space-craft tests) controlled by sinusoidal and transient test signals. Sinusoidal tests of both, heavy loads and small loads are also performed using electro–magnetic shakers. Transient tests of small loads are performed using servo–pneumatic test facilities. Due to the complex behavior of multi–axis test facilities, nonlinear control algorithms based on exact linearization techniques are used to achieve a good control behavior ([3]).

This paper describes how such exact linearization controllers can be used to fit computer simulations to laboratory experiments and to judge the degree of agreement between process model and process. This procedure has been applied to computer simulations and to laboratory experiments of a planar servo–pneumatic test facility.

In **Section 2** the hardware realization of the planar test facility will be briefly described. In **Section 3** nonlinear model equations of both, the servo–pneumatic actuators and the test facility and payload mechanics will be presented. In **Section 4** the model fitting process judged by time histories of computer simulations and of associated laboratory experiments will be discussed in several steps.

2 Laboratory experiment

The test facility used for performing the laboratory experiments includes the following subsystems (cf. 1):

- test table and payload (rigid bodies),
- three servo–pneumatic drives including specially designed pneumatic actuators with minimized friction and high response servo–valves,
- sensing elements including displacement sensors (recording the actuator piston displacements), acceleration sensors (recording the actuator piston accelerations), specially designed pressure sensors (recording the pressure directly in each actuator chamber), and force transducers (placed into the attachment point between each actuator and the test table).

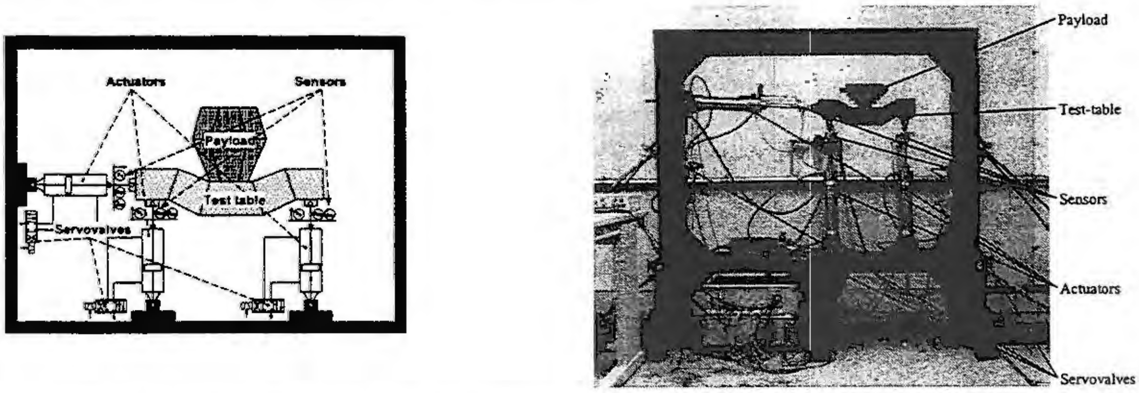


Figure 1: Computer animation graphics and photo of the planar test facility

The controller–hardware used to collect the measured data and to control the test-facility has been specially designed for implementing sophisticated control algorithms. It includes

- a Transputer based AD/DA card as a gateway to the real process, and
- a RISC-card with an i860 CPU as a number cruncher for complicated control algorithms. This RISC-card is connected to the Transputer-board via a communication Transputer and LINK-channels.

3 Mathematical models of the test facility

This section briefly describes the mathematical subsystem models of the test facility, a. o.

- linear model equations of the servo-valve dynamics,
- nonlinear model equations of the pressure evolution in each actuator chamber of the three pneumatic actuators, and
- linear and nonlinear models of the test-facility mechanics.

Mathematical models of the servo–pneumatic actuators

The mathematical model of the **servo–pneumatic actuators** used is described in detail in [4], included in this volume.

Mathematical models of the test facility mechanics

The mathematical models of the test facility mechanics used in this paper are

- **linear reduced model equations** of the test facility mechanics,
- **nonlinear reduced model equations** of the test facility mechanics ([3]), and
- **nonlinear extended model equations** of the test facility mechanics.

Linear reduced model of the test facility mechanics

The linear reduced model of the test facility mechanics has been derived on several assumptions from a nonlinear reduced model discussed in [3]. The linear model equations are

$$\bar{M} \cdot \ddot{\bar{x}} = \bar{F} - \bar{q}_G \qquad \in \mathbb{R}^3 \tag{1}$$

with

$$\ddot{\bar{x}} \;=\; \ddot{x} := (\; \ddot{x}_P^R \,,\, \ddot{z}_P^R \,,\, \ddot{\theta} \;)^T \qquad \text{(acceleration of the test table),} \tag{2}$$

$$\bar{F} \;:=\; T_d^T \cdot [A_K \cdot p_L - d_K \cdot T_d \cdot \dot{\bar{x}}] \qquad \text{(actuator forces),} \tag{3}$$

$$\bar{M} \;:=\; \begin{bmatrix} m \;, & 0 \;, & m \cdot z_{CP}^L \\ 0 \;, & m \;, & -m \cdot x_{CP}^L \\ m \cdot z_{CP}^L \;, & -m \cdot x_{CP}^L \;, & J_{Cy}^L + m \cdot [(x_{CP}^L)^2 + (z_{CP}^L)^2] \end{bmatrix} \qquad \text{(inertia matrix),} \tag{4}$$

$$\bar{q}_G \;:=\; \begin{pmatrix} 0 \\ -m \cdot g \\ m \cdot g \cdot x_{CP}^L \end{pmatrix} \qquad \text{(vector of gravitational forces and torque), and} \tag{5}$$

$$T_d \;:=\; \bar{J}_x(x) = \begin{bmatrix} 1\,,0\,, & z_{P1P}^L \\ 0\,,1\,, & -x_{P2P}^L \\ 0\,,1\,, & -x_{P3P}^L \end{bmatrix} \qquad \text{(transformation matrix mapping system variables from actuator housing fixed frames } K_i \text{ to test table fixed frame } L \text{ (cf. 2)).} \tag{6}$$

Nonlinear extended model of the test facility mechanics

The nonlinear reduced model equations of the test-facility mechanics described in [3] have been derived omitting the inertia properties of the actuators. In this section the mechanical behavior of the actuators

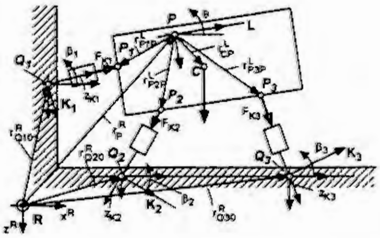


Figure 2: Scheme of the planar test facility

is included in the model equations. Each actuator is divided into two rigid bodies (actuator housing and actuator piston). This yields to nonlinear extended model equations, describing the planar motion of seven rigid bodies. Compactly written, the nonlinear extended model equations are (compare [5]):

$$\tilde{M}(x) \cdot \ddot{x} + \tilde{q}_{Fl}(\dot{x},x) + \tilde{q}_G(x) = J_x^T \cdot F \qquad \in \mathbb{R}^3 \tag{7}$$

with a common inertia matrix $\tilde{M}(x)$, describing the various inertia properties of seven rigid bodies

$$\tilde{M}(x) := M(x) - J_x^T(x) \cdot \left[-M_K \cdot J_x + T_{qn} \cdot \left(J_{ges} \cdot Z_K^{-1} \cdot J_\beta + M_K \cdot R_{CK^2} \cdot Z_K^{-1}\right)\right] \ , \tag{8}$$

with a nonlinear vector $q_{FL}(\dot{x},x)$, describing the various centrifugal forces

$$\begin{aligned}\tilde{q}_{FL}(\dot{x},x) := q_{FL}(\dot{x},x) - J_x^T(x) \cdot \Big[&-M_K \frac{\partial J_x}{\partial t}\dot{x} + M_K R_{CK} \dot{\boldsymbol{\beta}}^2 + T_{qn} J_{ges} Z_K^{-1} \frac{\partial J_\beta}{\partial t}\dot{x} \\ &+2T_{qn} M_K \dot{Z}_K \operatorname{diag}(\dot{\beta}) Z_K^{-1} r_{CK} + T_{qn} M_K R_{CK^2} Z_K^{-1} \frac{\partial J_\beta}{\partial t}\dot{x}\Big] \ ,\end{aligned} \tag{9}$$

and with a vector $\tilde{q}_G(x)$, including the gravitational forces and torque

$$\tilde{q}_G(x) := q_G(x) - J_x^T(x) \cdot \left[\mathrm{g} \cdot M_K \cdot K_2 + \mathrm{g} \cdot T_{qn} \cdot M_{KZ} \cdot K_1 \cdot Z_K^{-1} \cdot r_{Cges}\right] \ . \tag{10}$$

The matrix $M(x)$ (common inertia Matrix) as well as the vectors $q_{FL}(\dot{x},x)$ (vector of centrifugal forces), $q_G(x)$ (vector of gravitational forces) and F (vector of actuator forces) belong to the nonlinear reduced model described in [3]. In addition the equations (8) to (10) include the matrices and vectors

$$\begin{aligned}
Z_K &:= \operatorname{diag}(z_{K1}, z_{K2}, z_{K3}) \ , & \dot{Z}_K &:= \operatorname{diag}(\dot{z}_{K1}, \dot{z}_{K2}, \dot{z}_{K3}) \ , \\
R_{CK} &:= \operatorname{diag}(r_{CK_1}, r_{CK_2}, r_{CK_3}) \ , & R_{CK^2} &:= \operatorname{diag}\left(r_{CK_1}^2, -r_{CK_2}^2, -r_{CK_3}^2\right) \ , \\
r_{CK} &:= \left(r_{CK_1}, -r_{CK_2}, -r_{CK_3}\right)^T \ , & r_{Cges} &:= \left(r_{Cges_1}, -r_{Cges_2}, -r_{Cges_3}\right)^T \ , \\
J_{ges} &:= \operatorname{diag}(J_{ges_1}, -J_{ges_2}, -J_{ges_3}) \ , & T_{qn} &:= T_b^{-1} \cdot T_a \ , \\
M_K &:= \operatorname{diag}(m_{K_1}, m_{K_2}, m_{K_3}) \ , & M_{KZ} &:= \operatorname{diag}(m_1, m_2, m_3) \ , \\
K_1 &:= (-\sin\beta_1, \cos\beta_2, \cos\beta_3)^T \ , & K_2 &:= \operatorname{diag}(\cos\beta_1, \sin\beta_2, \sin\beta_3) \ , \\
\dot{\boldsymbol{\beta}}^2 &:= \left(\dot{\beta}_1^2, \dot{\beta}_2^2, \dot{\beta}_3^2\right)^T \ , & \operatorname{diag}\left(\dot{\beta}\right) &:= \operatorname{diag}\left(\dot{\beta}_1, \dot{\beta}_3, \dot{\beta}_3\right) \ ,
\end{aligned}$$

with $(i = 1,2,3)$

m_{K_i} mass of the actuator piston i,

m_{Z_i} mass of the actuator housing i,

$m_i := m_{K_i} + m_{Z_i}$ common mass of the actuator piston and actuator housing,

r_{CK0_i} distance between the coupling point of the actuator P_i and the center of gravity C_{K_i} of the actuator piston (for actuator piston displacement $z_{K_i} = 0$),

$r_{CK_i} := z_{K_i} + r_{CK0_i}$ distance from P_i to C_{K_i}, depending on the actuator piston displacement z_{K_i},

r_{CZ_i} distance between P_i and the center of gravity C_{Z_i} of the actuator housing,

$r_{Cges_i} := \frac{m_{K_i} \cdot r_{CK_i} + m_{Z_i} \cdot r_{CZ_i}}{m_i}$ distance between P_i and the center of gravity C_{ges_i} of the common body actuator housing and actuator piston, depending on the actuator piston displacement z_{K_i},

J_{CK_iy} moment of inertia of the actuator piston i with respect to C_{K_i},

J_{CZ_iy} moment of inertia of the actuator housing i with respect to C_{Z_i},

$J_{ges_i} := \left[J_{CK_iy} + m_{K_i} \cdot r_{CK_i}^2 + J_{CK_iy} + m_{K_i} \cdot r_{CK_i}^2\right]$ moment of inertia of the whole actuator (actuator housing and piston) with respect to P_i,

T_a, T_b factor matrices of T_{qn}, mapping the transversal forces of the actuators to normal forces, with

$$T_a := \begin{bmatrix} \sin\beta_1 & \cos\beta_1 & 0 \\ \cos\beta_2 & -\sin\beta_2 & \begin{array}{r}\left(z_{Q20}^R + A_2 - z_{Q10}^R - A_1\right) \cdot \cos\beta_2 \\ + \left(z_{Q20}^R + B_2 - z_{Q10}^R - B_1\right) \cdot \sin\beta_2\end{array} \\ \cos\beta_3 & -\sin\beta_3 & \begin{array}{r}\left(z_{Q30}^R + A_3 - z_{Q10}^R - A_1\right) \cdot \cos\beta_2 \\ + \left(z_{Q30}^R + B_3 - z_{Q10}^R - B_1\right) \cdot \sin\beta_3\end{array} \end{bmatrix}^T \ , \quad T_b := \begin{bmatrix} \cos\beta_1 & -\sin\beta_1 & 0 \\ \sin\beta_2 & \cos\beta_2 & \begin{array}{r}\left(z_{Q20}^R + A_2 - z_{Q10}^R - A_1\right) \cdot \sin\beta_2 \\ - \left(z_{Q20}^R + B_2 - z_{Q10}^R - B_1\right) \cdot \cos\beta_2\end{array} \\ \sin\beta_3 & \cos\beta_3 & \begin{array}{r}\left(z_{Q30}^R + A_3 - z_{Q10}^R - A_1\right) \cdot \sin\beta_3 \\ - \left(z_{Q30}^R + B_3 - z_{Q10}^R - B_1\right) \cdot \cos\beta_3\end{array} \end{bmatrix}^T \ , \tag{11}$$

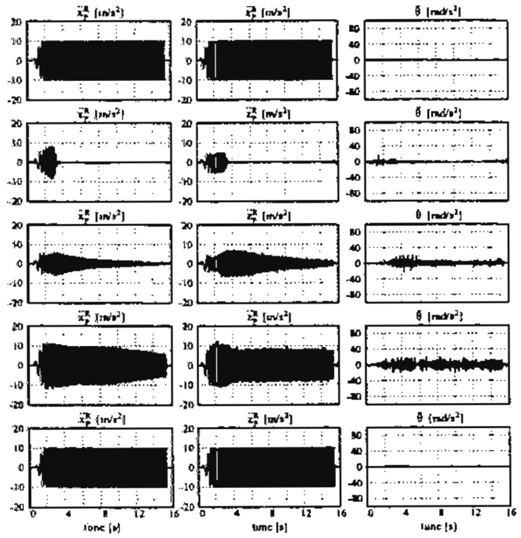
J_β nonlinear jakobian matrix of the orientation angles of the actuators β_i, with

$$J_\beta := \frac{\partial \beta}{\partial x} := \begin{bmatrix} \frac{\partial \beta_1}{\partial x_p^R} & \frac{\partial \beta_1}{\partial z_p^R} & \frac{\partial \beta_1}{\partial \theta} \\ \frac{\partial \beta_2}{\partial x_p^R} & \frac{\partial \beta_2}{\partial z_p^R} & \frac{\partial \beta_2}{\partial \theta} \\ \frac{\partial \beta_3}{\partial x_p^R} & \frac{\partial \beta_3}{\partial z_p^R} & \frac{\partial \beta_3}{\partial \theta} \end{bmatrix} . \tag{12}$$

4 Fitting mathematical models to the laboratory experiment by using exact linearization techniques

In this section mathematical models of a planar servo–pneumatic test facility described in Section 3 are fitted to laboratory experiments using exact linearization techniques. The **idea of this approach** is:

- Sine sweep signals of constant amplitude of the form shown in Figure 3a are characterized by the fact
  - that they have an extreme **simple geometrical pattern** with straight lines as envelopes, and
  - that they include information of a **wide frequency range**.
- Stimulating dynamical systems by sine sweeps provides outputs that include (in the ideal case) the complete frequency contents of the operator of the system.
- In most applications of industrial practice the output signals obtained from sine sweeps as input signals are complex geometrical pattern not suited for identifying variations of subsystems by direct inspection (cf. 3b, 3c).
- On the other hand, systems controlled by exact linearization controllers reproduce (in the ideal case) input signals at the output. This implies that sine sweep stimuli provide sine sweeps as output signals (cf. 3e). Due to their simple structure (straight lines as envelopes) these output signals are extreme sensitive and well suited for identifying variations of subsystems or of subsystem models by direct inspection (cf. 3d).
- Therefore this combination (sine sweep stimuli and exact linearization controllers) provides a suitable method for fitting mathematical process models to laboratory experiments.



a) desired output signals

b) p-controller

c) multi sensor controller

d) partial exact linearization controller (ELLMNP)

e) complete exact linearization controller (ELLMNP)

Figure 3: System reactions (accelerations of the test table $\ddot{x}$ obtained from computer simulations) due to sine sweep stimuli using different controllers

The previously discussed fitting process will be performed in several steps in agreement with Figure 4.

Step I: Initialization of the fitting process

In a first step the fitting process will be initialized. That means:

- a "simple" computer simulation model G_{p_i} $(i = 0)$ called ideal model is chosen, including the nonlinear model of the servo-pneumatic actuators ([4]) without servo-valve dynamics and the linear model of the test facility mechanics without friction (Eq. (1)),
- realistic operating conditions O_j $(j = 1)$ are adjusted by choosing sine sweep test signals ($f = 1 - 25$ Hz, amplitude $\hat{a} = 10$ m/s^2) and a payload with mass $m_p = 13.4$ kg,
- realistic model parameters k_κ $(\kappa = 0)$ are chosen and
- an exact linearization controller C_l $(l = 0)$ including the model equations G_{p_0} is used.

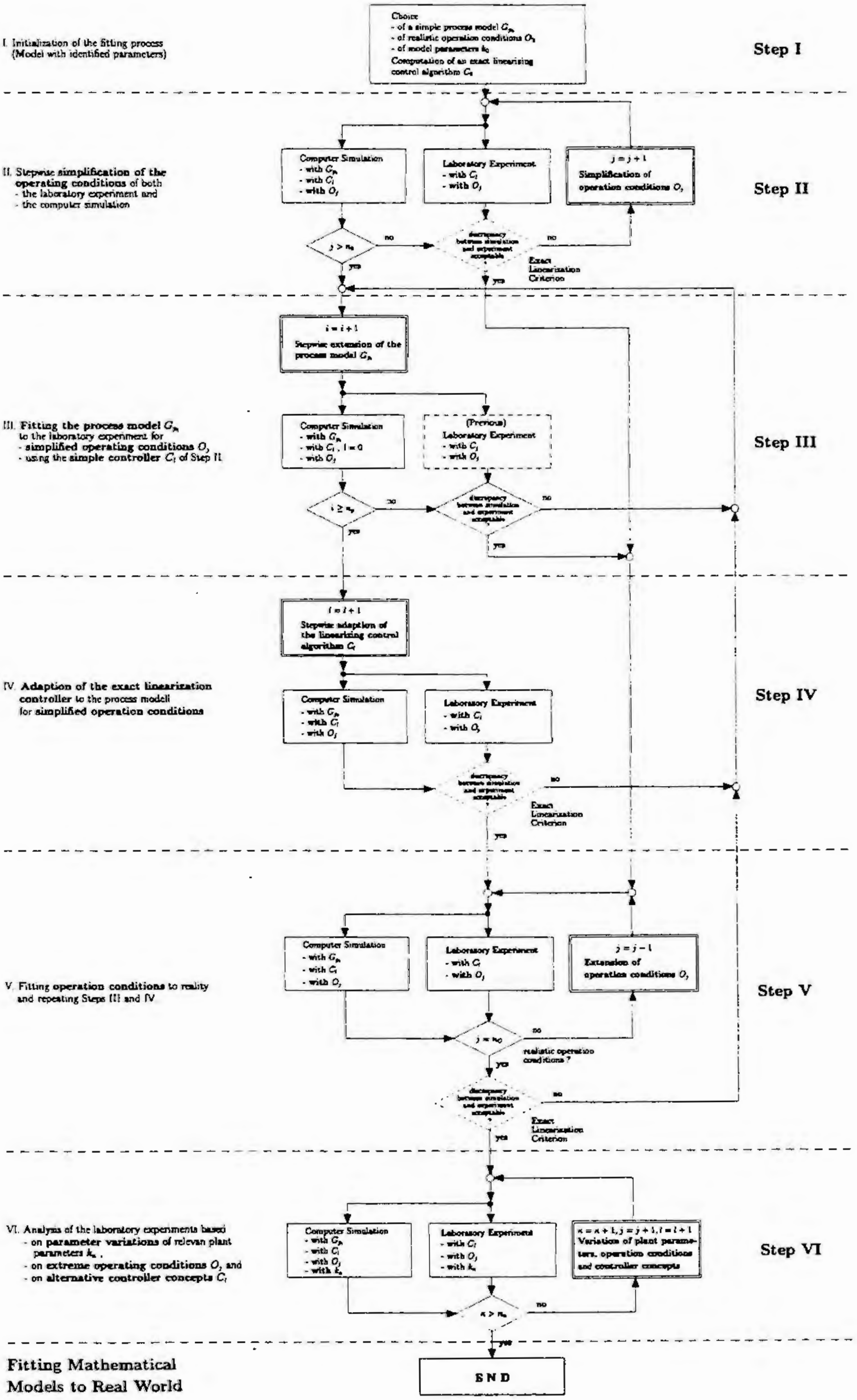

I. Initialization of the fitting process
(Model with identified parameters)
Choice
- of a simple process model G_{p_0}
- of realistic operation conditions O_0
- of model parameters k_0
Computation of an exact linearising control algorithm C_0
Step I
II. Stepwise simplification of the operating conditions of both
- the laboratory experiment and
- the computer simulation
Computer Simulation
- with G_{p_i}
- with C_l
- with O_j
Laboratory Experiment
- with C_l
- with O_j
$j = j + 1$
Simplification of operation conditions O_j
Step II
$j > n_0$
no
yes
discrepancy between simulation and experiment acceptable ?
Exact Linearization Criterion
$i = i + 1$
Stepwise extension of the process model G_{p_i}
III. Fitting the process model G_{p_i}
to the laboratory experiment for
- simplified operating conditions O_j
- using the simple controller C_l of Step II
Computer Simulation
- with G_{p_i}
- with C_l, $l = 0$
- with O_j
(Previous)
Laboratory Experiment
- with C_l
- with O_j
$i \geq n_p$
Step III
$l = l + 1$
Stepwise adaption of the linearizing control algorithm C_l
IV. Adaption of the exact linearization
controller to the process modell
for simplified operation conditions
Step IV
$j = j - 1$
Extension of operation conditions O_j
V. Fitting operation conditions to reality
and repeating Steps III and IV
$j = n_O$
realistic operation conditions ?
Step V
VI. Analysis of the laboratory experiments based
- on parameter variations of relevan plant parameters k_κ,
- on extreme operating conditions O_j and
- on alternative controller concepts C_l
- with k_κ
$\kappa = \kappa + 1, j = j + 1, l = l + 1$
Variation of plant parameters, operation conditions and controller concepts
$\kappa > n_\kappa$
Step VI
END
Fitting Mathematical
Models to Real World

Figure 4: Block diagram of the model fitting process

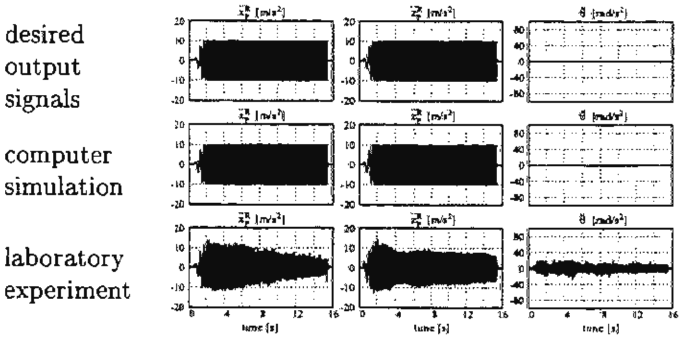


Figure 5: Results of Step I

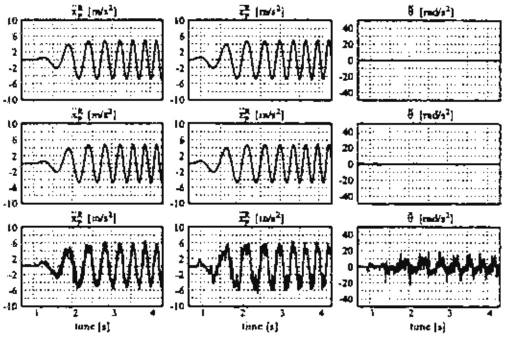


Figure 6: Results of Step II

Results of Step I (laboratory experiment and computer simulation (G_{p0}, O_1, k_0 and C_0):

The results of Step I are shown in Figure 5. The computer simulation results (second row) show an ideal control behavior compared to the test input signal (first row) but differ strongly from the laboratory experiment (third row). The control behavior of the laboratory experiment is not ideal compared to the test signal. To reduce the discrepancies between the computer simulations and the laboratory experiments the operating conditions O_0 will be weakened in a next step (cf. 4).

Step II: Stepwise simplification of the operating conditions

In this step the operating conditions O_j ($j = 1, 2, ..., n_O$) are weakened

- for reducing the discrepancy between computer simulations and laboratory experiments, and
- for improving the control behavior of the laboratory experiment.

This is done by reducing the frequency interval of the sine sweep to $f = 1 - 4$ Hz and the amplitude to 5 m/s^2.

Results of Step II (laboratory experiment and computer simulation (G_{p0}, O_{nO}, k_0 and C_0):

The results of Step II are shown in Figure 6. The control behavior of the laboratory experiment (third row) has been improved. The discrepancy between computer simulations (second row) and laboratory experiments (third row) has been reduced but not completely eliminated. To overcome this discrepancy the computer simulation model G_{p0} will be refined in the next step.

Step III: Refinement of the process model using weakened operating conditions

For decreasing the discrepancy between laboratory experiments and computer simulations a friction model is included in the computer simulation which provides the simulation model G_{p1}. Due to the restricted space of this paper a discussion of various other steps for including different submodels into the process model (cf. 8) is omitted here,

Results of Step III (laboratory experiment and computer simulation (G_{p1}, O_{n_O}, k_0 and C_0):

The results of Figure 7 show a further improved agreement between computer simulations and laboratory experiments. Due to the fact, that the control algorithm C_0 does not include the friction model, the control behavior of the simulated control loop is no longer ideal as in Figures 5 and 6. For the weakened

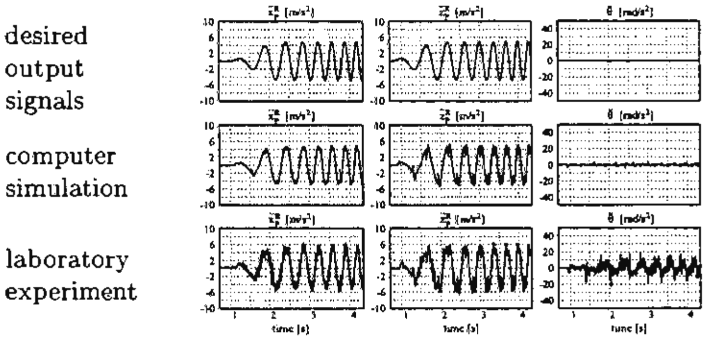


Figure 7: Results of Step III (accelerations of the test table $\ddot{x}$)

operating conditions O_{nO} considered, the deviation of the achieved control behavior from the ideal control behavior is caused by the friction of the actuators.

Step IV: Comparison of the results of the laboratory experiment and the computer simulation, both using the exact linearization controller

In Step IV the various submodels that have been stepwise included into the process model in Step III have been included into the control algorithm. Due to limited space available these results are omitted in this paper.

Step V: Repeating of Steps III and IV for realistic operating conditions

In this step the operating conditions O_{nO} have been reset to the more realistic operating conditions O_0 of Step I. In addition, various submodels have been stepwise included into and excluded from the process model of the computer simulation to investigate the influence of each of these submodels on the discrepancy between computer simulations and laboratory experiments for the more realistic operating conditions O_0.

Results of Step V (laboratory experiments and computer simulations (G_{p1}, O_0, k_0 and C_0):

The results of Step V are shown in Figure 8. They include time histories of the three accelerations ($\ddot{x}_P^R$, $\ddot{y}_P^R$ and $\ddot{\theta}$) and envelopes for various variations of submodels G_{pi}, $i = 1, 2, ..., n_{G_p}$. Comparing these results (rows b to h of Figure 8) with those obtained from the ideal process model (row a) and from the laboratory experiment (row i) leads to the following conclusions:

Fig. 8b: including the extended mechanics model into the process model (Eq. 7) will damp the output signals in the frequency range above 5 Hz and will amplify the output signals below 5 Hz,

Fig. 8c: including the submodel of the servo-valve dynamics ([4]) into the process model will amplify the output signals at frequencies above 5 Hz (reverse effect of case b),

Fig. 8d: including friction forces into the process model will tie up the amplitudes of the output signals at frequencies of about 7 Hz. In addition this submodel produces irregular peaks (like noise) in the amplitudes of the output signals,

Fig. 8e: limitations of the servo-valve piston displacement will damp the output signals beyond 20 Hz,

Fig. 8f: including signal transmission blocks of the discretization and quantization of the transmitted signals into the computer simulation will tie up the output signals at 7 Hz

Fig. 8g: including the signal filters used in the laboratory experiment into the computer simulation will damp the output signal at frequencies beyond 10 Hz.

Fig. 8h: including all submodels of cases b to g into the process model yields the time histories in the last but one row (Case h of Figure 8). A comparison of these time histories (obtained from computer simulations) with the time histories obtained from laboratory experiments (Case i of Fig. 8) proves that the mathematical models have been well fitted to the laboratory experiment.

In addition the specific influence of various subsystem models on the behavior of the computer simulation model has been clearly shown.

Step VI: Verification of the process model

In this step the success of the model fitting process has been finally tested using different process controllers (C_l, $l = 1, 2, 3, 4, 5$) that are capable of providing quite different operating conditions both, of the process model and of the process. The controllers used are

- linear standard industrial controllers
  - a proportional controller (P) and
  - a multi sensor controller (MSR), and
- exact linearization controllers based on different process models
  - ELLMLP: based on linear reduced mechanical models (Eq. (1)) and on linearized servo-pneumatic models ([4]),
  - ELLMRP: based on linear reduced mechanical models (Eq. (1)) and on nonlinear reduced servo-pneumatic models ([6]), and
  - ELLMNP: based on linear reduced mechanical models (Eq. (1)) and on nonlinear extended servo-pneumatic models ([4]).

The results of the laboratory experiments have been compared with associated computer simulations.

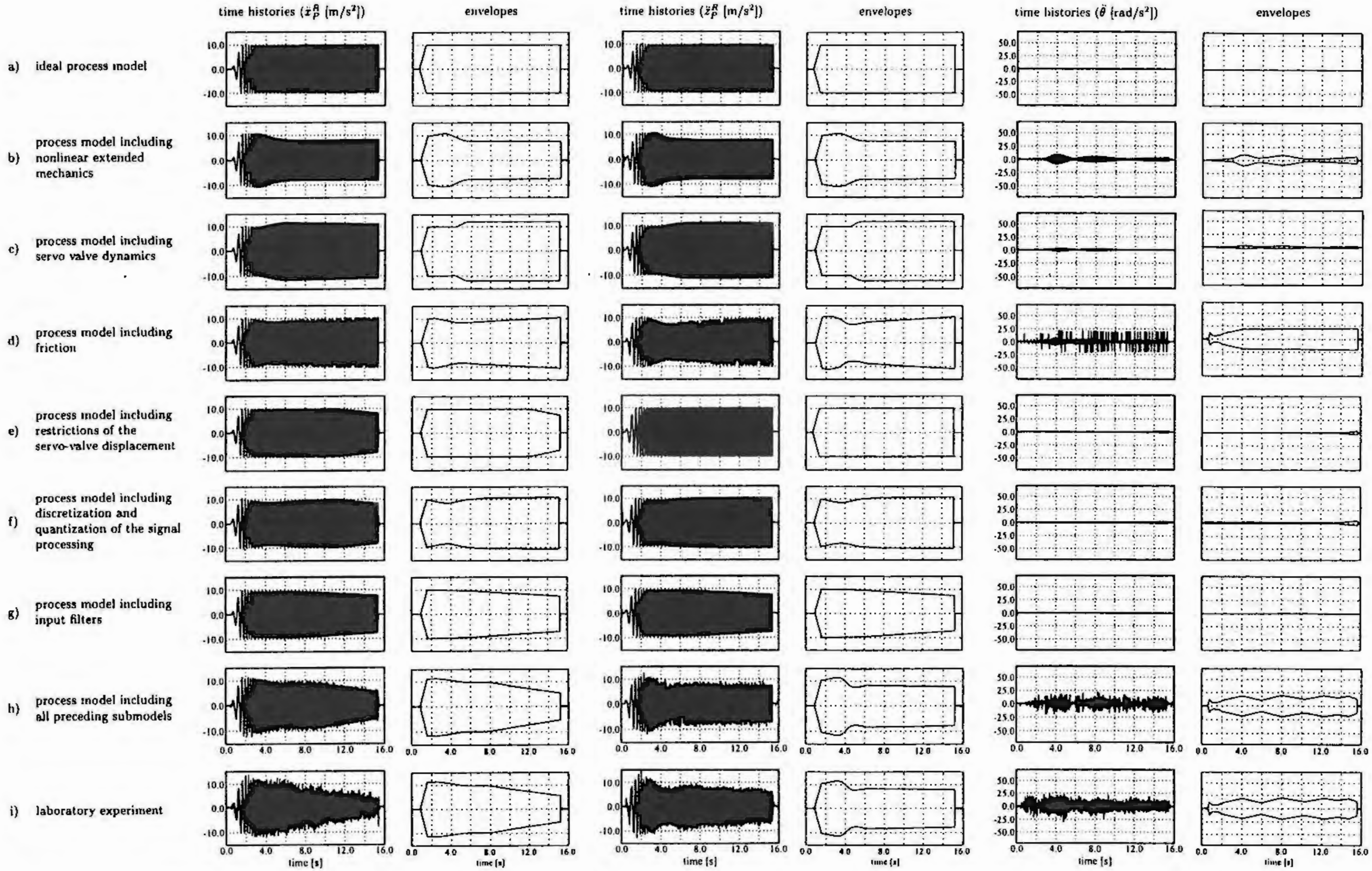


Figure 8: Variation of submodels of the process model and their influence on the discrepancies between computer simulations and laboratory experiments (time history of the frequency of the desired output signal $x_d(t)$: $f(t) = \left(\frac{24}{15} \cdot \left(\frac{t}{8} - 0.5\right) + 1\right)$ Hz, for time $t = 0.5, ..., 15.5$ s)

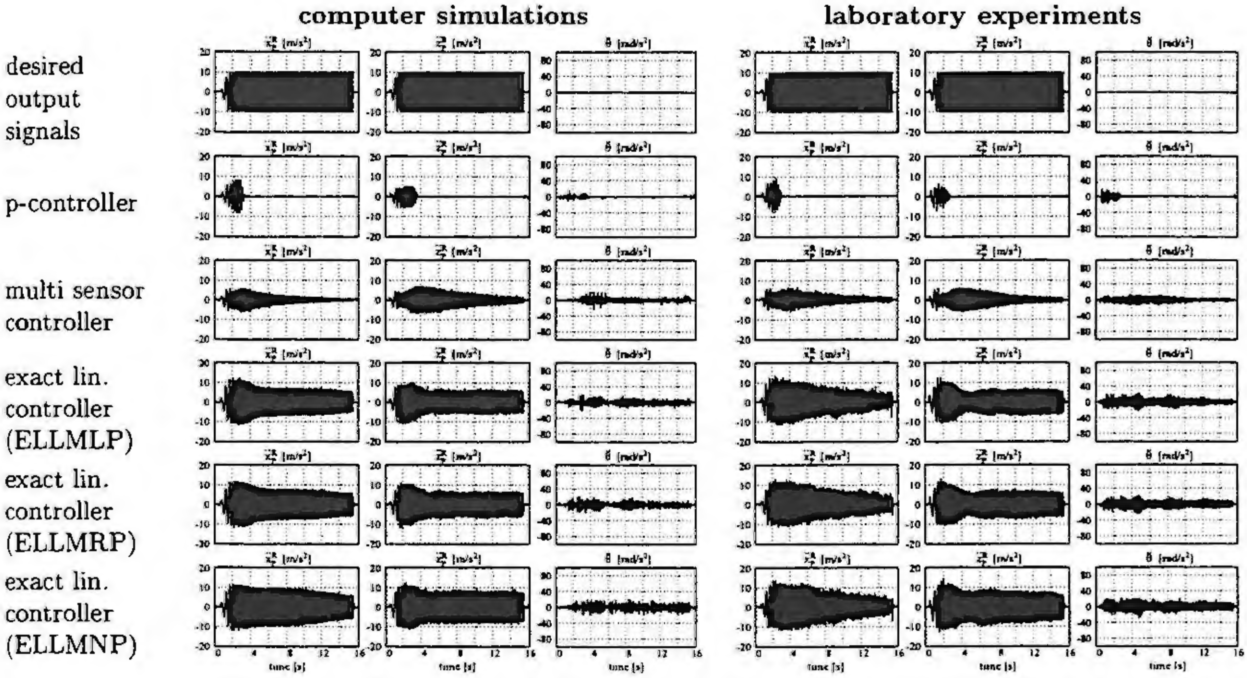


Figure 9: Results of Step VI (accelerations of the test table $\ddot{x}$)

Results of Step VI (laboratory experiments and computer simulations (G_{p1}, O_0, k_0, $C_{1,\ldots,5}$):

The results of Step VI are shown in Figure 9. They demonstrate an excellent agreement of the process model with the laboratory experiment under extreme variations of the operating conditions caused by different control algorithms.

Conclusions

In this paper a new methodology has been presented for fitting mathematical models (in our case mathematical models of a planar servo–pneumatic test facility) to laboratory experiments based on exact linearization techniques. This method is well suitable

- for investigating the specific effects of various submodels included in or excluded from the process model,
- for identifying and eliminating the sources of discrepancies between computer simulations and laboratory experiments, and
- for fitting mathematical models to real processes.

As a consequence this method provides a deep physical insight into the process behavior.

References

[1] Hahn, H.; Raasch, W. *Multi-axis Vibration Tests on Spacecraft using Hydraulic Exciters.* In *Conference Proceedings No. 397, p. 23-1 to 23-23.* AGARD, 1986.

[2] Merklinghaus, W.; Raasch, W.; Eggert, H.; Kugler, R. *Rechnergesteuerte hydraulische Schwinganlage zur Prüfung der Erdbeebensicherheit von Hochspannungsschaltgeräten. Siemens–Zeitschrift,* 51, Heft 3, 1977.

[3] Hahn, H.; Zhang, X.; Leimbach, K.-D.; Sommer, H.-J. *Nonlinear Control of a Planar Multi-Axis Servohydraulic Test-Facility Using Exact Linearization Techniques. Kybernetica,* 1994.

[4] Piepenbrink, A.; Hahn, H. *Fitting mathematical models of a servopneumatic actuator to the laboratory experiment using exact linearization techniques.* In *this volume.*

[5] Fürst, D.; Hecker, F. *Modellbildung eines ebenen Mehrachsenprüfstandes mit massebehafteten Antriebszylindern durch explizite Differentialgleichungen.* RTS–Bericht RT-21, Fachgebiet Regelungstechnik und Systemdynamik, Universität Kassel, 1996.

[6] Piepenbrink, A. *Identifikation und Regelung servopneumatischer Antriebe.* Dissertation, Fachgebiet Regelungstechnik (Maschinenbau), Universität-Gh Kassel, 1996.

FITTING MATHEMATICAL MODELS OF A SERVOPNEUMATIC ACTUATOR TO LABORATORY EXPERIMENTS USING EXACT LINEARIZATION TECHNIQUES

A. Piepenbrink[1]and H. Hahn[2]

[1]ZF Friedrichshafen AG, Forschungs– und Entwicklungszentrum, Abt. TE–P,
Allmannsweilerstr. 43, 88038 Friedrichshafen, Germany
Phone +49 7541-77-7357, Fax +49 7541-77-7523, e-Mail: piepenbrink@t-online.de
[2]Control Engineering and System Theory Group, Department of Mechanical
Engineering (FB15), University of Kassel, 34109 Kassel, Mönchebergstraße 7, Germany,
Phone +49 561-804 32 60, Fax +49 561-804 77 68, e-Mail: hahn@hrz.uni-kassel.de

Abstract. In this paper results of a model fitting process applied to a servopneumatic actuator are presented. In this fitting process the method of exact linearization techniques is used to judge the difference of the results obtained in laboratory experiments and in computer simulations, including different linear and nonlinear process models. The model fitting process includes a sequence of systematic model fitting steps. These model fitting steps include variations of linear and nonlinear plant models, variations of operating conditions and variations of linear and nonlinear exact linearization controllers. The final nonlinear process model obtained provides excellent agreement between computer simulations and laboratory experiments for a wide range of operating conditions.

1 Introduction

Servopneumatic actuators are extensively used in various industrial applications ([1], [2]). In this paper the method of exact linearization techniques is used to find a plant model of a servopneumatic actuator that accurately describes the laboratory experiment. The linear and nonlinear process models are collected in Section 2. They include linear and nonlinear differential equations of the thermodynamical, fluid mechanical and mechanical processes of the actuator components. The exact linearization controllers are briefly described in Section 3. In Section 4 the model fitting process and the results obtained in computer simulations and in laboratory experiments are discussed. These results show that a nonlinear simulation model, which accurately describes the laboratory experiments under various operating conditions, has been obtained by using the systematic fitting procedure presented.

2 Mathematical models of the servopneumatic actuator

Nonlinear plant model

The various components of a servopneumatic actuator (cf. 1) have been modeled by the following equations describing the dynamical behaviour of the servo valve, of the actuator, of the mass flows over the control egdes of the servo valve and of the actuator piston mechanics, including the load.

Servo valve magnetics:

$$\dot{F}_v + a_M \cdot F_V = k_M \cdot u \tag{1}$$

with **model variables** F_v (force of the magnetic amplifier) and u (input voltage of the magnetic amplifier) and with **model parameters** k_M (magnetic amplifier gain factor) and a_M (pole of the transfer function of (1)).

Servo valve mechanics:

$$\ddot{x}_v + 2 \cdot \zeta_v \cdot \omega_v \cdot \dot{x}_v + \omega_v^2 \cdot x_v = k_V \cdot \omega_v^2 \cdot F_v \tag{2}$$

with **model variable** x_v (servo valve piston displacement) and with **model parameters** k_V (servo valve gain factor), ω_v (servo valve frequency) and ζ_v (servo valve damping factor).

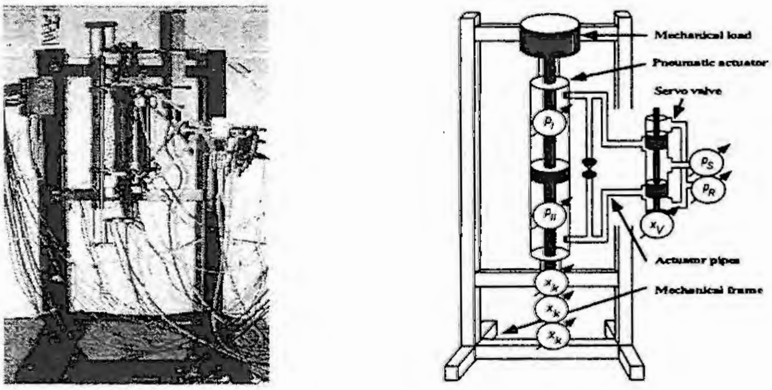


Figure 1: Hardware realization of the laboratory experiment (photo and computer drawing) including sensing elements

Pressure evolution in the actuator chambers:

$$\frac{\dot{p}_I}{\kappa}\cdot(V_{0I}+A_I\cdot x_k) \quad + A_I\cdot\dot{x}_k\cdot p_I \quad = +R{\cdot}T_I\cdot\,(\dot{m}_2-\dot{m}_1-\dot{m}_{BP}) \quad , \tag{3}$$

$$\frac{\dot{p}_{II}}{\kappa}\cdot(V_{0II}-A_{II}\cdot x_k) - A_{II}\cdot\dot{x}_k\cdot p_{II} = -R{\cdot}T_{II}{\cdot}(\dot{m}_4-\dot{m}_3-\dot{m}_{BP}) \tag{4}$$

with **model variables** x_k (actuator piston displacement), $\dot{x}_k$ (actuator piston velocity), $p_{I,II}$ (actuator chamber I,II pressures), $T_{I,II}$ (actuator chamber I,II temperatures), $\dot{m}_{1,2,3,4}$ (mass flows over the control edges of the servo valve) and $\dot{m}_{BP}$ (bypass mass flows across the actuator) and with **model parameters** $V_{0I,0II}$ (initial volume of actuator chambers I,II), $A_{I,II}$ (areas of actuator piston in the actuator chambers I,II), R (gas constant) and κ (adiabatic exponent).

Mass flow over the control egdes of the servovalve:

$$m_1'\cdot x_v :=\dot{m}_1 = -\alpha_{D1}\cdot\pi\cdot d_1{\cdot}\psi\left(\frac{p_R}{p_I}\right)\cdot p_I\cdot\sqrt{\frac{2}{R\cdot T_I}}\;\cdot\sigma(-x_v)\cdot x_v \quad , \tag{5}$$

$$m_2'\cdot x_v :=\dot{m}_2 = \;\;\alpha_{D2}\cdot\pi\cdot d_2{\cdot}\psi\left(\frac{p_I}{p_S}\right)\cdot p_S\cdot\sqrt{\frac{2}{R\cdot T_S}}\;\cdot\sigma(x_v)\cdot x_v \quad , \tag{6}$$

$$m_3'\cdot x_v :=\dot{m}_3 = -\alpha_{D3}\cdot\pi\cdot d_3{\cdot}\psi\left(\frac{p_{II}}{p_S}\right)\cdot p_S\cdot\sqrt{\frac{2}{R\cdot T_S}}\;\cdot\sigma(-x_v)\cdot x_v \quad , \tag{7}$$

$$m_4'\cdot x_v :=\dot{m}_4 = \;\;\alpha_{D4}\cdot\pi\cdot d_4{\cdot}\psi\left(\frac{p_R}{p_{II}}\right)\cdot p_{II}\cdot\sqrt{\frac{2}{R\cdot T_{II}}}{\cdot}\sigma(x_v)\cdot x_v \tag{8}$$

and **bypass mass flow across the actuator:**

$$\dot{m}_{BP} = \left\{\begin{array}{lll} \alpha_{BP}\cdot A_{BP}\cdot\psi\left(\frac{p_{II}}{p_I}\right)\cdot p_I\cdot\sqrt{\frac{2}{R\cdot T_I}} & \text{for} & p_I \geq p_{II} \\ -\alpha_{BP}\cdot A_{BP}\cdot\psi\left(\frac{p_I}{p_{II}}\right)\cdot p_{II}\cdot\sqrt{\frac{2}{R\cdot T_{II}}} & \text{for} & p_I < p_{II} \end{array}\right\} \tag{9}$$

with **model variables** $p_{R,S}$ (pressures of the fluid supply) and with **model parameters** $\alpha_{D1,D2,D3,D4,BP}$ (coefficients of the valve orifice flows and the bypass mass flow), $d_{1,2,3,4}$ (diameters of the control edges of the servo valve) and A_{BP} (area of the bypass mass flow). In (5) – (8) a servo valve with ideal critical center is assumed. These equations include the flow function of ISO 6358 [3]

$$\psi(\frac{p_a}{p_b}) = \psi_0\cdot\left(\begin{array}{ll} \sqrt{1-\left(\frac{\frac{p_a}{p_b}-p_{krit}}{1-p_{krit}}\right)^2} & \text{for } \frac{p_a}{p_b}\geq p_{krit} \\ 1 & \text{for } \frac{p_a}{p_b} < p_{krit} \end{array}\right) \;;\; \psi_0 = 0.484 \tag{10}$$

with **model variable** p_a/p_b (pressure ratio at the control edge [3]) and with **model parameter** p_{krit} (critical pressure ratio [3]).

Actuator piston mechanics:

$$m_k \cdot \ddot{x}_k - p_I \cdot A_I + p_{II} \cdot A_{II} - m_k \cdot g + c_k \cdot \dot{x}_k + sign(\dot{x}_k) \cdot d_k = 0 \tag{11}$$

with **model variable** $\ddot{x}_k$ (actuator piston acceleration) and with **model parameters** c_k (viscous damping coefficient) and d_k (Coulomb friction coefficient).

Linear plant model

The linear model equations of a servopneumatic actuator are

Servovalve magnetics and mechanics:

$$\dot{F}_v + a_M \cdot F_V = k_M \cdot u \quad , \tag{12}$$

$$\ddot{x}_v + 2 \cdot \zeta_v \cdot \omega_v \cdot \dot{x}_v + \omega_v^2 \cdot x_v = k_V \cdot \omega_v^2 \cdot F_v \quad . \tag{13}$$

Pressure evolution in the actuator chambers:

$$\dot{p}_L = \frac{M_X}{C_P} \cdot x_v + \frac{M_P}{C_P} \cdot p_L + \frac{M_K}{C_P} \cdot \dot{x}_k \tag{14}$$

including the coefficients (relative to an operating point x_{vc}, p_{Lc}, $\dot{x}_{kc} = 0$)

$$M_X = \begin{pmatrix} \pi \cdot \alpha_{D2} \cdot d_2 \cdot \sqrt{2 \cdot R \cdot T_S} \cdot \psi_0 \cdot (f_i(p_S,\, p_R) - p_{LC}) & \text{for} & & & x_{\ddot{u}2} & \leq & x_{vC} \\ 0 & \text{for} & -x_{\ddot{u}1} & \leq & x_{vC} & \leq & x_{\ddot{u}2} \\ \pi \cdot \alpha_{D1} \cdot d_1 \cdot \sqrt{2 \cdot R \cdot T_S} \cdot \psi_0 \cdot (f_i(p_S,\, p_R) + p_{LC}) & \text{for} & & & x_{vC} & \leq & -x_{\ddot{u}1} \end{pmatrix} \quad , \tag{15}$$

$$M_P = \begin{pmatrix} -\pi \cdot \alpha_{D2} \cdot d_2 \cdot \sqrt{2 \cdot R \cdot T_S} \cdot \psi_0 \cdot (x_{vC} - x_{\ddot{u}2}) - k_{BP} & \text{for} & & & x_{\ddot{u}2} & \leq & x_{vC} \\ -k_{BP} & \text{for} & -x_{\ddot{u}1} & \leq & x_{vC} & \leq & x_{\ddot{u}2} \\ \pi \cdot \alpha_{D1} \cdot d_1 \cdot \sqrt{2 \cdot R \cdot T_S} \cdot \psi_0 \cdot (x_{vC} + x_{\ddot{u}1}) - k_{BP} & \text{for} & & & x_{vC} & \leq & -x_{\ddot{u}1} \end{pmatrix} \quad , \tag{16}$$

$$k_{BP} = \frac{\kappa}{V_0} \frac{A_{BP} \cdot \alpha_{BP} \cdot \sqrt{2 \cdot R \cdot T_S} \cdot \psi_0 \cdot [f_i(p_S,\, p_R) \cdot (1 - p_{krit}) - 2 \cdot p_{krit} |p_{LC}|]}{2 \cdot (1 - p_{krit}) \cdot \sqrt{f_i(p_S,\, p_R) \cdot (1 - p_{krit}) \cdot |p_{LC}| - |p_{LC}|^2 \cdot p_{krit}}} \tag{17}$$

and

$$M_K = -A \cdot \; f_i(p_S,\, p_R) \quad , \tag{18}$$

where the Cases $i = I,\ II$ take into account different pressure relations describing different operating conditions of the actuator [4]

$$f_I = p_I + p_{II} = + p_S \cdot (p_{krit} + \sqrt{2 \cdot (1 - p_{krit}) + p_{krit}^2}) \quad , \tag{19}$$

$$f_{II} = p_I + p_{II} = -\frac{p_{krit} \cdot p_R}{1 - 2 \cdot p_{krit}} + p_{krit} \cdot p_S + \frac{\sqrt{2} \cdot (1 - p_{krit})}{(1 - 2 \cdot p_{krit})} \cdot \sqrt{p_R^2 \cdot (1 - p_{krit}) + p_S^2 \cdot (1 - p_{krit}) \cdot (1 - 2 \cdot p_{krit})} \quad . \tag{20}$$

Actuator piston mechanics:

$$-m_k \cdot \ddot{x}_k \; + A \cdot p_L - m_k \cdot g = c_k \cdot \dot{x}_k \quad . \tag{21}$$

The relations (14) – (20) have been derived in [4] from the nonlinear model equations of Section 2.

3 Controller design using exact linearization techniques

In this section the design of linear and nonlinear compensation controllers will be briefly discussed [4]. These controllers are used in the model fitting approach described in Section 4.

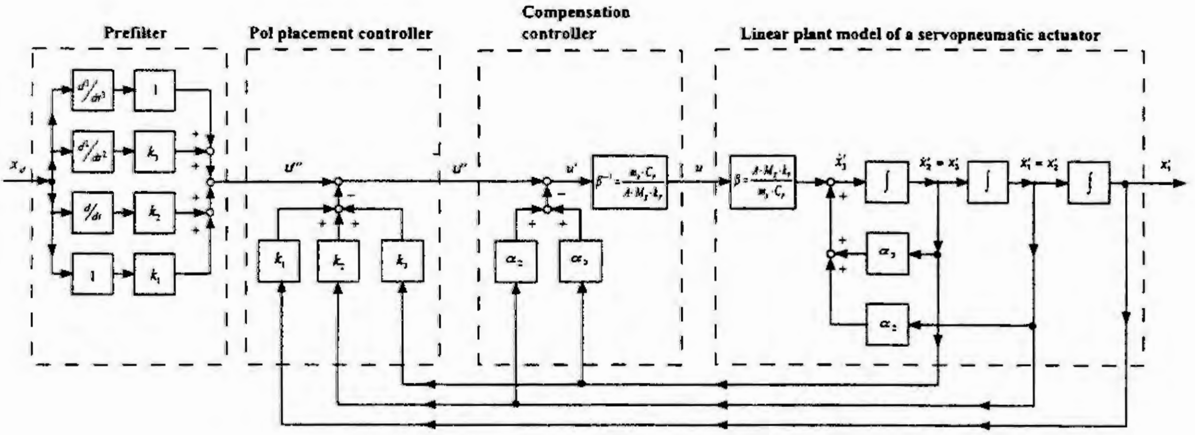


Figure 2: Block diagram of the linear plant (22) with linear compensation controller (25)

Linear compensation controller

The state space representation of the linear plant model (14), (21) [4] using $x' = (x_1, x_2, x_3)^T = (x_k, \dot{x}_k, \ddot{x}_k)^T$ as state vector and $y = x_k$ as output variable is written in normal form

$$\underbrace{\begin{bmatrix} \dot{x}_1' \\ \dot{x}_2' \\ \dot{x}_3' \end{bmatrix}}_{\dot{x}'} = \underbrace{\begin{bmatrix} 0 & , & 1 & , & 0 \\ 0 & , & 0 & , & 1 \\ 0 & , & \alpha_2 & , & \alpha_3 \end{bmatrix}}_{A'} \cdot \underbrace{\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix}}_{x'} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \beta \end{bmatrix}}_{B'} \cdot u \quad , \tag{22}$$

$$y = \underbrace{[1, 0, 0]}_{C'} \cdot x' = x_1' = x_k \quad , \tag{23}$$

with

$$\alpha = (\alpha_1, \alpha_2, \alpha_3)^T = (0, \frac{M_P \cdot c_k}{m_k \cdot C_P} + \frac{A \cdot m_k}{M_K \cdot C_P}, \frac{M_P}{C_P} - \frac{c_k}{m_k})^T \quad \text{and} \quad \beta = \frac{A \cdot M_X \cdot k_v}{m_k \cdot C_P} \quad . \tag{24}$$

The linear compensation controller (abbreviated with $EXL3o$ in subsequent time histories)

$$\begin{aligned} u = [&+ \dddot{x}_d + k_3 \cdot (\ddot{x}_d - \ddot{x}_k) + k_2 \cdot (\dot{x}_d - \dot{x}_k) + k_1 \cdot (x_d - x_k) \\ &- \alpha_1 \cdot x_k - \alpha_2 \cdot \dot{x}_k - \alpha_3 \cdot \ddot{x}_k \;] \cdot \beta^{-1} \quad , \end{aligned} \tag{25}$$

including a pole placement controller k_1, k_2 and k_3, and a standard prefilter, provides an ideal transmission behaviour of the linear plant model (22), using x_d as desired command input signal and $y = x_k$ as output function (cf. 2).

Nonlinear exact linearization controller

In analogy to the linear compensation controller design with linear plant model (22), (23) and (24), the nonlinear plant model (3), (4), (5), (6), (7), (8), (10) and (11) (with d_k=0) is written in the form

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = x_3, \quad \dot{x}_3 = \alpha(x) + \beta(x) \cdot u \quad \text{and with} \quad x := (x_1, x_2, x_3)^T = (x_k, \dot{x}_k, \ddot{x}_k)^T \quad , \tag{26}$$

including the nonlinear functions $\alpha(x)$ and $\beta(x)$

$$\alpha(x) = -\frac{c_k}{m_k} \cdot \ddot{x}_k - \frac{\kappa \cdot A_I^2 \cdot \dot{x}_k \cdot p_I}{(V_{0I} + A_I \cdot x_k) \cdot m_k} - \frac{\kappa \cdot A_{II}^2 \cdot \dot{x}_k \cdot p_{II}}{(V_{0II} - A_{II} \cdot x_k) \cdot m_k} \quad , \tag{27}$$

$$\beta(x) = \frac{A_I \cdot \kappa \cdot R \cdot T_I}{m_k} \cdot \frac{(\dot{m}_2' - \dot{m}_1') \cdot k_v}{V_{0I} + A_I \cdot x_k} + \frac{A_{II} \cdot \kappa \cdot R \cdot T_{II}}{m_k} \cdot \frac{(\dot{m}_4' - \dot{m}_3') \cdot k_v}{V_{0II} - A_{II} \cdot x_k} \quad . \tag{28}$$

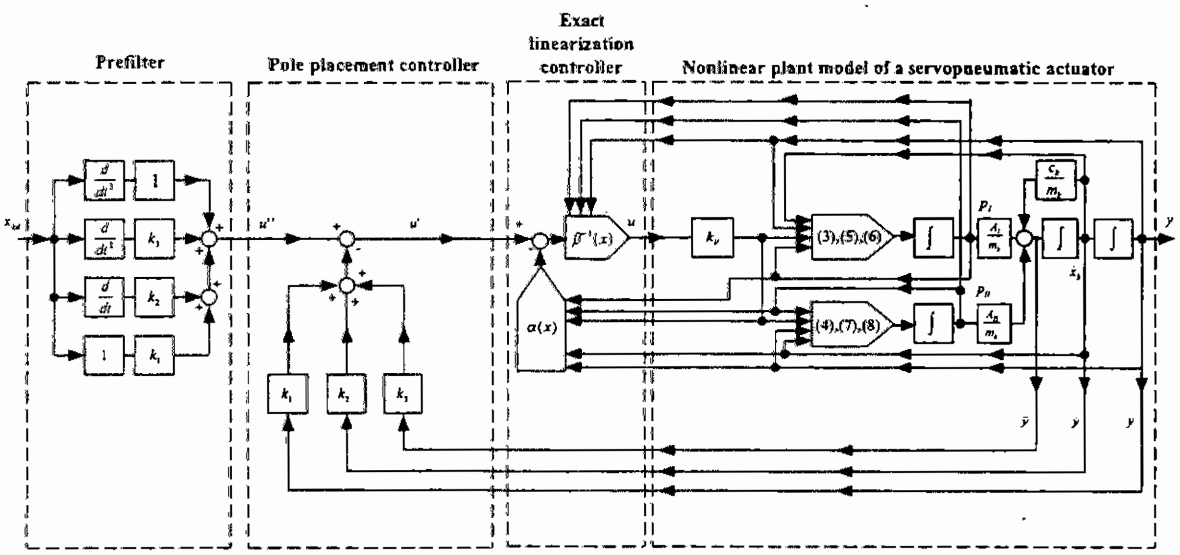


Figure 3: Block diagram of the nonlinear plant (3), (4), (5), (6), (7), (8), (10) and (11) (with d_k=0) with nonlinear controller (29)

Using exact i/o–linearization techniques ([4], [5], [6]) with $y = x_k$ as output function of the control loop, yields the following nonlinear controller (abbreviated with *EXNL4o* in subsequent time histories)

$$u = [+\dddot{x}_d + k_3 \cdot (\ddot{x}_d - \ddot{x}_k) + k_2 \cdot (\dot{x}_d - \dot{x}_k) + k_1 \cdot (x_d - x_k) - \alpha(x)] \; / \; \beta(x) \quad , \tag{29}$$

with $\alpha(x)$ and $\beta(x)$ defined in (27) and (28), including a pole placement controller k_1, k_2 and k_3, and a standard prefilter and providing an ideal transmission behaviour of the i/o–linearized nonlinear plant model between output x_k and input u (cf. 3).

4 Model fitting approach and model fitting results

The **model fitting approach** presented in this paper is based on exact linearization techniques. The idea underlying this approach can be briefly described as follows: Let $x(t)$ be a time signal of a simple geometrical shape (like a sine sweep of constant amplitude, a saw tooth or a sequence of pulses of identical shape). Then even small changes of those signals can be identified by direct inspection. On the other hand, entering command input signals $u(t)$ into a given dynamical system (even if $u(t)$ has a simple shape as $x(t)$) provides output signals $y_1(t)$ of a much complexer geometrical shape. Variations of the structure of the system provide (together with the same input signal $u(t)$) new output signals $y_2(t)$, again of a complex shape. Due to the complexity of the two output signals $y_1(t)$ and $y_2(t)$ it is in general hard to identify and to characterize the modification of the system structure from the two output signals $y_1(t)$ and $y_2(t)$. This is different if the system considered is controlled by an exact linearizing controller. Then the controlled system reproduces (in the ideal case) input signals (of a simple shape) at the output, and slight modifications of the structure of the system considered provide characteristic modifications of the output signal $y(t)$ that can be easily recognized, in case the controller has been kept constant and does not include the modification of the system. This model fitting approach is represented in the block diagram of Figure 4. It includes tbe following steps:

(I) **Initialization of the model fitting process** (choice of a first process model, of an ideal associated compensation controller and of initial operating conditions).

(II) Stepwise **simplification of the operating conditions** of both, the laboratory experiment and the computer simulation.

(III) **Fitting of the process model** to the laboratory experiment for **simplified operating conditions** using the **controller of step II.**

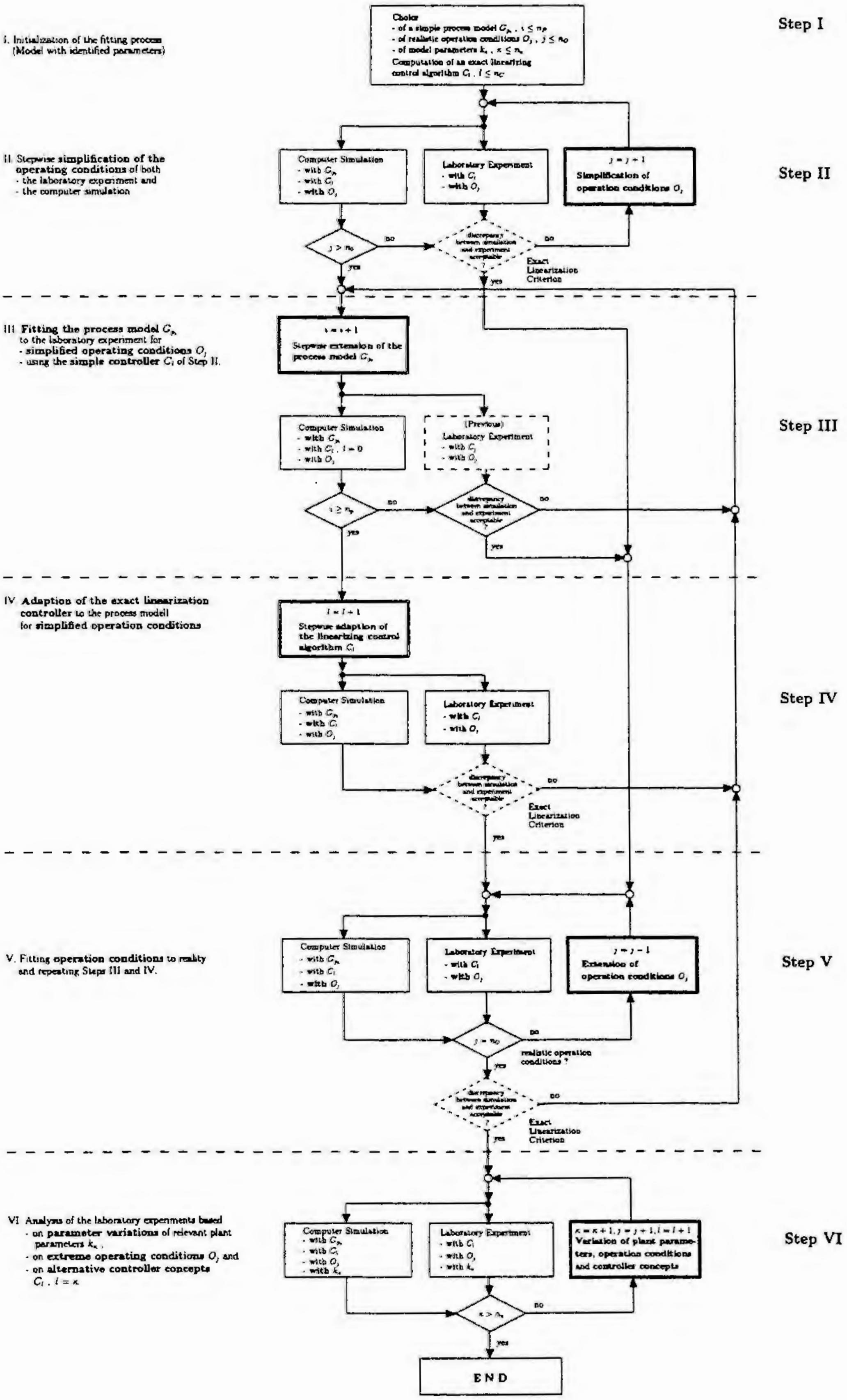
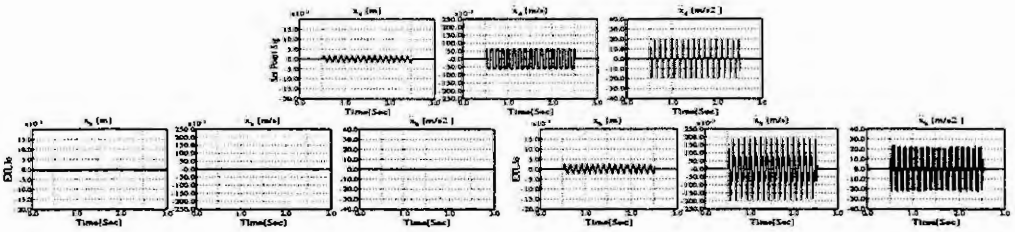
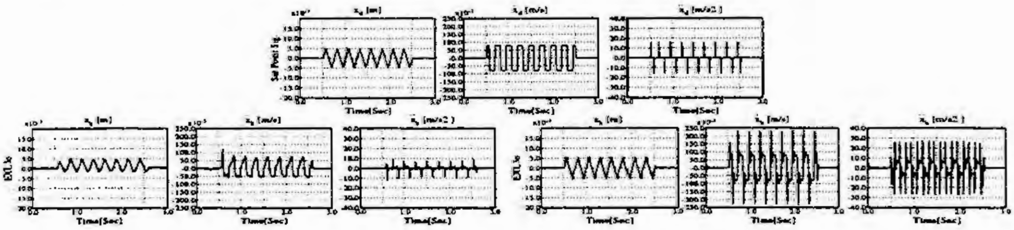


Figure 4: Block diagram of the model fitting procedure



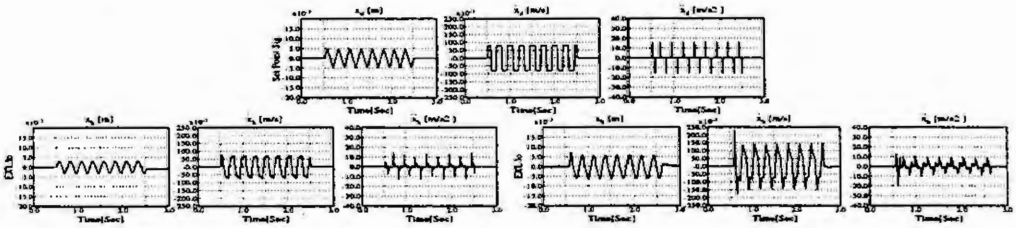
(a) **laboratory experiment** (b) **computer simulation**

Figure 5: Time histories of laboratory experiments and of computer simulations (**Step I of Figure 4**. Computer simulations with **linear controller EXL3o** and without friction model in the **linear plant model**; **8 Hz** triangular command input signal)



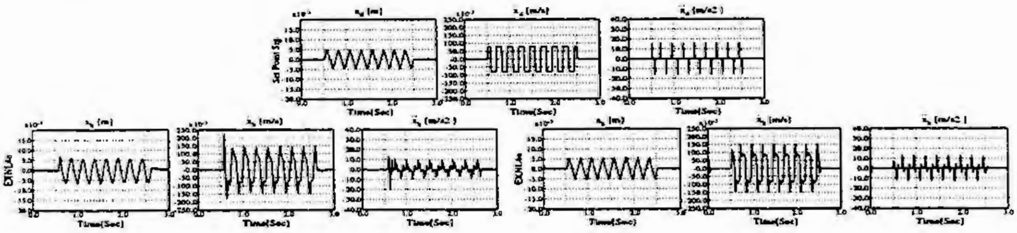
(a) **laboratory experiment** (b) **computer simulation**

Figure 6: Time histories of laboratory experiments and of computer simulations (**Step II of Figure 4**. Computer simulations with **linear controller EXL3o** and without friction model in the **linear plant model**; **4 Hz** triangular command input signal)



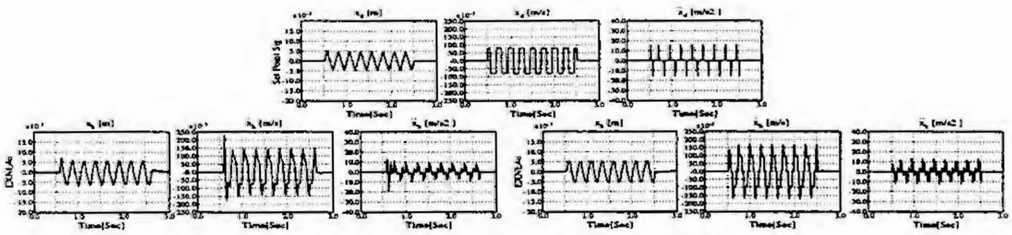
(a) **laboratory experiment** (b) **computer simulation**

Figure 7: Time histories of laboratory experiments and of computer simulations (**Step III of Figure 4**. Computer simulations and laboratory experiments with **linear controller EXL3o** and without friction model in the **nonlinear plant model**; **4 Hz** triangular command input signal)



(a) **laboratory experiment** (b) **computer simulation**

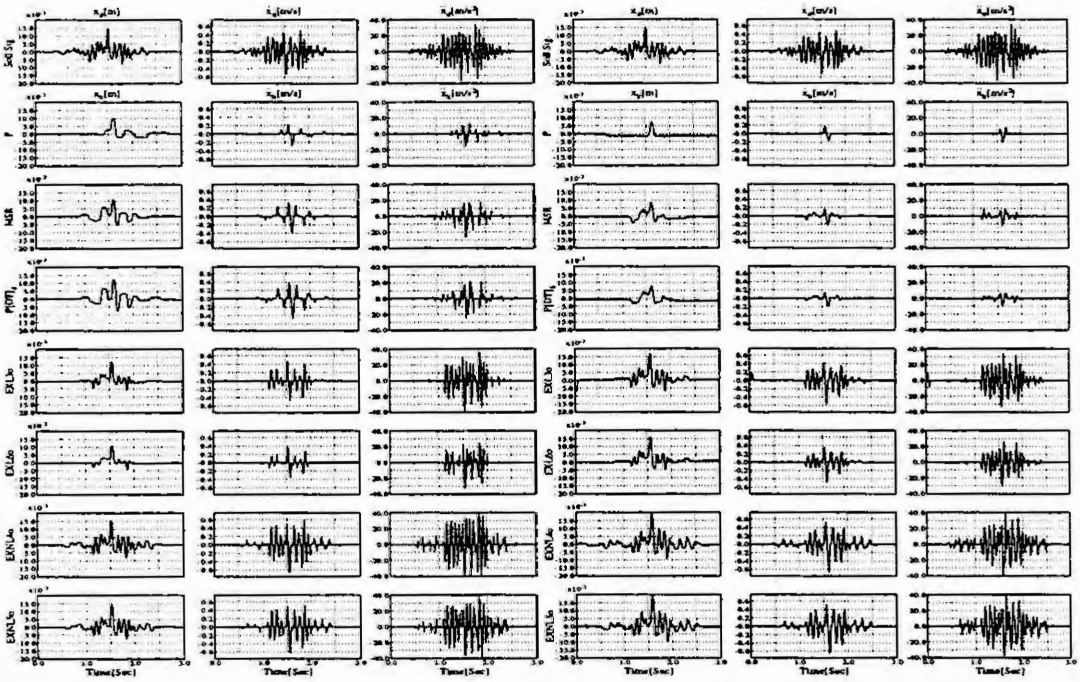
Figure 8: Time histories of laboratory experiments and of computer simulations (**Step IV of Figure 4**. Computer simulations with **nonlinear controller EXNL4o** and without friction model in the **nonlinear plant model**; **4 Hz** triangular command input signal)



(a) **laboratory experiment**

(b) **computer simulation**

Figure 9: Time histories of lahoratory experiments and of computer simulations (**Step III of Figure 4**. Computer simulations with **nonlinear controller EXNL4o** and **with friction** model in the **nonlinear plant model**; **4 Hz** triangular command input signal)



(a) **laboratory experiments**

(b) **computer simulations**

Figure 10: Time histories of laboratory experiments and of computer simulations using various linear and nonlinear controllers (**Step V of Figure 4**, using industrial test signals and (O3))

References

[1] Hahn, H. Concept of a safety system for a servohydraulic test facility. In Proceedings of Workshop on spacecraft vibration testing. ESTEC, 1983.

[2] Hahn, H.; Krenz, D. Simulation of electro–servo–hydraulic actuator system for high quality testing. In Proceedings of Workshop on spacecraft vibration testing. ESTEC, 1983.

[3] ISO-6358. Pneumatic fluid power - Components using compressible fluids - Determination of flow rate characteristics. International Standard Norm, 1989.

[4] Piepenbrink, A. Experimentelle Identifikation und Regelung servo–pneumatischer Antriebe. Dissertation, Fachgebiet Regelungstechnik (Maschinenbau), Universität Kassel (Maschinenbau), 1996.

[5] Hahn, H.; Leimbach, K.-D. Nonlinear Control Systems. IMAT Bericht RT-18, Fachgebiet Regelungstechnik (Maschinenbau), Universität Kassel, 1995.

[6] Slotine, E.; Weiping, L. Applied Nonlinear Control. Prentice Hall, New Jersey, 1991.

MATHEMATICAL MODELING, COMPUTER SIMULATION AND DYNAMICAL ANALYSIS OF A SINGLE–POINT–DRIVE ECCENTRIC PRESS

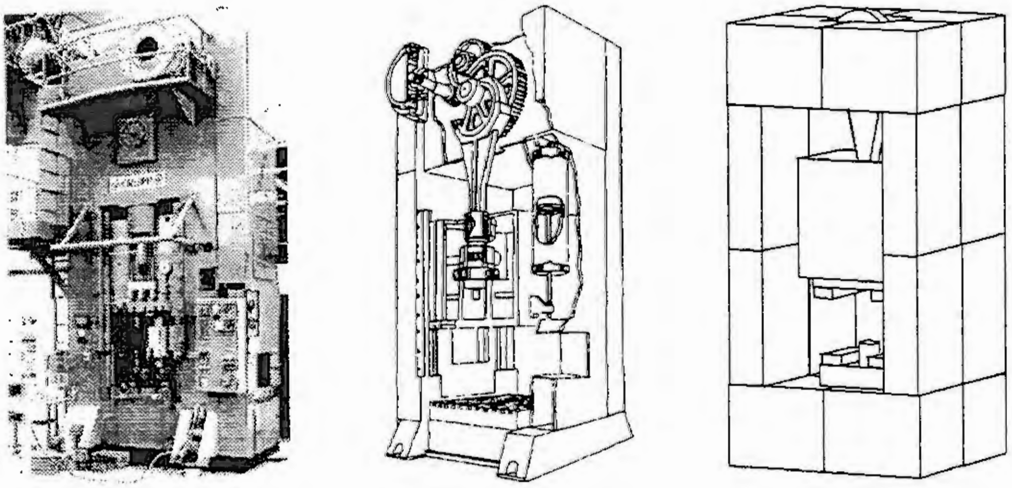
M. Neumann, H. Hahn[1], K.-D. Leimbach

Control Engineering and System Theory Group, Department of Mechanical Engineering (FB 15), University of Kassel, 34109 Kassel, Moenchebergstr. 7, Germany, eMail: hahn@hrz.uni-kassel.de

Abstract. In this paper computer simulation models of different complexity of a single–point–drive eccentric press are presented. These models as well as the model parameters have been systematically fitted to laboratory experiments of such a press. The simulation results obtained agree very well with the results of various laboratory experiments.

1 Introduction.

Mechanical presses are widely used in motor industry for e.g. sheet metal forming of car body components. The working precision of such mechanical presses depends on e.g. dissipative friction effects in the various bearings, on clearances in the bearings, on elastic frame / ram distortions, on spatial ram displacements and on spatial ram tiltings ([1],[2]). In order to better understand these undesirable effects both, qualitatively and quantitatively, and to finally overcome these effects by suitable constructions of presses, a systematic theoretical and experimental analysis of their static and dynamic behavior is needed. To construct presses more stiff, various investigations of presses and of components of presses have been done based on FE–methods ([3],[4]). Finit–Element investigations provide usfull results for improving structural components of presses. On the other hand, they don't consider the dynamic effects of the forming process and its dependence on clearances, on tiltings and on displacements of the ram. This paper presents an alternative approach to analyse the dynamic behavior of a single–point–drive eccentric press (cf. 1) using a computer simulation of the press based on rigid body dynamics ([5],[6]). This computer simulation provides a basis for improving both, the design of mechanical presses and their working precision. In this paper three mathematical models of different complexity of a press (cf. 1 and Section 2) have been systematically fitted to different laboratory experiments. The laboratory experiments used have been performed by the LVWU (Laboratorium für Verfahren und Werkzeugmaschinen der Umformtechnik, University of Kassel, Prof. Dr. Wagener) using a 250 ton single–point–drive



(a) Mechanical Press (b) Schematic Drawing (c) Computer Graphic

Fig. 1: Foto, schematic drawing and computer graphic of a Single–Point–Drive Eccentric Press.

[1] This project (AiF 9618) has been supported by the European Resarch Council for Sheet Metal Forming (EFB).

eccentric press (cf. 1). The results of computer simulations and of laboratory experiments (Section 3) associated to those show excellent agreement.

2 Design of engineer models of the press

The press has been modeled by three different engineer models used in the fitting process. The first (simplest) model will be called **base model**, the second (more refined) model will be called **refined model** and the third model will be called **sophisticated model**.

The base engineer model includes the following **components** (cf. 2):

- three rigid bodies (single–axis eccentric drive (e), planar pitman (p), single–axis ram with tool (s)),
- several coupling elements
  - force coupling elements (spring of the workpiece (**W**) and damper (**D**) of the ram),
  - joints (three revolute joints, one translational joint).

The base engineer model of the press includes the following **model parameters** (cf. 2a and 3).

- Parameters of the rigid bodies: mass (m_i), moment of inertia ($J^{L_k}_{C_{k,y}}$) and geometry parameters ($x^R_{P_{ij}C_i}$, $z^R_{P_{ij}C_i}$, l_m) $(i = e, p, s;\ j = 1, 2, \ldots, N_i;\ k = e, p;\ m = p, s)$.
- Parameters of the coupling elements (c_W, d_W, l_W, d_D).

Computer simulations of the base engineer model provide time histories of the following **variables** (cf. 2b and 3) that also have been measured in laboratory experiments:

- single–axis rotation angle of the eccentric drive around an inertial fixed rotation axis ($\theta_{P_e} := \theta_{L_{P_e}R}$) through the center of gravity ($P_e := C_e$),
- displacement of the ram in z–direction ($z^R_{C_s}$),
- velocity of the ram in z–direction ($\dot{z}^R_{C_s}$),
- forming force in z–direction ($F^R_{W,z}$).

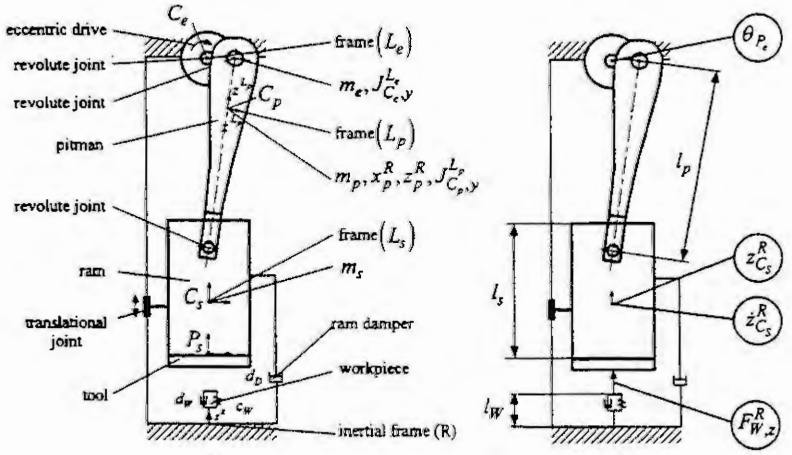
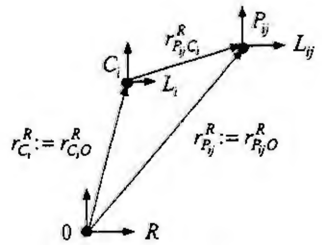


Fig. 2: Base engineer model.

The refined engineer model includes the following **components** (cf. 4):

- three rigid bodies (single–axis eccentric drive, spatial pitman, spatial ram with tool),
- coupling elements
  - force coupling elements (spring of the workpiece, 2 dampers of the ram and 8 block elements), (block elements (**B**):=spring–damper elements horizontally placed between ram and frame (**F**), simulating the elastic and dissipative contact forces between these two elements and the clearance between them),
  - joints (1 revolute joint and 2 spherical joints).



$$r^R_{C_i} := \left(x^R_{C_i}, y^R_{C_i}, z^R_{C_i}\right)^T \in \mathrm{IR}^3,$$

$$r^R_{P_{ij}} := \left(x^R_{P_{ij}}, y^R_{P_{ij}}, z^R_{P_{ij}}\right)^T \in \mathrm{IR}^3,$$

$$r^R_{P_{ij}C_i} := \left(x^R_{P_{ij}C_i}, y^R_{P_{ij}C_i}, z^R_{P_{ij}C_i}\right)^T \in \mathrm{IR}^3,$$

$$(i = e, p, s, F_1, F_2, \cdots, F_{18}; j = 1, 2, \cdots, N_i)$$

C:= center of gravity,
L:= body fixed frame,
R:= inertial frame,
P:= body fixed point

Fig. 3: Vector diagram including vectors and frames used.

This engineer model includes the following **model parameters** (cf. 4a and 3).

- Parameters of the rigid bodies: mass (m_i), moments of inertia ($J^{L_k}_{C_k,x}$, $J^{L_i}_{C_i,y}$, $J^{L_k}_{C_k,z}$, $J^{L_k}_{C_k,xy}$, $J^{L_k}_{C_k,xz}$, $J^{L_k}_{C_k,yz}$), and geometry parameters ($x^{L_i}_{P_{ij}C_i}$, $z^{L_i}_{P_{ij}C_i}$, l_k,), $(i = e, p, s;\ j = 1, 2, \ldots, N_i;\ k = p, s)$.
- Parameters of the coupling elements: (c_W, d_W, l_W, d_{D_i}, c_{B_j}, d_{B_j}, l_{B_j}), $(i = 1, 2;\ j = 1, 2, \ldots, 8)$.

Computer simulations of the refined engineer model provide time histories of the following **variables** which have also been measured in the laboratory experiment (cf. 4b and 3):

- single–axis rotational angle of the eccentric drive around an inertial fixed rotation axis (θ_{P_e}),
- spatial motions of the ram: translations ($x^R_{P_s}, y^R_{P_s}$) and rotations (φ_{P_s}, θ_{P_s}), (φ_{P_s} and θ_{P_s} are cardan angles around the x_{P_s} and y_{P_s} axes, respectively), (the location of the point $P_s \in P_{ij}$ is shown in Figure 8),
- forming force in z–direction ($F^R_{W,z}$).

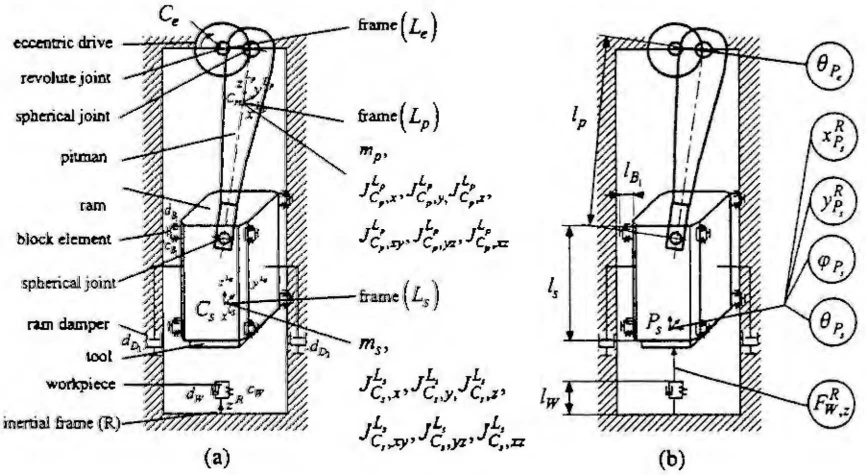


Fig. 4: Refined engineer model.

The sophisticated engineer model of the press includes the following **components** (cf. 5):

- 21 rigid bodies in space (spatial eccentric drive, spatial pitman, spatial ram with tool, 18 spatial rigid bodies representing the elastic frame of the press),
- coupling elements
  - force coupling elements (spring of the workpiece, 5 dampers of the ram, 8 block elements, 32 spatial springs and 32 spatial dampers of the frame of the press),
  - joints (1 revolute joint and 6 spherical joints).

This engineer model includes the following **model parameters** (cf. 5a).

- Parameters of the rigid bodies: mass (m_i), moments of inertia ($J^{L_i}_{C_i,x}$, $J^{L_i}_{C_i,y}$, $J^{L_i}_{C_i,z}$, $J^{L_i}_{C_i,xy}$, $J^{L_i}_{C_i,xz}$, $J^{L_i}_{C_i,yz}$), and geometry parameters ($x^{L_i}_{P_{ij}C_i}$, $y^{L_i}_{P_{ij}C_i}$, $z^{L_i}_{P_{ij}C_i}$, l_k), $(i = e, p, s, F_1, F_2, \ldots, F_{18};\ j = 1, 2, \ldots, N_i;\ k = p, s)$.
- Parameters of the coupling elements: (c_W, d_W, l_W, d_{D_i}, c_{B_j}, d_{B_j}, l_{B_j}, $c_{F_m,x}$, $c_{F_m,y}$, $c_{F_m,z}$, $d_{F_m,x}$, $d_{F_m,y}$, $d_{F_m,z}$, $c_{F_m,\varphi}$, $c_{F_m,\theta}$, $c_{F_m,\psi}$, $d_{F_m,\varphi}$, $d_{F_m,\theta}$, $d_{F_m,\psi}$,), $(i = 1, 4;\ j = 1, 2, \ldots, 8;\ m = 1, 2, \ldots, 32)$.

Computer simulations of the sophisticated engineer model provide time histories of the following variables which have also been measured in the laboratory experiment (cf. 5b and 3):

- rotational angle of the eccentric drive around an inertial fixed rotation axis (θ_{P_e}),
- spatial motions ($x^R_{P_s}$, $y^R_{P_s}$) and rotations (φ_{P_s}, θ_{P_s}) of the ram,
- forming force in $z-$direction ($F^R_{W,z}$).

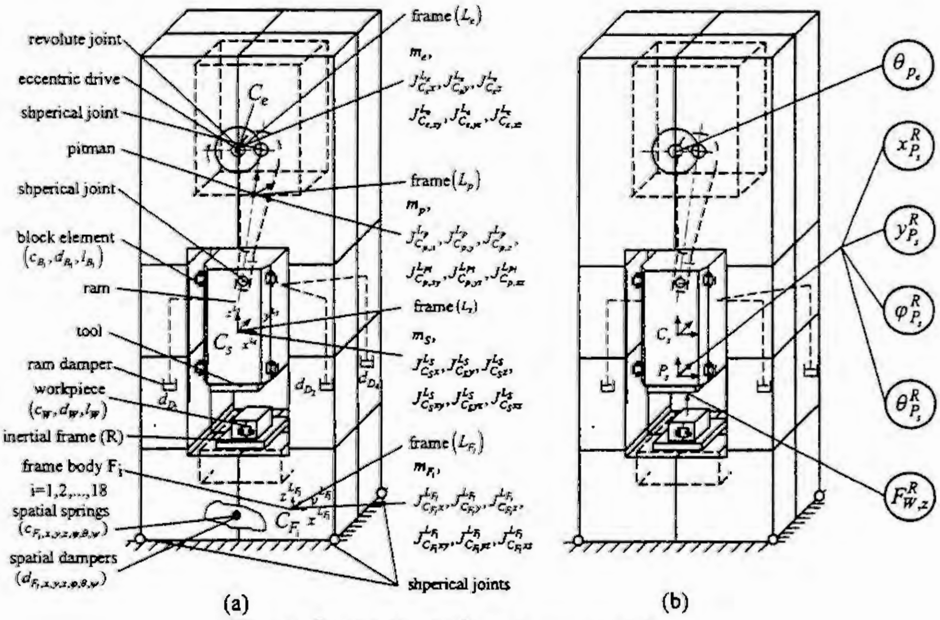

revolute joint
eccentric drive
shperical joint
pitman
shperical joint
block element
$(c_{B_i}, d_{B_i}, l_{B_i})$
ram
tool
ram damper
workpiece
(c_W, d_W, l_W)
inertial frame (R)
frame body F_i
i=1,2,...,18
spatial springs
$(c_{F_i,x,y,z,\varphi,\theta,\psi})$
spatial dampers
$(d_{F_i,x,y,z,\varphi,\theta,\psi})$
C_e
C_S
C_{F_i}
d_{D_1}
d_{D_3}
d_{D_4}
frame(L_e)
m_e,
$J^{L_e}_{C_e x}, J^{L_e}_{C_e y}, J^{L_e}_{C_e z}$
$J^{L_e}_{C_e,xy}, J^{L_e}_{C_e,yz}, J^{L_e}_{C_e,xz}$
frame(L_p)
m_p,
$J^{L_p}_{C_{p,x}}, J^{L_p}_{C_{p,y}}, J^{L_p}_{C_{p,z}}$,
$J^{L_{pi}}_{C_{p,xy}}, J^{L_{pi}}_{C_{p,yz}}, J^{L_{pi}}_{C_{p,xz}}$
frame(L_s)
m_S,
$J^{L_S}_{C_S x}, J^{L_S}_{C_S y}, J^{L_S}_{C_S z}$,
$J^{L_S}_{C_S xy}, J^{L_S}_{C_S yz}, J^{L_S}_{C_S xz}$
frame(L_{F_i})
m_{F_i},
$J^{L_{F_i}}_{C_{F_i}x}, J^{L_{F_i}}_{C_{F_i}y}, J^{L_{F_i}}_{C_{F_i}z}$,
$J^{L_{F_i}}_{C_{F_i}xy}, J^{L_{F_i}}_{C_{F_i}yz}, J^{L_{F_i}}_{C_{F_i}xz}$
shperical joints
C_s
P_s
θ_{P_e}
$x^R_{P_s}$
$y^R_{P_s}$
$\varphi^R_{P_s}$
$\theta^R_{P_s}$
$F^R_{W,z}$
(a)
(b)

Fig. 5: Sophisticated engineer model.

3 Fitting of engineer models of the press to laboratory experiments

In this section the engineer models of the previous section will be fitted step by step to laboratory experiments (cf. 6).

In a **first step** a laboratory experiment has been performed were the vertical motion $z^R_{C_S}$ and velocity $\dot{z}^R_{C_S}$ of the ram of the press together with the forming force in $z-$direction $F^R_{W,z}$ have been recorded during a retardation experiment (cf. 7a). This laboratory experiment has been modeled and simulated by the base engineer model of Section 2 using the initial condition $\dot{\theta}_{P_e}(t=0)$. The model parameters m_i, $J^{L_k}_{C_{k,v}}$, $J^{L_k}_{C_{k,v}}$, $x^R_{P_{ij}C_i}$, $z^R_{P_{ij}C_i}$ and l_m have been calculated from press data. The model parameters c_W, d_W, d_D and the initial condition $\dot{\theta}_{P_e}(t=0)$ were unknown. The results of a first computer simulation with suitably chosen model parameters

$$c_W = 1 \cdot 10^7 [\frac{N}{m}] \quad , \quad d_W = 1 \cdot 10^1 [\frac{Ns}{m}] \quad , \quad d_D = 1 \cdot 10^5 [\frac{Ns}{m}] \quad , \quad \dot{\theta}_{P_e}(t=0) = 50[\frac{\circ}{s}] \tag{1}$$

are shown in Figure 7b together with the results of associated laboratory experiments in Figure 7a. The results of this computer simulation do not accurately model the laboratory experiment. In a first fitting loop (cf. 6) the unknown model parameters have been identified as

$$c_W = 3.1 \cdot 10^7 [\frac{N}{m}] \quad , \quad d_W = 0.6 \cdot 10^1 [\frac{Ns}{m}] \quad , \quad d_D = 2.35 \cdot 10^4 [\frac{Ns}{m}] \quad , \quad \dot{\theta}_{P_e}(t=0) = 97[\frac{\circ}{s}]. \tag{2}$$

Together with these model parameters the base engineer model provides results (cf. 7c) that accurately model the laboratory experiment (cf. 7a). The amplitutes of the velocity ($\dot{z}^R_{C_s}$) of the retardation experiment and of the computer simulation obtained for model parameters (1) and (2) decrease exponentially. The time histories of Figure 7b obtained by computer simulations with model parameters (1)

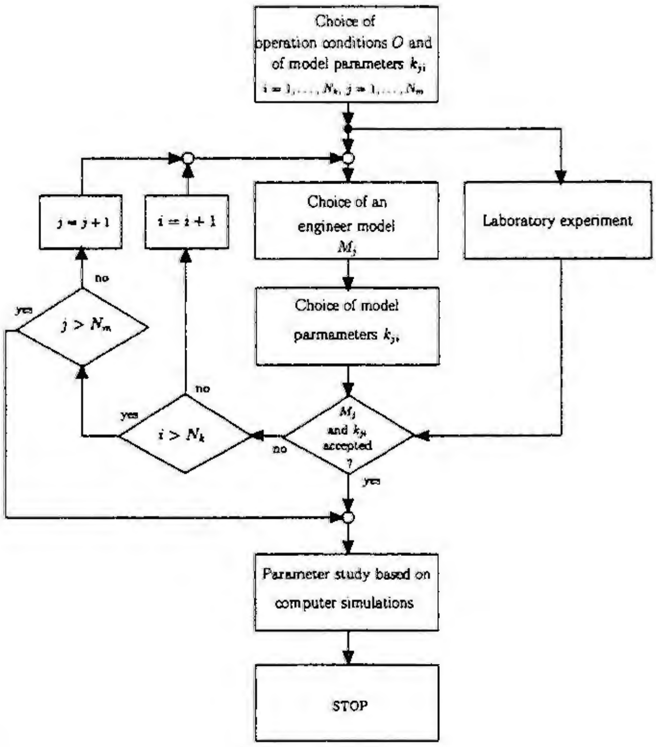

Choice of operation conditions O and of model parameters k_{ji}
$i = 1, \ldots, N_k, j = 1, \ldots, N_m$
$j = j + 1$
$i = i + 1$
Choice of an engineer model M_j
Laboratory experiment
Choice of model parmameters k_{ji}
no
yes
$j > N_m$
no
yes
$i > N_k$
no
M_j and k_{ji} accepted ?
yes
Parameter study based on computer simulations
STOP

Fig. 6: Flow diagram of the model fitting procedure.

deviate from the laboratory experiments (cf. 7a). The time histories of Figure 7c obtained by computer simulations with model parameters (2) agree very well with the laboratory experiments (cf. 7a). The non differentiable behavior of the measured time histories $\dot{z}^R_{C_s}(t)$ of Figure 7a results from static friction effects of the sensor used in the laboratory experiments.

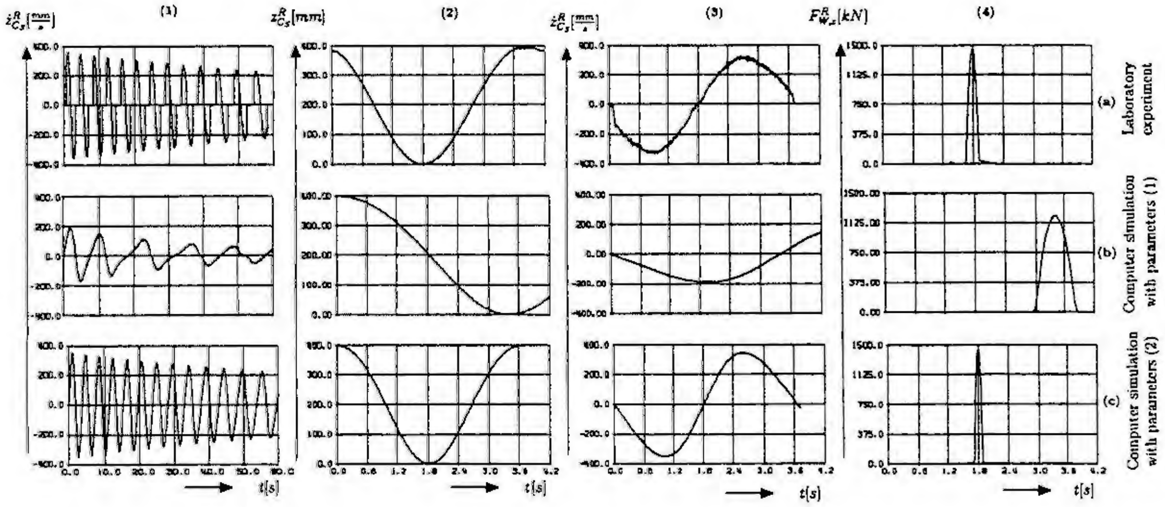


Fig. 7: Comparison of laboratory experiments with computer simulations of the base engineer model obtained from retardation experiments.

The base engineer model of the press together with the model parameters (2) will be included as submodels in the refined engineer models of the next steps.

In a **second laboratory experiment** the coordinates φ_x and φ_y of **spatial ram tiltings** (ram rotations around inertial fixed rotation axes) and the coordinates x_d and y_d of **spatial ram displacements** have been measured with respect to point P_s on the ram (cf. 8). Here P_s is the origin of frame L_{P_s} with axes orientated in parallel to the axes of R, where P_s is located on the center of the bottom side of the ram.

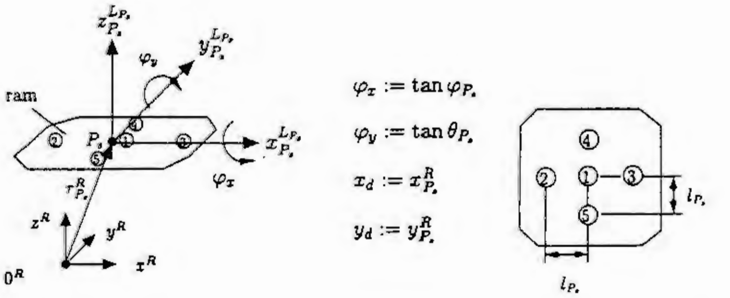


Fig. 8: Location of reference point P_s of the ram displacements (x_d, y_d) and of the ram tiltings (φ_x, φ_y) for different locations (1,2,3,4,5) of the workpiece relative to the ram.

This laboratory experiment has been repeated for different locations (1,2,3,4,5) of the workpiece relative to the ram (cf. 8). Due to the restricted space of this paper only two time histories $\varphi_x(F_{W,z}^R)$ and $x_d(\theta_{P_e})$ of the recordings $\varphi_x(F_{W,z}^R)$, $\varphi_y(F_{W,z}^R)$, $x_d(\theta_{P_e})$ and $y_d(\theta_{P_e})$ are shown in Figure 9 for each of the five load points.

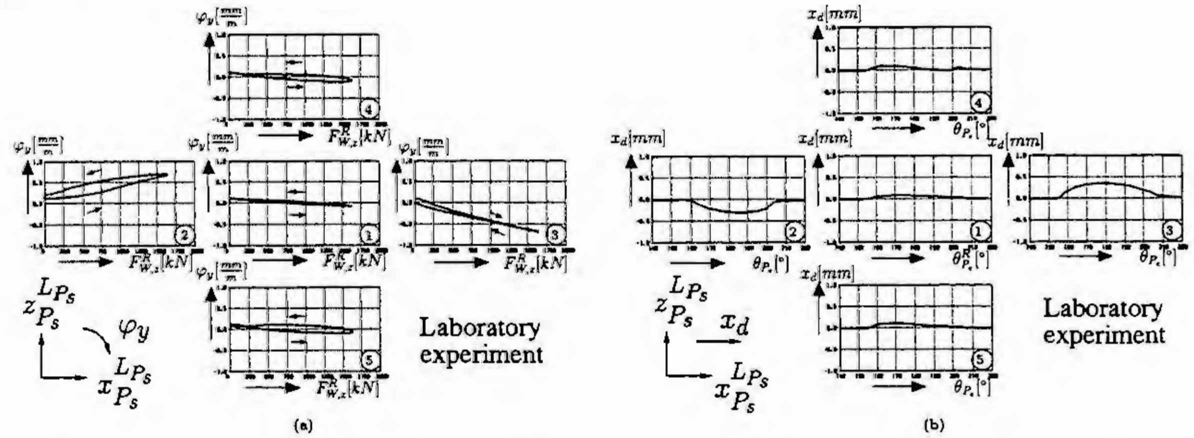


Fig. 9: Laboratory experiments showing ram tilting φ_y and ram displacement x_d for each of the locations, 1,2,3,4,5 of the workpiece relative to the ram (cf. 8).

In the subsequent discussion only two of the five diagrams associated to the five locations (1,2,3,4,5) of the workpiece relative to the ram of Figure 9a and 9b will be compared with computer simulations (cf. 10b, 10c and 10d). These diagrams are associated to the locations 3 and 4 of the workpiece relative to the ram. The results of the laboratory experiments shown in Figure 9 cannot be simulated by the base engineer model. Instead the second engineer model has been used (Section 2). The model parameters m_i, $J_{C_k,x}^{L_k}$, $J_{C_i,y}^{L_i}$, $J_{C_k,z}^{L_k}$, $J_{C_k,xy}^{L_k}$, $J_{C_k,xz}^{L_k}$, $J_{C_k,yz}^{L_k}$, $x_{P_{ij}C_i}^R$, $z_{P_{ij}C_i}^R$ and l_k of the refined engineer model have been calculated from the geometry of the press. The model parameters c_W, d_W, l_W and $d_{D_i} = \frac{1}{2}d_D$ and the initial condition $\dot{\theta}_{P_e}(t=0)$ were taken from the base engineer model. The remainder model parameters c_{B_j}, d_{B_j} and l_{B_j} of the refined engineer model have been suitably chosen in a first fitting step. The computer simulation results obtained by these parameters

$$c_{B_j} = 5 \cdot 10^8 [\frac{N}{m}] \quad , \quad d_{B_j} = 1 \cdot 10^4 [\frac{Ns}{m}] \quad , \quad l_{B_j} = 0.002[m] \quad , \qquad (j = 1, 2, \ldots, 8) \qquad (3)$$

are shown in Figure 10b. They do not agree with the associated laboratory experiments of Figure 10a.

In a next step the model parameters of the refined engineer model have been fitted to the laboratory experiment. They are

$$
\begin{aligned}
c_{B_i} &= 1.75 \cdot 10^8[\frac{N}{m}] \quad , \quad d_{B_k} = 8.0 \cdot 10^6[\frac{Ns}{m}] \quad , \qquad\qquad (4)\\
c_{B_j} &= 7.0 \cdot 10^8[\frac{N}{m}] \quad , \quad l_{B_k} = 0.00212[m] \quad , \quad (i = 1,2,3,4; j = 5,6,7,8; k = 1,2,\ldots,8).
\end{aligned}
$$

The results of computer simulations of the refined engineer model together with the fitted model parameters (4) are shown in Figure 10c. A comparison of Figure 10a and 10c shows a very good agreement of the maximum values of the tiltings $\varphi_x(F^R_{W,z})$ and $\varphi_y(F^R_{W,z})$ and of the bending of the curve $\varphi_x(F^R_{W,z})$ in forward stroke. In backward stroke the tilting $\varphi_x(F^R_{W,z})$ and the displacements $x_d(\theta_{P_e})$ and $y_d(\theta_{P_e})$ decrease faster than in the laboratory experiment.

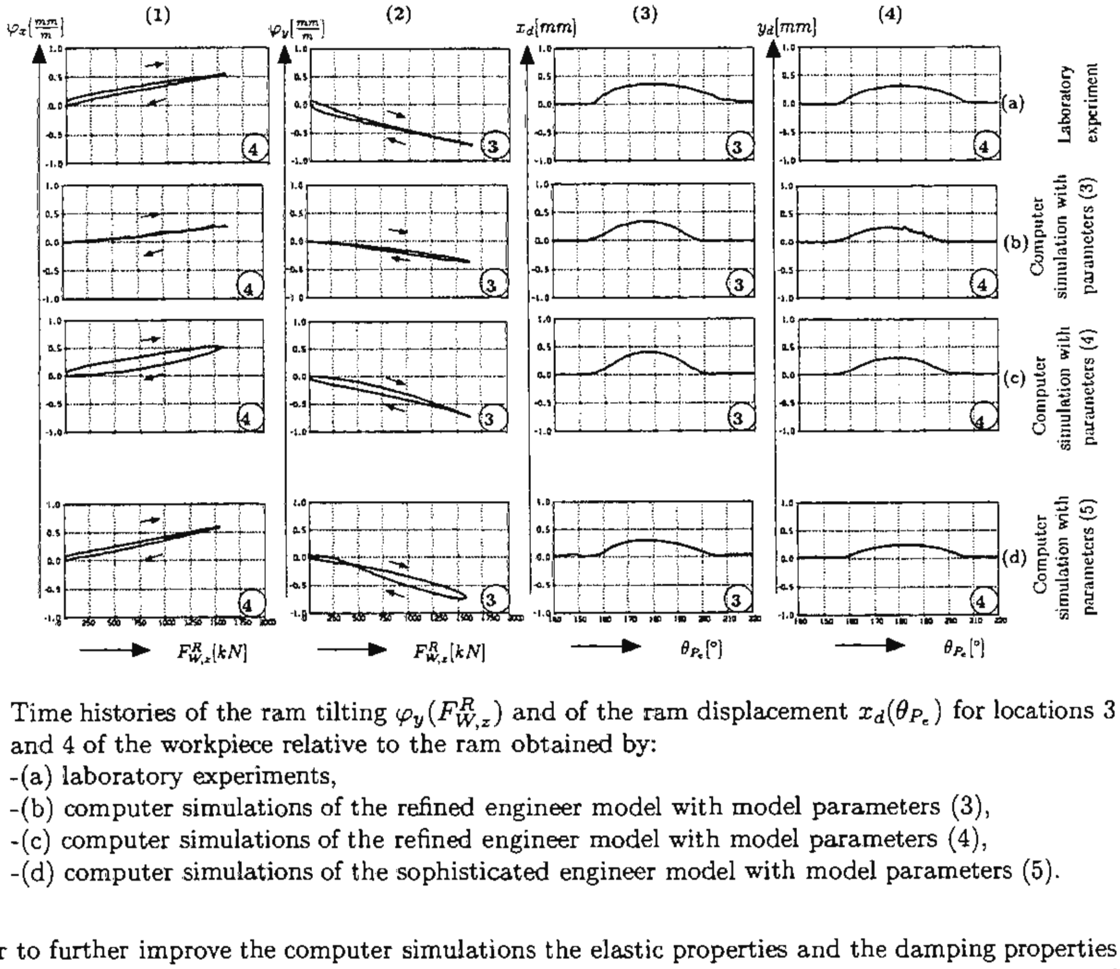


Fig. 10: Time histories of the ram tilting $\varphi_y(F^R_{W,z})$ and of the ram displacement $x_d(\theta_{P_e})$ for locations 3 and 4 of the workpiece relative to the ram obtained by:
-(a) laboratory experiments,
-(b) computer simulations of the refined engineer model with model parameters (3),
-(c) computer simulations of the refined engineer model with model parameters (4),
-(d) computer simulations of the sophisticated engineer model with model parameters (5).

In order to further improve the computer simulations the elastic properties and the damping properties of the frame of the press have been included into the press model. This provides the **sophisticated engineer model** (Section 2). The model parameters of the sophisticated engineer model obtained in an additional fitting process are

$$
\begin{aligned}
c_{B_i} &= 6 \cdot 10^{11}[\frac{N}{m}] \quad , \quad c_{B_j} = 2 \cdot 10^{11}[\frac{N}{m}] \quad , \quad d_{B_k} = 1 \cdot 10^9[\frac{Ns}{m}] \quad , \quad l_{B_k} = 0.00212[m],\\
c_{F_m,x} &= c_{F_m,y} = 1.5 \cdot 10^8[\frac{N}{m}] \quad , \quad c_{F_m,z} = 4 \cdot 10^9[\frac{N}{m}] \quad , \quad d_{F_m,x} = d_{F_m,y} = d_{F_m,z} = 1 \cdot 10^6[\frac{Ns}{m}],\\
c_{F_m,\varphi} &= c_{F_m,\theta} = c_{F_m,\psi} = 4 \cdot 10^7[\frac{Ns}{m}] \quad , \quad d_{F_m,\varphi} = d_{F_m,\theta} = d_{F_m,\psi} = 1 \cdot 10^5[\frac{Ns}{m}], \qquad (5)\\
&(i = 1,2,\cdots,4; \quad j = 5,6,\cdots,8; \quad k = 1,2,\cdots,8; \quad m = 1,2,\cdots,32).
\end{aligned}
$$

The Final simulation results obtained by the sophisticated engineer model with model parameters (5) are collected in Figure 10d. The bends of the ram–tilting kurves $\varphi_x(F^R_{W,z})$ and $\varphi_y(F^R_{W,z})$ and the displacements $x_d(\theta_{P_e})$ and $y_d(\theta_{P_e})$ of the ram show a further improved agreement between computer simulations and laboratory experiments.

Figure 11 shows computer animation graphics of the forward and backward stroke of the ram of the press based on computer simulation results of the refined engineer model. To visualize the ram tiltings (φ_x, φ_y) and the ram displacements (x_d, y_d), these motions have been plotted in an extremly enlarged scale in Figure 11.

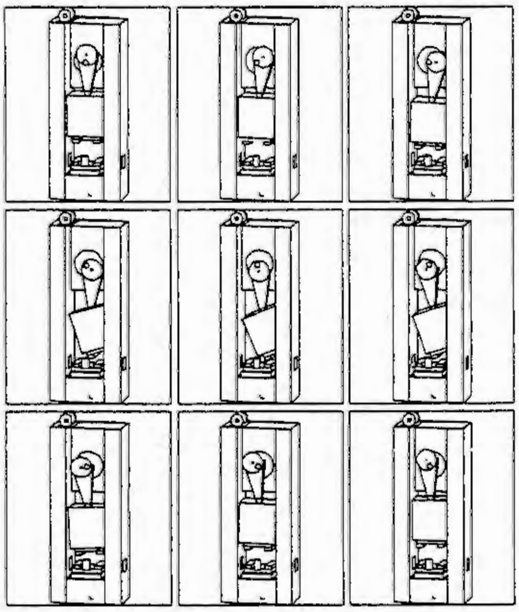


Fig. 11: Computer animation graphics of the refined engineer model shown for a single press stroke of the ram (one forward and one backward stroke) with ram tiltings (φ_x, φ_y) and ram displacements (x_d, y_d) of location 3 of the workpiece represented in an extremly enlarged plotting scale .

The results of Figures 7 and 10 show that the press models have been successfully fitted to the different laboratory experiments.

The sophisticated engineer model including a model of the elastic frame of the press provides results of highest accuracy. On the other hand the computation time needed to provide these results was much bigger than the computation time needed by the refined engineer model. As a consequence most of the various parameter variations performed have been done by the refined engineer model. Some of those results are shown in Figure 12. Each line of this Figure includes the same diagrams $\varphi_x(F^R_{W,z})$, $\varphi_y(F^R_{W,z})$, $x_d(\theta_{P_e})$ and $y_d(\theta_{P_e})$ as Figure 10. The rows include computer simulation results obtained by parameter variations

(a) of the eccentric load position l_{P_s} of the workpiece relative to the ram ($l_{P_s} = 40[mm]$ to $l_{P_s} = 600[mm]$), defined in Figure 8,

(b) of the damping constant d_{B_i} of the block elements between the ram and the frame of the press ($d_{B_i} = 0.26 \cdot 10^6[\frac{Ns}{m}]$ to $d_{B_i} = 3.2 \cdot E7[\frac{Ns}{m}]$),

(c) of the length l_p of the pitman ($l_p = 50\%$ to $l_p = 200\%$), (cf. 4),

(d) of the heights l_s of the ram ($l_s = 50\%$ to $l_s = 200\%$), (cf. 4),

(e) of the initial velocity $\dot{\theta}_{P_e}$ of the eccentric drive ($\dot{\theta}_{P_e} = 60[\frac{\circ}{s}]$ to $\dot{\theta}_{P_e} = 600[\frac{\circ}{s}]$,

obtained for locations 2 and 5 of the workpiece relative to the ram (cf. 8).

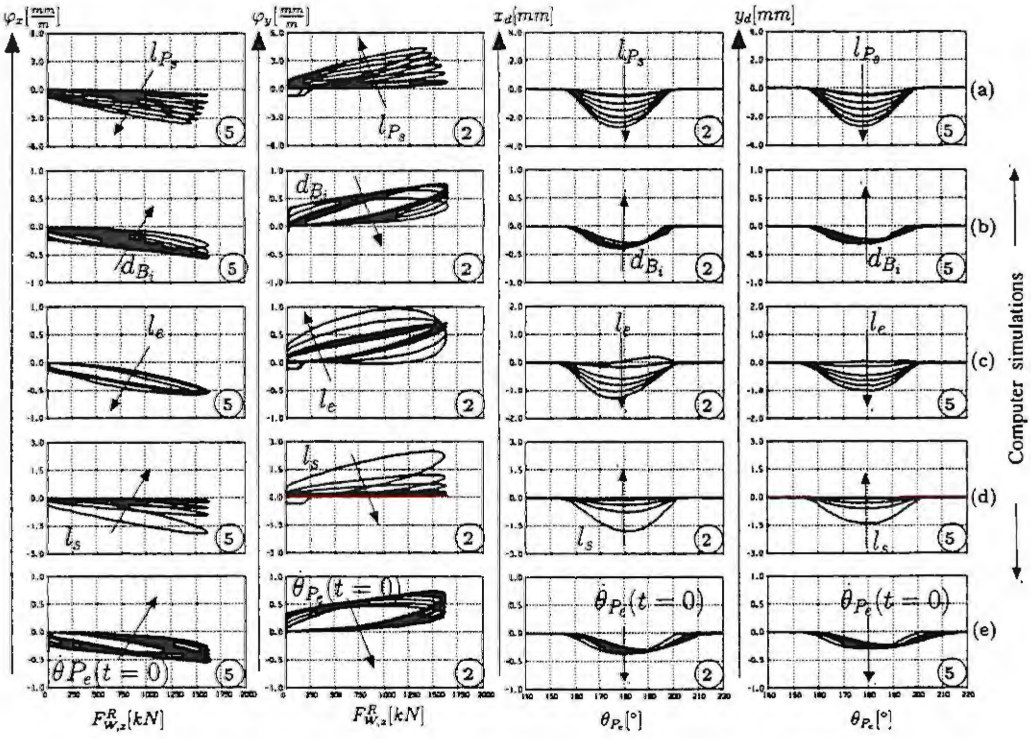


Fig. 12: Study of model parameter variations in computer simulations of the second engineer model for location 2 ($\varphi_y(F^R_{W,z})$, $x_d(\theta_{P_e})$) and location 5 ($\varphi_x(F^R_{W,z})$, $y_d(\theta_{P_e})$) of the workpiece relative to the ram.

4 Conclusion

In this paper three mathematical models of different complexity of a press (cf. 1 and Section 2) have been systematically fitted to different laboratory experiments. The results of the computer simulations agree very well with the results of the laboratory experiments (Section 3).

References

1. Wagener, H.W.; Schlott, C. Influence of Die Guidance Systems on the Angular Deflection of Press Slide and Die under Eccentric Loading. Journal of Mechanical Working Technology 20, S.463/475. 1989.
2. Bogon, P. Einflußgrößen auf die dynamische Federung von Exzenterpressen. Dissertation, Laboratorium für Verfahren und Werkzeugmaschinen der Umformtechnik, Universität Kassel. 1991.
3. Haft, G.; Löwen, J. Finite Elemente Berechnung großer Strukturen am Beispiel einer Pressenberechnung. Thyssen Berichte, Heft 7/78, S. 51–57. 1978.
4. Burchert, H. Untersuchung der Steifigkeit mechanischer Pressen in geschlossener Bauart mit der Finite-Elemente-Methode. Dissertation, Institut für Umformtechnik und Umformmaschinen, Technische Universität Hannover. 1980.
5. H. Hahn. Mathematische Modelle masseloser Gelenke zur Simulation räumlicher Bewegungen von Starrkörpersystemen. IMAT Bericht RT-1, Fachgebiet Regelungstechnik (Maschinenbau), Universität Kassel. 1988.
6. H. Hahn. Einführung in die Theorie räumlicher Bewegungen von Starrkörpersystemen. Skript TMB 2, Fachgebiet Regelungstechnik (Maschinenbau), Universität Kassel. 1988.

NONLINEAR MODEL STRUCTURE IDENTIFICATION USING GENETIC PROGRAMMING

Gary J. Gray, David J. Murray-Smith, Yun Li & Ken C. Sharman
Centre for Systems & Control and Deptartment of Electronics & Electrical Engineering
University of Glasgow, Glasgow, Scotland G12 8LT
G.GrayǀD.Murray-SmithǀY.LiǀK.Sharman@eng.gla.ac.uk

Abstract. Genetic Programming [1] is a nonlinear structure optimisation procedure. It can optimise an equation or model structure by minimising some error criterion. In this paper, the Genetic Programming algorithm is configured to build a block diagram simulation model of a dynamic system. The sum of the squares of the error between the simulation output and recorded experimental data is minimised by the algorithm to give a nonlinear block diagram description of the dynamic system. System parameters are estimated by a combined simulated annealing-simplex method. The technique is applied to a coupled water tank system and a representative model is identified.

Introduction

Identification of nonlinear physical models presents many problems since both the structure and parameters of the physical model need to be determined. If physical understanding of the identified model structure is to be improved, any prior knowledge of the type of dynamics likely to exist in that system must be part of the model optimisation process. This prior knowledge can take the form of known system dynamics or suspected structural features. Often a trial and error approach is adopted to choose between a number of candidate models. Possible structures are deduced from engineering knowledge of the system and the parameters of these models are estimated from available measured experimental data. Automation of this process would mean that a much larger range of potential model structures could be investigated. Genetic Programming (GP) is an optimisation method which can be used to optimise the nonlinear structure of a dynamic system by selecting model structure elements from a database.

GP works by emulating natural evolution to generate a model structure that best maximises (or minimises) some fitness function. A population of model structures (represented as trees as in Figure 1) evolves through many generations towards the solution using certain evolutionary operators and a "survival-of-the-fittest" selection scheme.

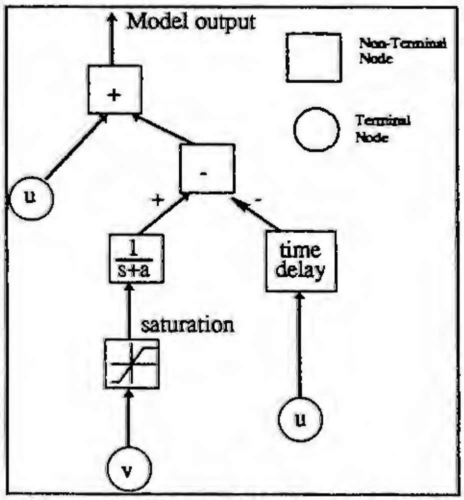


Figure 1 GP tree for a typical block diagram function

Each individual tree is of variable length, is constructed of nodes and represents one candidate model structure for the system. The nodes can be terminal nodes at the end of a branch signifying an input or a constant, or non-terminal nodes representing functions performing some action on one or more signals within the structure to produce an output signal. In Figure 1, the terminal nodes are the system inputs u and v, and the non-terminal nodes are functions from the library selected for this dynamic system. The library consists of functions which could conceivably form part of the dynamic system. The nodes are used to build the trees according to grammar rules specifying the number of inputs of each node-type. Each tree is evaluated by simulating the corresponding dynamic model to give some fitness function defining the quality of that model with respect to experimental data.

The genetic programming population contains typically a few hundred trees and evolves through the action of operators known as crossover, mutation and selection. The initial population is created entirely at random. The tree strucutre is limited only by the GP grammar and the maximum tree size. Crossover and mutation processes are applied to branches, i.e. that part of the structure down from a randomly selected point in the tree. Crossover is applied to perhaps 80% of each generation and involves the branches from two parent trees being interchanged. This means that characteristics of both parents will survive to the next generation but are combined differently, sometimes leading to offspring fitter than either parent. Mutation, at a low rate, means the creation of a completely new branch determined at random. This procedure is less likely to improve a specific structure but it can help the algorithm to escape from a local minimum. Selection involves evaluating the fitness of each population member and selecting the fittest to succeed.

The fitness function could be the sum of the squares of the error or a correlation function [2]. There are various selection strategies [1]. The selection method used in this paper is tournament selection. A number of trees (perhaps 5~10) is selected from the population and the best of this group survives whilst the worst die off. Crossover, mutation and selection improve the general fitness of the population and the algorithm repeats through many generations until some convergence criterion is satisfied. The model can then be used for further investigation of the physical system or to validate the structure of an existing model developed in some other way.

GP has been applied to several nonlinear modelling tasks including the development of signal processing algorithms [3][4] and the identification of chemical processes [5][6]. In the areas of continuous time system identification, GP has been used to identify the nonlinear differential equation describing a dynamic system [7]. The application of a block diagram oriented simulation method to GP optimisation is introduced in [8].

In this paper, a method of genetic programming for structural modelling using block diagram simulation is described. The method is then applied to the identification of a coupled water tank system. The coupled water tank is identified as a single-input-single-output system using the GP approach.

Method

Nonlinear physical models are often portrayed using block diagram constructs. Many block diagram based simulation tool are available (e.g. SIMULINK[9] which is a simulation toolbox for MATLAB[10]). Such tools use a library of system elements both linear and nonlinear to construct a block diagram of the dynamic system to be simulated. A wide range of dynamic systems can be represented using simulation tools of this kind. In this paper, the GP is used to develop a SIMULINK block diagram of the dynamic system under investigation.

The GP algorithm can choose from a library of SIMULINK blocks. The grammar inherent in the GP tree ensures that the blocks are connected logically. The block diagram representation of the model evolves as the GP algorithm minimises the fitness function, $f=\Sigma e^2$, where e is the error between model output and experimental data. This process eventually produces a nonlinear block diagram description of the system.

The GP optimisation program is written in C++ and the SIMULINK block diagram simulations are run from MATLAB. The C++ code is based on a program called gpc++ [11]. It controls the GP part of the program. The terminal nodes in this case are system inputs and the non-terminal nodes are standard library function system blocks. To evaluate an expression tree, that tree must be converted to a SIMULINK block diagram. To do this, the C++ program writes a SIMULINK script file describing the dynamic system and this SIMULINK file is executed from a MATLAB engine process running alongside the C++ executable on a UNIX workstation. To make the program faster, the SIMULINK models are compiled using the accelerator option before running.

Many of the blocks contain numerical parameters. These could be the coefficients of the transfer functions, a gain value or in the case of a time delay, the delay itself. It is necessary to identify the numerical parameters of each nonlinear model before simulating it and evaluating its fitness. The models are randomly generated and will therefore contain linearly dependent parameters and parameters which have no effect on the output. Because of this, gradient based methods cannot be used. The method chosen was a combination of Nelder-Simplex and simulated annealing [12].

Simulated annealing optimises by a method which is analogous to the cooling process of a metal. As a metal cools, the atoms organise themselves into an ordered minimum energy structure. The amount of vibration or movement in the atoms is dependent on temperature. As the temperature decreases, the movements, though still random, become smaller in amplitude and as long as the temperature decreases slowly enough, the atoms order themselves into the minimum energy structure. In simulated annealing, the parameters start off at some random value and they are allowed to change their values within the search space by an amount related to a quantity defined as system "temperature". If a parameter change improves overall fitness, it is accepted, if it reduces fitness it is accepted with a certain probability. The temperature decreases according to some pre-determined "cooling" schedule and the parameter values should converge to some solution as the temperature drops.

In this application, the simulated annealing process is combined with Nelder-simplex optimisation. The simplex is an (n+1) dimensional shape where n is the number of parameters. This simplex explores the search space slowly by changing its shape around the optimum solution. It closes in on the optimum of the function converging on the solution. The simulated annealing adds a random component and the temperature scheduling to the simplex algorithm thus improving the robustness of the method.

If the system cools too fast, the parameters will not have time to adjust to their optimum and the optimisation will fail. If it cools too slowly, the optimisation will take longer to execute. The optimum cooling schedule depends on the number of parameters and the nature of the problem. For this problem, the cooling schedule varies according to the number of parameters being estimated. The temperature starts at 1.0 and reduces according to;

$$t_n = t_{n-1} \times (0.6 + 0.02n_p) \qquad (1)$$

where n_p is the number of parameters. Four iterations are performed at each temperature until the temperature reaches 0.01.

After some generations (perhaps about 25) a solution block diagram of the system evolves. This block diagram often contains a lot of redundant information. For example, a complete branch could be nullified by a gain of zero or by being subtracted from a copy of itself. After manual post-processing, the result is an algebraic expression describing the unmodelled part of the system with estimated values of any numerical parameters [8].

Most of the execution time of the program is in the evaluation of the fitness functions. This is time consuming because it includes estimation of the numerical parameters. To speed up this process, each tree is stored in a linked list along with its fitness and before evaluation of a new tree the list is searched to see if its fitness has already been calculated. This seems to occur in about 25% of cases giving a substantial speed increase for the optimisation.

Example

The GP structure identification technique was applied to the modelling of a coupled water tank system. The system consists of two coupled water tanks with a connecting pipe between them, an input to the first tank, and an outlet pipe from the second tank. The system is nonlinear and is shown in Figure 2.

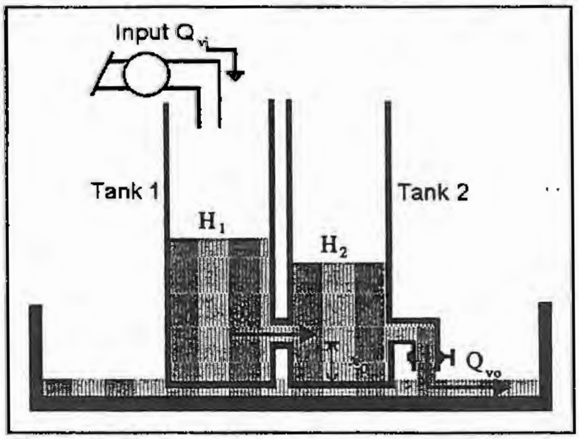


Figure 2 Twin tank system

Q_{vi} is the flow into tank one, Q_{v1} is the flow from tank one to tank two, and Q_{vo} is the output flow from tank two. A_1 and A_2 are the cross-sectional areas of tanks one and two respectively. H_1 is the depth of the water in tank one and H_2 is the depth in tank two. It is known that for this system, the flow between the tanks and particularly in the output pipe is nonlinear and represents the main modelling error. A function library with relevance to the physical dynamics of the problem is required. Table 1 lists possible nodes.

Table I Function library for genetic programming structure estimation routine.

Function	inputs	parameters
Non-terminal nodes		
+	2	0
-	2	0
gain	1	1
1st order lag	1	2
$\omega_n{}^2 / s^2+2\omega_n\zeta+\omega_n{}^2$	1	0
lead-lag	1	2
$(s^2+2\omega_n\zeta+\omega_n{}^2)/$ $(s^2+2\omega_n\zeta+\omega_n{}^2)$	1	4
$(s+a) / (s^2+2\omega_n\varsigma+\omega_n{}^2)$	1	3
integrator	1	0
time delay	1	1
Terminal node		
input	-	0

The input signal was a step of magnitude 3.7×10^{-5}m^3/sec. The GP allowed a dynamic expression for flow out of tank two with respect to the input to tank one to be developed, including numerical parameters estimated using the combined simplex- simulated annealing method. The expression was coded as a SIMULINK diagram and simulated. The sum of the squares of the error between predicted H_2 and measured H_2 was calculated. A log function (Equation (2)) was used to convert this to a fitness function for the GP.

$$\mathit{fitness} = 5000\times\left(8 - \log_{10}(\Sigma e^2)\right) \tag{2}$$

This conversion was used because the sum of square error function for integrated differential equations can cover a very wide range. The scaling is not important since a tournament based selection scheme was used.

Results Analysis

The GP produced the dynamic model ilustrated in Figure 3.

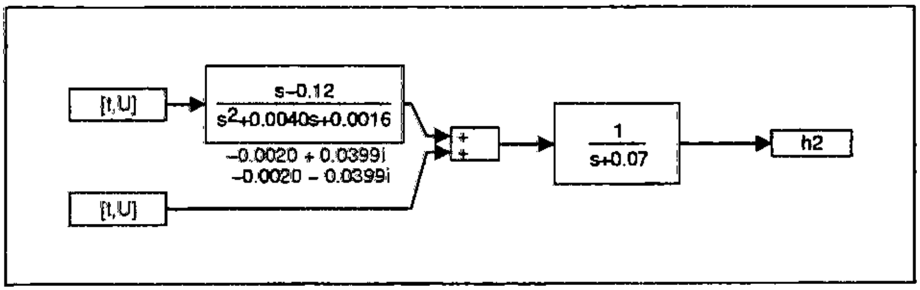


Figure 3 GP optimised model of twin tank system

The resulting model often contains redundant information, and it is then necessary to simplify the system. This was not necesary for this system. Figure 3 shows the system to have second order dynamics with a non-minimum phase characteristic and a first order filter. The extra input signal added to the output of the transfer function block is negligible because it is of very low magnitude (3.7×10^{-5}). The fit for H_2 is of a reasonable quality in dynamic response (Figure 4) indicating that the model described in Figure 3 is a good representation of this system.

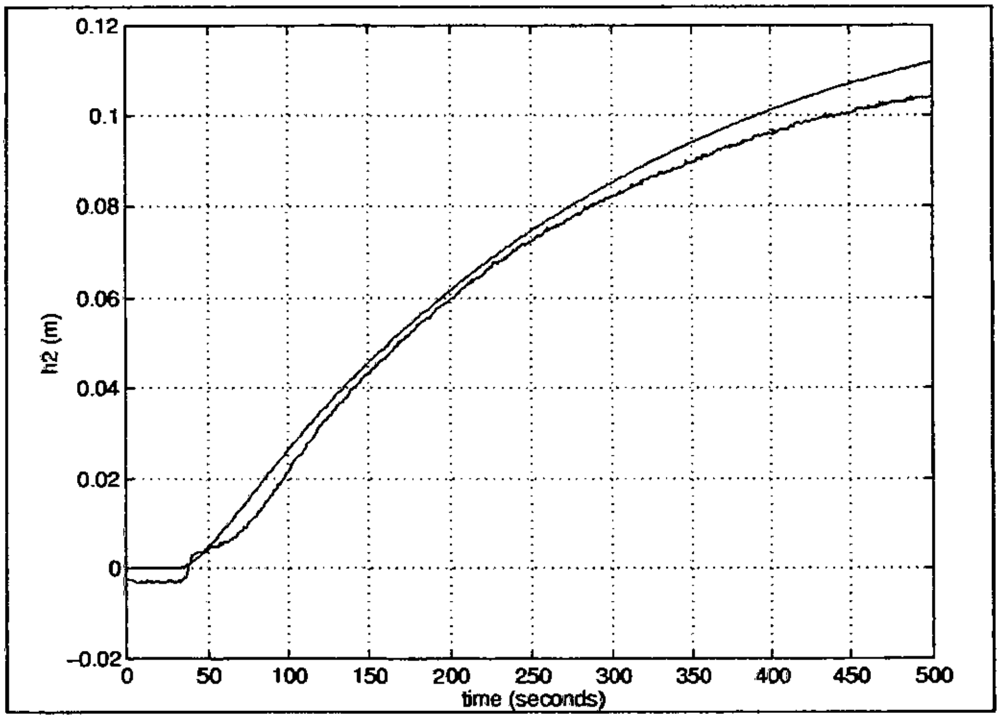


Figure 4 Time response of GP evolved model with experimental data

Discussion

Genetic Programming is a valuable tool for the modelling of nonlinear dynamic systems. It can develop nonlinear model structures that best fit experimental data. The GP automates the trial and error process of structure estimation and can therefore test a lot more potential model components and structures. It is also capable of retaining the best parts of these structures and recombining them to give new and sometimes better models.

The selection of the library functions is important. It has been found that if the functions are too general, the GP produces a model which is an empirical fit to the data. This can be desirable in some cases but the main objective is often to achieve physically meaningful models. Specific nonlinear elements such as look up tables or hard discontinuities can be implemented even if their importance is doubtful. Experience with this method has shown that specific nonlinearities not present in the system being modelled will not occur in the GP solution.

The model resulting from the GP can reveal some aspects of the physical structure of the system. The dynamics of the system are described by the GP solution so the GP optimisation method can select the order of the model as well as any nonlinearities present. Such information can be used for further investigation of the physical system or to validate the structure of an existing model developed in some other way.

Conclusions

Genetic programming is a useful technique for identification of the structure of nonlinear dynamic models. Used with a carefully selected library of functions, it can reveal information about the physical structure of a system and produce an accurate model describing the system. Including more complex functions in the library can reveal if that function is indeed present in the model.

The technique was applied to the identification of a coupled water tank system. A simple nonlinear structure was found that gave an accurate representation of the system and some insight into the structural dynamics of the system.

Acknowledgements. This research is supported by the UK Engineering and Physical Sciences Research Council under grant GR/K24987 ("Evolutionary Programming for Nonlinear Control"). The authors would like to thank the other members of the evolutionary computing research group for their useful discussions. The core program for the genetic programming is called gpc++ and was written by A.P. Fraser (1994).

References

1. Koza, J., Genetic Programming: On the programming of computers by means of natural selection, The MIT Press, 1992.

2. McKay, B., Willis, M.J., Hiden, H.J., Montague, G.A., Barton, G.W., Identification of industrial processes using genetic programming. Proceedings of Identification in Engineering Systems, Swansea, UK, 1996.

3. Sharman, K.C., Esparcia-Alcázar, A.I. & Li, Y., Evolving signal processing algorithms by genetic programming, Proceedings of IEE/IEEE GALESIA '95, Sheffield, England, 1995, pp.473-480.

4. Sharman, K.C., Esparcia-Alcázar, A.I., Some Applications of Genetic Programming in Digital Signal Processing, In: Late breaking papers at the Genetic Programming '96 conference, Stanford, CA.

5. Bettenhausen, K.D., Marenbach, P., Freyer, S., Rettenmaier, H. and Nieken, U., Self-organizing structured modelling of a biotechnological fed-batch fermentation by means of genetic programming. Proceedings of IEE/IEEE GALESIA '95, England, 1995, pp.481-486.

6. Marenbach, P., Bettenhausen, K.D., Signal Path Oriented Approach for Generation of Dynamic Process Models. Genetic Programming '96 conference, Stanford, CA.

7. Gray, G.J., Murray-Smith, D.J., Li, Y., and Sharman, K.C., Nonlinear model structure identification using genetic programming. In: Late breaking papers at the Genetic Programming '96 conference, Stanford, CA.

8. Gray, G.J., Murray-Smith, D.J., Li, Y., and Sharman, K.C., Structural System Identification using genetic programming and a block diagram oriented simulation tool. Electronics Letters, Vol. 32, No. 15, 1996, 1422-1424.

9. SIMULINK User's guide, The MathWorks, Inc.,Mass. USA, 1991-2.

10. MATLAB Reference guide, The MathWorks, Inc.,Mass. USA, 1992.

11. Fraser, A.P., 1994, 'Genetic Programming in C++', University of Salford, Salford, England.

12. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., Numerical Recipes in C, 2nd Edition, Cambridge University Press, 1992, Chapter 10.

MODELLING AND IDENTIFICATION FOR HIGHLY NONLINEAR PROCESSES

M. Boutayeb[1,2], M. Darouach[2] and P. M. Frank[1]

[1] University of Duisburg
Gerhard Mercator Universität GH Duisburg, 81-BB, 47048 GERMANY.
[2] University of Henri Poincaré - Nancy I - CRAN CNRS URA 821
186, rue de Lorraine, 54400 FRANCE. Email : boutayeb@iut-longwy.u-nancy.fr

Abstract

This paper is devoted to modelling and identification of non linear dynamic systems. At first, we propose a new input-output representation to describe highly and/or large-scale non linear processes. The proposed mathematical model is non linear in parameters and is written as a product of several polynomials which may be selected in a sequential approach. In the second part of this note, a simple and recursive identification technique is detailed. It is shown that, under strong persistently exciting condition, global convergence of the parameters estimation algorithm is guaranteed. One of the main results of this contribution is that parameters to be estimated are considerably reduced in comparison with the general Kolmogorov-Gabor structure.

1. Introduction

As most of industrial plants are characterized by non linear behaviours, many research activities were focused on non linear models developments in order to increase the accuracy and the performances of control schemes. There exist several mathematical representations, to describe non linear processes, such as the Hammerstein model, composed of a static non linearity in series with a linear dynamic system [11], exponential time series models of Ozaki [12], trigonometric and rational models [2]. For more details the reader is referred to Haber and Unbehauen [6]; Billings, Chen and Korenberg [1]; Kortman, Janiszowski and Unbehauen [7] papers and the references inside. For highly non linear processes Eykhoff [5] have introduced a general discrete time model, the so-called the Kolmogorov-Gabor polynomial model. A stochastic version was developed by Leontarities and Billings [10]. Unfortunately, the proposed model is linear in parameters with non linear data and consequently a large number of parameters have to be considered in order to describe adequately the non linear behaviour of the process over the entire range of operating conditions. Identification algorithms of this non linear model may lead, especially for large scale systems, to considerable computational requirements with possibly numerical instabilities and convergence problems. Some selection techniques of the significant terms have been proposed [8]-[9] and [13]. In this note, a new structure for modelling highly and/or large-scale non linear processes is presented. The proposed model may be seen as a reduced form of Kolmogorov-Gabor polynomials. One of the main features of this representation is that parameters to be estimated are considerably reduced in comparison with the extended form, since the model we propose is written as a product of linear dynamic polynomials. Consequently, the obtained form is non linear in parameters. In the second part of this paper, a simple identification algorithm is developed. The main property of the proposed approach is that parameters are estimated in a recursive way while the global convergence is guaranteed, also a global version of the algorithm is established. The proposed method consists first to transform the non linear representation into an input-output model linear in parameters, what for a regular transformation based on the pseudo-inverse technique allows us to estimate in the least squares sense parameters vector of the original realization. Sufficient conditions for global convergence are derived. Finally, a numerical example to show the accuracy and the performances of the proposed approach is provided.

2. Problem formulation

The general Kolmogorov-Gabor polynomial is represented by a series expansion of time-shifted input and output signals to approximate non linear model outputs. For notation simplicity, only SISO systems of order p are considered :

$$\begin{aligned} y_k = \bar{y} + \sum_{i=1}^{n} a_{1i} y_{k-i} + \sum_{j=1}^{m} b_{1j} u_{k-j} + \sum_{i=1}^{n}\sum_{j=i}^{n} c_{ij} y_{k-i} y_{k-j} + \sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij} y_{k-i} u_{k-j} \\ + \sum_{i=1}^{m}\sum_{j=i}^{m} e_{ij} u_{k-i} u_{k-j} + \ldots + \sum_{i=1}^{n} \cdots \sum_{j=r}^{n} f_{i\ldots j} \underbrace{y_{k-i} \cdots y_{k-j}}_{\text{p times}} + \ldots + \sum_{i=1}^{m} \cdots \sum_{j=r}^{m} g_{i\ldots j} \underbrace{u_{k-i} \cdots u_{k-j}}_{\text{p times}} \end{aligned} \qquad 1$$

$\bar{y}$ is a mean value, u_{k-i} and y_{k-j} denote the input and output data at time instant k-i and k-j respectively. As can be expected, the number of possible parameters to be estimated is considerable, especially for high order p.

The aim of this contribution consists in building a class of the general Kolmogorov-Gabor polynomial in a very reduced form. The non linear model of order p that we propose here is of the form :

$$y_k = \left(A_1 y_k + B_1 u_k\right)\left[1 + \left(A_2 y_k + B_2 u_k\right)\left[1 + \left(A_3 y_k + B_3 u_k\right)\ldots\left[1 + \left(A_p y_k + B_p u_k\right)\right]\ldots\right]\right] \qquad 2$$

with $A_i = a_{i1} q^{-1} + \ldots + a_{in_i} q^{-n_i}$ and $B_i = b_{i1} q^{-1} + \ldots + b_{im_i} q^{-m_i}$

q^{-1} is the delay operator, $a_{i1}, a_{i2}, \ldots, a_{in_i}$ and $b_{i1}, b_{i2}, \ldots, b_{im_j}$ are the parameters to be estimated of the ith sub-model, i = 1, ..., p. Owing to a lack of space, model structure determination will not be treated in the sequel. Orders p, n_i and m_i are then assumed to be known. For simplicity and without the loss of generality, we assume that $n_i = n$ and $m_i = m$. Before we give the main result of this note, it should be mentioned that all the terms in the general Kolmogorov-Gabor structure are in (2). Indeed, let us write the output system y_k in the extended form:

$$\begin{aligned} y_k = {} & \left(A_1 y_k + B_1 u_k\right) + \left(A_1 y_k + B_1 u_k\right)\left(A_2 y_k + B_2 u_k\right) + \ldots \\ & \left(A_1 y_k + B_1 u_k\right)\left(A_2 y_k + B_2 u_k\right)\ldots\left(A_p y_k + B_p u_k\right) \end{aligned} \qquad 3$$

or equivalently :

$$\begin{aligned} y_k = {} & \sum_{i=1}^{n} a_{1i} y_{k-i} + \sum_{j=1}^{m} b_{1j} u_{k-j} + \sum_{i=1}^{n}\sum_{j=i}^{n} c_{ij} y_{k-i} y_{k-j} + \sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij} y_{k-i} u_{k-j} \\ & + \sum_{i=1}^{m}\sum_{j=i}^{m} e_{ij} u_{k-i} u_{k-j} + \ldots + \sum_{i=1}^{n} \ldots \sum_{j=r}^{n} f_{i\ldots j} y_{k-i} \ldots y_{k-j} + \ldots + \sum_{i=1}^{m} \ldots \sum_{j=r}^{m} g_{i\ldots j} u_{k-i} \ldots u_{k-j} \end{aligned} \qquad 4$$

parameters $c_{ij}, d_{ij}, e_{ij}, \ldots, f_{i\ldots j}, g_{i\ldots j}$ are non linear functions of a_{iq} and b_{js} for i, j = 1, ..., p; q = 1, ..., n and s = 1, ..., m. We notice that parameters number, which is equal to $\sum_{i=1}^{p}(n_i + m_i)$, is very weak in comparison with those of the extended form (4). The goal of this contribution is to provide a simple and recursive procedure to estimate parameters a_{iq} and b_{js} of the proposed input-output model (2) and in the same time to ensure global convergence while the model is a non linear one.

3. Main result

The approach that we consider here, consists first in writing the non linear representation (2) into the form of (4) which is linear in $c_{ij}, d_{ij}, e_{ij}, \ldots, f_{i\ldots j}, g_{i\ldots j}$. what for a regular transformation based on the pseudo-inverse technique allows us to estimate in the least squares sense parameters vector of the original realization.

To make the paper more understandable, we consider first a 2-order model so that we obtain all parameters dependencies. Next, a general formula to deduce original parameters is provided.

Consider the following noisy bilinear system :

$$y_k = \bar{y} + \left(A_1 y_k + B_1 u_k\right) + \left(A_1 y_k + B_1 u_k\right)\left(A_2 y_k + B_2 u_k\right) + \varepsilon_k \qquad 5$$

that we transform into a signal vector :

$$y_k = \varphi_k^T \theta + \varepsilon_k \qquad 6$$

with $\varphi_k^T = \left(1 \;\; \varphi_{yk}^T \; \varphi_{uk}^T \; \varphi_{yyk}^T \; \varphi_{yuk}^T \; \varphi_{uuk}^T\right)$, $\theta^T = \left(\bar{y} \;\; \theta_{A_1}^T \;\; \theta_{B_1}^T \;\; \theta_C^T \; \theta_D^T \; \theta_E^T\right)$

$$\theta_{A_1}^T = \left(a_{11} \ldots a_{1n}\right), \; \theta_{B_1}^T = \left(b_{11} \ldots b_{1m}\right)$$

$$\theta_C^T = \left(c_{11}\, c_{12} \ldots c_{1n}\, c_{22}\, c_{23} \ldots c_{2n} \ldots c_{(n-1)(n-1)}\, c_{(n-1)n}\, c_{nn}\right)$$

$$\theta_D^T = \left(d_{11}\, d_{12} \ldots d_{1\max(n,m)} \ldots d_{\min(n,m)1} \ldots d_{\min(n,m)\max(n,m)}\right)$$

and $\theta_E^T = \left(e_{11}\, e_{12} \cdots e_{1m}\, e_{22}\, e_{23} \cdots e_{2n} \cdots e_{(m-1)(m-1)}\, e_{(m-1)m}\, e_{mm} \right)$

where e_k denotes approximation errors as well as additional disturbances.
On an horizon time of length N, (6) is equivalent to :

$$Y_{k,N} = \Phi_{k,N}\theta + \varepsilon_{k,N} \qquad 7$$

with $Y_{k,N} = \begin{pmatrix} y_k \\ y_{k+1} \\ . \\ . \\ y_{k+N} \end{pmatrix}$, $\Phi_{k,N} = \begin{pmatrix} \varphi_k^T \\ \varphi_{k+1}^T \\ . \\ . \\ \varphi_{k+N}^T \end{pmatrix}$ and $\varepsilon_{k,N} = = \begin{pmatrix} \varepsilon_k^T \\ \varepsilon_{k+1}^T \\ . \\ . \\ \varepsilon_{k+N}^T \end{pmatrix}$

Under the assumption that $\Phi_{k,N}$ is of full column rank, the least squares estimation of θ is given by :

$$\hat{\theta} = \left(\Phi_{k,N}^T \Phi_{k,N} \right)^{-1} \Phi_{k,N}^T Y_{k,N} \qquad 8$$

It could be mentioned that parameters of the linear part A_1 and B_1 are directly obtained from $\hat{\theta}$. In what follows, we give a simple procedure to deduce $\hat{A}_2$, $\hat{B}_2$ from $\hat{\theta}_C$, $\hat{\theta}_D$, $\hat{\theta}_E$. Indeed, we notice that parameters c_{ij}, d_{ij} and e_{ij} are in the form :

$$c_{ii} = a_{1i}a_{2i}$$
$$c_{ij} = a_{1i}a_{2j} + a_{2i}a_{1j} \text{ for } i \neq j$$
$$d_{ij} = a_{1i}b_{2j} + a_{2i}b_{1j}$$
$$e_{ii} = b_{1i}b_{2i}$$

and $e_{ij} = b_{1i}b_{2j} + b_{2i}b_{1j}$ for $i \neq j$

that we can write in the form :

$$M_{A_1,B_1}\theta_{A_2,B_2} = \theta_{C,D,E} \qquad 9$$

where M_{A_1,B_1} is a matrix, of full column rank, linear in the parameters of A_1 and $B_1$and :

$$\theta_{A_2,B_2} = \begin{pmatrix} a_{21} \\ . \\ a_{2n} \\ b_{21} \\ . \\ b_{2m} \end{pmatrix} \text{ and } \theta_{C,D,E} = \begin{pmatrix} c_{11} \\ . \\ c_{nn} \\ d_{11} \\ . \\ e_{mm} \end{pmatrix}$$

The least squares estimation of θ_{A_2,B_2} is then given by :

$$\hat{\theta}_{A_2,B_2} = \hat{M}_{A_1,B_1}^{+}\hat{\theta}_{C,D,E} \qquad 10$$

where parameters of A_1, B_1, C, D and E are replaced by their estimates. $\hat{M}_{A_1,B_1}^{+}$ is the moore-penrose pseudo-inverse of $\hat{M}_{A_1,B_1}$ with $\hat{M}_{A_1,B_1}^{+} = \left(\hat{M}_{A_1,B_1}^T \hat{M}_{A_1,B_1} \right)^{-1} \hat{M}_{A_1,B_1}^T$.

Along the lines of this procedure, we establish then the following general result. First, let us decompose the extended parameters vector θ into p sub-vectors as follows :

$$\theta = \begin{pmatrix} \theta_1^T & \theta_2^T & . & . & \theta_p^T \end{pmatrix}^T \qquad 11$$

where θ_s denotes the sth sub-vector of order s i.e. each component of θ_s is a polynomial composed of the parameters of A_1, B_1, ..., A_s and B_s of order s.

The first step, consists in computing $\hat{\theta}$ from (8) and consequently, we obtain an estimation of θ_1 i.e. parameters of the linear part A_1 and B_1 and also an estimation of θ_2, ..., θ_p. In the second step, we investigate all parameters dependencies in order to obtain the following system :

$$M_{\theta_{s-1}} \theta_{A_s,B_s} = \theta_s$$

θ_{A_s,B_s}, corresponding to the parameters vector of A_s and B_s, is then computed by the sequential approach :

$$\hat{\theta}_{A_s,B_s} = \hat{M}^{+}_{\theta_{s-1}} \hat{\theta}_s \qquad \text{for } s = 2, ..., p \qquad 12$$

It could be noticed that the inversion of $\left(\Phi^T_{k,N} \Phi_{k,N} \right)$ may lead to high computational requirements with possibly numerical instabilities due to truncation errors and/or ill conditioned matrices. To avoid this problem a simple and recursive parameters estimation algorithm is established in the following theorem, whose proof may be developed along the lines of our recent work [3].

Theorem

Under the following assumptions :

1- u_k is a strong persistently exciting sequence

2 - ε_k is a zero mean white noise

the following recursive parameters estimation algorithm :

$$\hat{\theta}_{k+1} = \hat{\theta}_k + K_{k+1}(y_{k+1} - \varphi^T_{k+1} \hat{\theta}_k) \qquad 13$$

$$K_{k+1} = P_k \varphi_{k+1} (\varphi^T_{k+1} P_k \varphi_{k+1} + R_{k+1})^{-1} \qquad 14$$

$$P_{k+1} = (I - K_{k+1} \varphi_{k+1}) P_k \qquad 15$$

$$\hat{\theta}_{(k+1)A_s,B_s} = \hat{M}^{+}_{(k+1)\theta_{s-1}} \hat{\theta}_{(k+1)s} \qquad \text{for } s = 2, ..., p \qquad 16$$

ensures that $\lim_{k \to \infty} E(\hat{a}_{ijk} - a_{ij}) = \lim_{k \to \infty} E(\hat{b}_{irk} - b_{ir}) = 0$ w. p. 1

where the symbol E(.) denotes the mean value and $\hat{a}_{ijk}$, $\hat{b}_{irk}$ represent the estimates of a_{ij} and b_{ir} respectively at time instant k.

Remarks

1- The proposed parameters estimation algorithm may be extended to the case of coloured noises [3] or to the case of bounded disturbances [4].

2- The strong persistently exciting condition means that u_k and its powers are persistently exciting, this implies that $\Phi_{k,N}$ is of full column rank, for details see [14].

3- Instead of using the pseudo-inverse technique, (16) may also be computed in a recursive way.

4. Simulation results

The numerical example that we consider here is a third-order stochastic dynamic system with $A_i = a_{i1} q^{-1}$ and $B_i = b_{i1} q^{-1}$; i=1, 2, 3. The input signal u_k is a zero mean white noise sequence with unit standard deviation k = 1, ..., N=500. The added measurement noise ε_k is a zero mean white noise so that the Signal to Noise Ratio (SNR) is taken as 2. SNR is defined as : $\text{SNR} = \left(\frac{\text{var}(y_{ak})}{\text{var}(\varepsilon_k)} \right)^{1/2}$ where y_{ak} is the undisturbed output signal.

As initial conditions, each parameter is initialised at zero and the error covariance matrix $P_0 = 10^5 I$, where I is the identity matrix. Table 1 shows high performances of the proposed technique inspite of the very low value of the SNR.

Parameters	Actual value	Estimation
a_{11}	-0.2400	-0.2157
b_{11}	-0.2800	-0.2699
a_{21}	-0.9000	-0.9029
b_{21}	0.4000	0.4160
a_{31}	0.3000	0.3535
b_{31}	0.5000	0.4839

Table 1

5. Conclusion

This paper deals with modelling and identification of non linear systems. At first, we propose a new structure, written as a product of linear dynamic polynomials, to model highly and/or large scale non linear processes. One of the main features of this representation is that parameters to be estimated are considerably reduced in comparison with the general Kolmogorov-Gabor polynomial. In the second part, a simple and recursive identification algorithm was established. It is shown that under weak conditions global convergence is ensured. The accuracy and the performances of the proposed approach was shown through a numerical example with a very low SNR.

Acknowledgement. We thank the Alexander Von Humboldt Foundation for supporting this work.

References

1. Billings, S. A., Chen, S. and Korenberg, M., Identification of MIMO non linear systems using a forward regression orthogonal estimator. Int. J. Control, 49, 1989, 2157-2189.
2. Billings, S. A. and Chen, S., Identification of non linear rational systems using a prediction error estimation algorithm. Int. J. Sys. Sciences, 1989, 20, 467-494.
3. Boutayeb, M., Rafaralahy H. and Darouach, M., A robust and recursive identification method for the Hammerstein model. IFAC World Congress San-Francisco, 1996.
4. Boutayeb, M. and Darouach, M., A robust and recursive identification method for a class of non linear systems. IFAC Belfort-France, 1997 to appear.
5. Eykhoff, P., System Identification. Wiley : New York, 1974.
6. Haber, H. and Unbehauen, H., Structure Identification of Non linear Dynamic systems - A survey on Input/output Approaches. Automatica, 16, 1990, 651-677.
7. Kortmann, M., Janiszowski, K. and Unbehauen, H., Application and comparison of different identification schemes under industrial conditions. Int. J. Control, 48, 1988, 2275-2296.
8. Kortmann, M. and Unbehauen, H., Structure detection in the identification of non linear systems. APII, 22, 1988, 5-25.
9. Kortmann, M. and Unbehauen, H., Two algorithms for model structure determination for non linear dynamic systems with application to industrial processes. 8th IFAC SISPE, Peking, 1988, 939-946.
10. Leontaritis, I. J. and Billings, S. A., Input-Output parametric models for non linear systems. Part I : Deterministic non linear systems. Part II : Stochastic non linear systems. Int. J. Control, 41, 1985, 303-344.
11. Narendra, K. S. and Gallmann, P. G., An iterative method for the identification of non linear systems using a Hammerstein model. IEEE Trans. Auto. Control AC 11, 1966, 546-550.
12. Ozaki, T., Non linear time series models and dynamical systems. handbook of statistics, 1985, 25-83.
13. Pottmann, M., Unbehauen H. and Seborg, D. E., Application of a general multi-model approach for identification of highly non linear processes - a case study. Int. J. Control, Vol. 57, N° 1, 1993, 97-120.
14. Stoica, P. and Söderström, T., Instrumental variable methods for identification of Hammerstein systems. Int. J. Control, 35, 1982, 459-476.

TOWARDS LESS SENSITIVE IDENTIFICATION OF A SUBMARINE NONLINEAR DYNAMIC MODEL

S. Ziani-Cherif - G. Lebret
Ecole Centrale de Nantes, B.P. 92101
Laboratoire d'Automatique de Nantes, U.M.R.-C.N.R.S. No 6597,
1 rue de la Noë, 44321 Nantes cedex 3, France
E-mail : Salim.Ziani(Guy.Lebret)@lan.ec-nantes.fr

Abstract In order to perform precise tasks, the use of (nonlinear) dynamic model has showed to be very relevant for robots manipulators. On the basis of this model some identification attempts have been successfull (see [11],[16]). For submarine vehicles, the main tasks such as tracking and positioning also need good knowledge of their dynamic model. So we address in this paper a method for the identification of a submarine dynamic model. Naturally, for such a plant, external perturbations coming from water flows and weather conditions, as well as lacks in process modelling, are of large contribution when operating. So the identification method has to decrease their effects. We do so by designing and tracking some trajectories that are generated with regard to these perturbations.

1 Introduction

In order to design a control law, one needs a model of the process to be controlled and an "estimation" of its parameters. However, the natural nonlinear behaviour of a submarine vehicle due to hydrodynamic forces, and the external perturbations, such as wind effects, water flows ... explain that the model used for the controller design is usually very rough. In such a case, a robust control law is classically used. Moreover, it is obvious that this robust control will be more efficient when plant model is more well known. That is why we want to go deeper in the knowledge of the behaviour of such a plant. So we developp an approach for the identification of submersible with the two following specificities : the nonlinearity of the dynamic model and the perturbations properties. This will need a presentation of the mathematical model of our plant (section 2). Then, in section 3, the main known approaches of identification are presented with an attempt of classification. This allows us to introduce our approach as an alternative solution. As a consequence, the exciting trajectories design problem is discussed in section 4 with its correlation with system identification (SI) robustness. As an application, simulation results are adressed in section 5 and our conclusions are held last in section 6.

2 Dynamic model of submarine vehicle

As is usual for mechanical systems, the dynamic model of a submarine vehicle can be issued from the Newton-Euler's laws. This leads to the so-called dynamic model, expressing torque (Γ) versus reaction forces [8],[9]:

$$\left\{ \begin{array}{l} M.\dot{\nu} + C(\nu).\nu + D(\nu).\nu + g(\eta) = \Gamma \\ \dot{\eta} = J(\eta).\nu \end{array} \right. \tag{1}$$

with (η) the position and orientation vector, and (ν) the linear and angular velocities vector
M: mass and inertia components matrix
$C(\nu)$: Coriolis and centripetal components matrix
$D(\nu)$: Drag components
$g(\eta)$: Static (gravity and buyancy) components matrix

$C(\nu)$ and $D(\nu)$ matrices depend upon velocities components and hydrodynamic coefficients. M, $C(\nu)$, $D(\nu)$ and $g(\eta)$ are the matrices whose coefficients are to be estimated. The Jacobian matrix $J(\eta)$ expresses the relation between any choosed external reference frame and that settled to the vehicle.

3 System identification in marine applications

The present section resumes the essential approaches of identification in marine applications field.

3.1 Conventional testing technique

This technique is used by hydrodynamicians when testing on captive models (see [6], [12], [13], [14]). The vehicle is fixed on a planar motion mechanism (PMM). Tests are either made directly on vehicle or on a scaled model for larger vehicles. These tests consist in oscillating motions injected to the 'model'. Analysis of the varying oscillatory mode and speed of the PMM yields hydrodynamic coefficients. This technique is known to be very expensive both in time consuming and material necessities.

3.2 Automatician's classical approach

This method differs basically from the first one. In this case, some outputs (y) are obtained either by exciting the plant (Θ_p) by an input (U_p) or a model of it (U_m, Θ_m). Minimizing inputs difference w.r.t. parameters yields model coefficients (Θ_m) (fig. 1). In marine areas, this has been applied for underwater vehicles by Goheen [12]. Åström [3], Åström and Kallström [5], and Abkowitz [1] used the same approach for large ships identification. However, fundamental specificities of this approach are that the model is **linear** in its dynamics, and that choosed inputs are usually PRBS. As is shown in next section, our approach is basically different since we use a nonlinear model and another kind of inputs.

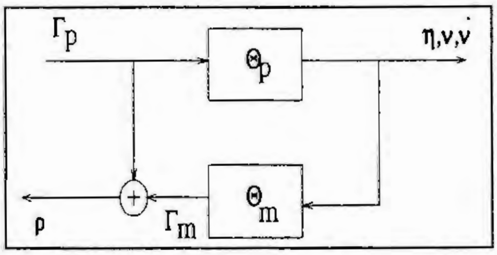


Figure 1: *Input-error scheme for SI.*

3.3 Nonlinear system identification

It can be shown that the nonlinear model (1) remains linear w.r.t the hydrodynamic coefficients. Indeed, it's useful to formulate it in a regression matrix form:

$$\Gamma_{(6,1)} = W(\nu, \dot{\nu})_{(6,p)}.\Theta_{(p,1)} \tag{2}$$

where
(Θ) is the vector of the model parameter to be estimated (p:number of parameters)
(W) is the regression matrix depending on orientation, velocities and acceleration components.
If $\Theta^* = \Theta_p$ are the 'true' (process) parameters, a torque Γ_p applied to the process gives:

$$(\nu_p, \dot{\nu}_p) = f(\Theta^*, \Gamma_p)$$

Since our model is written:

$$\Gamma_m = W(\nu, \dot{\nu}).\Theta_m$$

the ideal solution would be to find Θ_m such that $\Gamma_p = [\Gamma_m = W(\nu_p, \dot{\nu}_p).\Theta_m]$.

Using ordinary least square (LS) algorithm, the problem is formulated (fig.1) as:

$$min\Big\{\rho_{|\Theta_m} = ||\Gamma_m - \Gamma_p||\Big\}$$

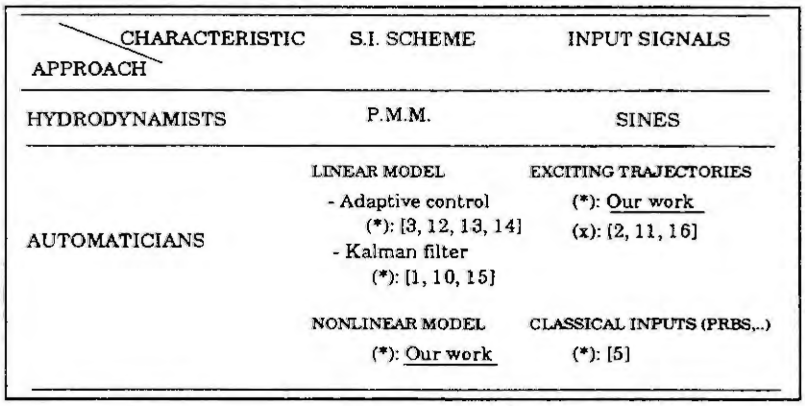
The analytical solution is found to be:

$$\Theta_m = W(\nu_p, \dot{\nu}_p)^{\dagger}.\Gamma_p \tag{3}$$

where $W^{\dagger}$ is the *pseudo − inverse* of (W).

Knowing that LS algorithm suffers from bias problem, our aim is to find a procedure to decrease this bias. We do so in what can be called an estimation "sensitivity approach" (robustness against perturbations). In fact, finding the exact solution of eq. (3) implies some constraints on regression matrix (W) i.e. its inversibility, condition number,..etc. Such constraints, since they relate some process properties, allow us to design the so−called 'exciting trajectories', or similarly exciting input, since torque (Γ) causes the tracking of these trajectories.

The previous sections can though be synthezised (table 1) w.r.t. the two characteristics that are of concern: the SI scheme (i.e. linear/nonlinear) and the choice of input signals. This is done for the two known approaches: hydrodynamist and automaticians one. The latest is often based on an adaptive control scheme [3,12,13,14] or a Kalman filter one [1,10,15], both relating a linear model of the submersible. Our approach holds for nonlinear model. Also for automaticians approach, last column of table 1 shows that the choice of PRBS as input appears only in [5]. The "exciting trajectories" based inputs are the alternative that we have choosed. This, of course, is not an exhaustive classification but shows different views of the tackled problem.



CHARACTERISTIC APPROACH	S.I. SCHEME	INPUT SIGNALS
HYDRODYNAMISTS	P.M.M.	SINES
AUTOMATICIANS	LINEAR MODEL - Adaptive control (*): [3, 12, 13, 14] - Kalman filter (*): [1, 10, 15]	EXCITING TRAJECTORIES (*): Our work (x): [2, 11, 16]
	NONLINEAR MODEL (*): Our work	CLASSICAL INPUTS (PRBS,..) (*): [5]

Table 1 -*Classification of some SI applications.*
(*): Marine's application - (x)*:Robotic's application.*

4 The exciting trajectories design

The notion of "excitation" is related to the fact that all dynamics of a plant must be "stimulated". The problem of exciting input signals has been largely evoked ([4],[8]) using PRBS, impulse, .. and advantages of such inputs are well known. However, when the input (torque Γ) is obtained by injecting some voltage through an actuator, the input signal properties will be completely modified and thus a PRBS input choice may be inadequate. Another way to stimulate the process is to choose other kinds of input signals. Such signals (torque Γ) may be obtained in a closed loop scheme by tracking some desired trajectories. The last are designed to be exciting of plant dynamics. A first problem that appears is that all dynamics, including perturbations, are re-injected in the loop (see discussion in [7]). This difficulty is cancelled by considering the torque vector (Γ), really injected to the process. A second problem is the design of the exciting trajectories. These will be obtained on the basis of natural plant dynamics (see below). The latest is largely investigated in robotics applications ([2],[11],[16]). Figure 2 shows an example of such exciting trajectories, naming yaw angle, velocity and acceleration, (resp. ψ, $r = \dot{\psi}$, $\dot{r}$) built on the basis of truncated five-degrees polynomials (figure 2).

Rewriting (2) when taking into account different perturbations gives:

$$\Gamma = W(\nu, \dot{\nu}).\Theta + \tilde{\Gamma} + \zeta \tag{4}$$

with:
$\tilde{\Gamma}$: Unmodeled dynamics (also said *'systematic'* errors)
ζ: Stochastic perturbations such as sensor noises, ..(*'random'* error)

It has been demonstrated [2] that bias that occurs in practical LS estimation can only be induced by these two kind of perturbations. Two major results are then issued:

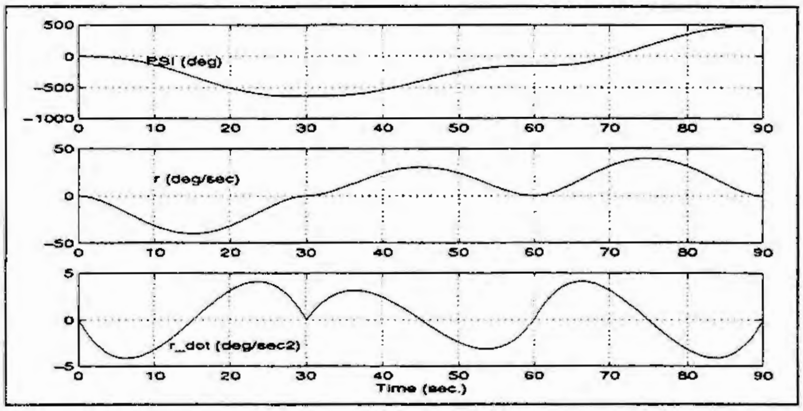


Figure 2: *Exciting trajectories in Yaw example.*
resp.: Angle(deg) – Velocity(deg/sec) – Acceleration(deg/sec^2).

- Let us call ($\mathcal{R}$) the input correlation matrix, defined as follows:

$$\mathcal{R} = \frac{1}{N}\sum_{k=1}^{N}(W^T.W) \tag{5}$$

with (N) the total number of samples along the generated trajectories; The bias sensitivity ($\mathcal{S}_1$) induced exclusively by systematic error is found to be [2]:

$$\mathcal{S}_1 = \frac{1}{\sqrt{\sigma_p(\mathcal{R})}} \tag{6}$$

where ($\sigma_p(\mathcal{R})$) is the smallest singular value of ($\mathcal{R}$). This shows that robustness of identification against perturbations is increased by maximizing ($\sigma_p(\mathcal{R})$), since this amounts to minimizing ($\mathcal{S}_1$).

- Issued from perturbations theory applied to LS algorithm, it has also been proved [17] that the bias sensitivity is closely related to ($\kappa(W)$), the condition number of regression matrix (W):

$$\mathcal{S}_2 = \frac{\left\|\Theta^* - \hat{\Theta}\right\|}{\left\|\hat{\Theta}\right\|} \leq \kappa^2(W).max(\left\|\frac{\Delta\Gamma}{\Gamma}\right\|;\left\|\frac{\Delta W}{W}\right\|) \tag{7}$$

where ($\Delta\Gamma$) and (ΔW) are perturbation quantities (see [17]).

This expression shows that the smaller ($\kappa(W)$) is, the smaller the sensitivity ($\mathcal{S}_2$) is. It can be shown that eq. (6) and (7) are not completely compatible. In fact, easy but tedious simulations yielding both ($\sigma_p(\mathcal{R})$) and ($\kappa(W)$) calculations show an inverse dependency between these two functions. Thus a new sensitivity function ($\mathcal{S}$) is to be defined. A "natural" solution is to choose a function of both ($\sigma_p(\mathcal{R})$) and ($\kappa(W)$). We do so by weighting the two sensitivities ($\mathcal{S}_1, \mathcal{S}_2$), or more precisely ($\mathcal{S}_1, \kappa(W)$):

$$min_{|\nu,\dot{\nu}}\Big\{J = \frac{\alpha_1}{\sigma_p(\mathcal{R}(\nu,\dot{\nu}))} + \alpha_2.\kappa(W(\nu,\dot{\nu}))\Big\} \tag{8}$$

Our aim is to generate some specific trajectories s.t. ($\mathcal{J}$) is minimum. The exciting trajectory problem is thus formulated as an optimization one. (α_1) and (α_2) are weighting coefficients that allow to solve the compromise between ($\mathcal{S}_1$) and ($\kappa(W)$).

5 Simulation Results

A representative model of VORTEX [1] is implemented using MATLAB/SIMULINK. A 6 degrees of freedom (DOF) model allows us to simulate the complete plant behaviour (eq. 1). However, for

[1] VORTEX: Submarine vehicle of IFREMER- Bregaillon, FRANCE

identification purpose, we limit our interest (in this paper) to a **yaw**−motion, ie. rotation around its (z)−vertical symmetry axis. This reduction to 1−DOF is justified since the different DOF of the VORTEX are naturally decoupled. So the identification model is represented as follows:

$$\Gamma_{(1,1)} = \Theta_1.\ddot{\psi} + \Theta_2.\dot{\psi} = W(\dot{\psi},\ddot{\psi})_{(1,2)}.\Theta_{(2,1)} \tag{9}$$

yielding two unknown parameters (Θ_1,Θ_2). Here, ($\dot{\psi}$,$\ddot{\psi}$) are the yaw velocity and acceleration. So one can generate three different exciting trajectories on the basis of the three previous criterions ((6),(7),(8)). These are used successively in the 6−DOF simulator, yielding in the three cases a pair of input/output datas. Eq. (3) is then used to estimate the values of the model parameters (Θ_1,Θ_2) for each case. A second step simulation is done to test procedure robustness by modification of up to 100 percent of the 6−DOF model parameters, except (Θ_1,Θ_2).

Application

The procedure described above yields the following results (table 2). This table shows, for different cost functions, the values of condition number ($\kappa(W)$) which contributes in bounding ($\mathcal{S}_2$) one the one hand, and sensitivity ($\mathcal{S}_1$) and its inversed square root ($\sigma_p(\mathcal{R})$) on the other hand. It also shows estimated parameters values in 'ordinary' case ($\widehat{\Theta_1}$,$\widehat{\Theta_2}$), and in robustness test ($\widehat{\Theta_1}(^*)$,$\widehat{\Theta_2}(^*)$); these are to be compared with '*real*' ones:

$$\theta_1^* = 7.07830(Kg.m^2) \quad \theta_2^* = -50.48000(Kg.m)$$

- Weighting criterions (6) and (7) together gives best results than these when taken independantly (Table 2). It is seen that neither the first cost function (from the left), nor the second, yield correct estimate values, even when counting for the estimate variance. In opposite the smallest value of sensitivity function is obtained for the third cost function.

COST FUNCTION:	$(\frac{1}{\sigma_p(R)})$	$(\frac{\sigma_1(W)}{\sigma_p(W)})$	$(\frac{50}{\sigma_p(R)} + \frac{\sigma_1(W)}{\sigma_p(W)})$
$\kappa(W)$	5.0852	1.0000	5.6817
$\sigma_p(R)$	1.6582	0.0589	2.1777
S_1	0.7766	1.2877	0.6776
$\hat{\theta}_1$	7.0803 +/- 3.35E-4	7.0558 +/- 1.77E-2	7.0786 +/- 3.71E-3
$\hat{\theta}_2$	-50.4786 +/- 6.5E-4	-50.4497 +/- 1.78E-2	-50.4794 +/- 6.14E-3
$\hat{\theta}_1(^*)$	7.078256 +/- 2.5E-3	7.080436 +/- 2.33E-3	7.081805 +/- 2.28E-3
$\hat{\theta}_2(^*)$	-50.473622 +/- 5E-3	-50.468313 +/- 7.6E-4	-50.47463 +/- 3.77E-4

(*) : With 100% coefficients change for robustness tests.

Table 2 -*Different criterion results comparison.*

- In robustness test, the modification of coefficient of the 6−DOF model (except (Θ_1,Θ_2)) can be seen as a "lack" in the model. It is again seen that the 3rd choosed criterion, unless the two others, yields true values of estimated parameters.

6 Conclusion

This paper focus on three important points:

- As it appears in table 1, the approach presented in this paper is fondamentally different from that usually used. In fact, the model to be identified is nonlinear and the inputs that are choosed for this purpose are more elaborated and model based.
- For identification purpose, the exciting trajectories approach is presented as an alternative to the classical input signals (PRBS, Impulse, ..) because of actuators constraints (cf. §4). This is not a new technique, but the innovation lies in its application to a submarine vehicle.
- An encouraging improvement is obtained w.r.t. trajectory generating criterion. This new criterion is obtained on the basis of perturbations type and the analysis of dependancy between two important matrices (regression matrix (W) and input correlation matrix ($\mathcal{R}$)).

Finally, notice that experimental tests are now implemented on the VORTEX submersible to confirm the previous results.

References

[1] **Abkowitz M.** *"Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification,"* SNAME Trans,Vol 88, pp 283-318, 1980.

[2] **Armstrong B.** *"On finding exciting trajectories for identification experiments involving system with nonlinear dynamics,"* Int. J. of Robotics Research, Vol. 8, No 6, pp 28-48, December 1989.

[3] **Åström, K.J.** *"Why use adaptive techniques for steering large tankers ?,"* Int. J. of Control, Vol. 32, No 4, pp 689-708, 1980.

[4] **Åström, K.J. and Eykhoff, P.** *"System Identification - A Survey,"* Automatica, Vol. 7, pp 123-162, 1971.

[5] **Åström, K.J. and Källström, C.J.** *"Identification of ship steering dynamics,"* Automatica, Vol. 12, pp 9-22, 1976.

[6] **Dand, I.W.** *"Some aspects of the hydrodynamics of remotely operated submersibles,"* Proc. IEE Control'85, Cambridge-UK, Vol. 1, pp 156-161, 1985.

[7] **Eykhoff, P.** *"System Identification ,"* John Wiley and Sons Ltd, 1974.

[8] **Fossen, T.I.** *"Guidance and Control of ocean vehicles,"* John Wiley and Sons Ltd, 1994.

[9] **Fossen, T.I. and Fjellstad, O.E.** *"Nonlinear modelling of marine vehicles in 6 degrees of freedom,"* Journal of Mathematical Modelling of Systems, Vol. 1, No 1, 1995.

[10] **Fossen,T.I. and Sagatun,S.I. and Sorensen,A.J.** *"Identification of dynamically positioned ships,"* IFAC/CAMS'95, Trondheim, Norway, 1995.

[11] **Gautier,M.** *"Contribution à la modélisation et à l'identification des robots,"* Thèse de Doctorat d'état, Université de Nantes, 1990.

[12] **Goheen, K.R.** *"The modelling and control of remotely operated underwater vehicles,"* PhD Thesis, University of London, 1986.

[13] **Goheen, K.R. and Jefferys, E.R.** *"The application of alternative modelling techniques to ROV dynamics,"* Proc IEEE ICRA, Cincinnati, Vol 2, pp 1302-1309, 1990.

[14] **Goheen, K.R.** *"Modelling methods for underwater robotic vehicle dynamics,"* Journal of Robotics Systems, 8(3), pp 295-317, 1991.

[15] **Maeda, H. and Tatsuta, S.** *"Prediction method for hydrodynamic stability derivatives of an autonomous non-tethered submerged vehicle,"* 8th Int. Conf. on Offshore Mechanics and Arctic Engineering, The Hague, (3), pp 19-23, 1989.

[16] **Pressé,C.** *"Identification des paramètres dynamiques des robots,"* Thèse de Doctorat d'état, Université de Nantes, 1994.

[17] **Golub, G.H. and Van Loan, C.F.** *"Matrix computation,"* J. Hopkins Univ. Press, 1983.

PROBABILITY MODEL IDENTIFICATION

L. Kuznetsov

Lipetsk State Technical University

30, Moskovskaya St., Lipetsk, 398055, RUSSIA

Abstract. A black box with fuzzy relations between input and output signals is considered. There is a set $\Omega=\{\omega\}$of the measured values $\omega\in\mathbf{R}^n$ of the input w values and a set $\Xi=\{\xi\}$ of the measured values $\xi\in\mathbf{R}^m$ of the output y values. We have to define $\mathbf{M}_y$ and $\mathbf{M}_w$ probability models on these sets under certain conditions so that the joint amount of information related to these probability models is maximum [2].

Introduction.

An appropriate example of such a task may be the definition of the technology of some material or product manufacturing in food, chemical and steel-making industries [1]. In such task product properties are preset. If we denote qualities as $\mathbf{y}=(y_1, y_2, \dots, y_m)$, their minimum and maximum permissible values as $\mathbf{y}'$, $\mathbf{y}''$, and the technological factor as $\mathbf{w}=(w_1, w_2, \dots, w_n)$, then the practically arising task is to determine the best possible ranges of $\mathbf{w}''-\mathbf{w}'$ for the technological values. Sets of values of technological factors **w** and product quality **y** measured on line are used as the initial information.

For metallurgical technology this task arises when technological control is needed. The raw material qualities stand for uncontrollable inputs. Finished properties of metal are the output values, otherwise the technological factors values are controllable inputs or controls. Usually the problem of technology definition arises which provides the finished metal production with the required properties from raw material with fuzzy characteristics.

Because of the single-valued models absence, incomplete information about processes and measurement errors, finished product property $\mathbf{y}\in\mathbf{R}^m=\mathbf{Y}$ are regulated by the permissible value intervals

$$\mathbf{y}'_k \le \mathbf{y} \le \mathbf{y}''_k, \quad k=1,2,\dots,K, \qquad (1)$$

where $\mathbf{y}'_k$, $\mathbf{y}''_k$ are vectors of minimum and maximum permissible values of the finished characteristics of k type metal.

In these conditions the technology $\mathbf{w}\in\mathbf{R}^n=\mathbf{W}$ must be preset in the form of permissible values intervals of technological factors

$$\mathbf{w}'_k \le \mathbf{w} \le \mathbf{w}''_k, \quad k=1,2,\dots,K, \qquad (2)$$

where permissible values, minimum $\mathbf{w}'_k$ and maximum $\mathbf{w}''_k$ of the technological factors define the technology of k type metal production.

Values $\mathbf{y}'_k$, $\mathbf{y}''_k$ are initially preset and the values $\mathbf{w}'_k$, $\mathbf{w}''_k$ are to be defined according to measurement selection $\{\omega\}$, $\{\sigma\}$ of $\mathbf{w}\in\mathbf{W}$ technological factors and $\mathbf{y}\in\mathbf{Y}$ finished product properties.

Model

$\mathbf{M}_w$ probability model on Ω set of measured ω values of **w** inputs of the black box looks like this:

$$\mathbf{M}_w = \{\Omega, \mathbf{A}, P(A) \mid A \in \mathbf{A}\}, \qquad (3)$$

where $\Omega=\{\omega\}$ is a set of **w** values,

A is the system (algebra) of the Ω set subsets,

P(A) is the probability of the event A.

In the simplest case having some practical value the subsets system can be preset as follows:

$$\mathbf{A} = (A, \ \bar{A}), \ A = \{ \omega \in \Omega \mid \mathbf{w}' \leq \mathbf{w} \ \leq \mathbf{w}'' \}, \ \bar{A} = \Omega \setminus A, \qquad (4)$$

where $\mathbf{w}'$, $\mathbf{w}''$ are fixed $\mathbf{w}$ values.

Since Ω is given, the $\mathbf{w}'$, $\mathbf{w}''$ values setting fully determines algebra **A** and $\mathbf{M_w}$ model. The probability model $\mathbf{M_y}$ on the set $\Xi = \{ \xi \}$ of the measured values of the output y values is similar

$$\mathbf{M_y} = \{ \Xi, \ \mathbf{B}, \ P(B) \ | B \ \in \mathbf{B} \}, \qquad (5)$$

where Ξ is a preset set of **y** values,

B is the algebra of the set Ξ subsets,

P (B) is the probability of the event B.

The subsets system **B** can be given like (4)

$$\mathbf{B} = (B , \ \bar{B}), \ B = \{ \xi \in \Xi \mid \mathbf{y}' \ \leq \ \xi \ \leq \ \mathbf{y}'' \}, \ \ \bar{B} \ = \ \Xi \setminus B, \qquad (6)$$

where $\mathbf{y}'$, $\mathbf{y}''$ are fixed values.

Here, as in model (3), Ξ is preset and, consequently, $\mathbf{y}', \mathbf{y}''$ representation fully determines $\mathbf{M_y}$ model. Thus, if we vary algebra's **A** and **B** we may change $\mathbf{M_w}$, $\mathbf{M_y}$ models. Informational correspondence of $\mathbf{M_w}$, $\mathbf{M_y}$ models may be estimated [4] through the amount of joint information which is the function of **A** , **B** algebra's:

$$I_{wy} (\mathbf{A} , \mathbf{B}) = H_w (\mathbf{A}) \ + \ H_y (\mathbf{B}) \ - \ H_{wy} (\mathbf{A} , \mathbf{B}), \qquad (7)$$

where H_w (**A**), H_y (**B**), H_{wy} (**A, B**) are the corresponding entropies.

In practical tasks, one of the algebra's, **B** for instance, is defined by $\mathbf{y}'$, $\mathbf{y}''$ values, and the aim is to estimate $\mathbf{w}'$, $\mathbf{w}''$ values for (6), providing the maximum joint communication value (7), which , in this case, will have the form :

$$I_{wy} (\mathbf{A} \ ) = H_w \ (\mathbf{A} \ ) \ \ + \ \ H_y \ \ - \ \ H_{wy} (\mathbf{A} \ ) . \qquad (8)$$

The model defining algorithm.

Let $\mathbf{M_y}$ model be fully defined. It is necessary to complete $\mathbf{M_w}$ model definition in terms algebra **A** and the definition of $P(A)$, $A \in \mathbf{A}$ possibilities related to it. From (4) we see that the values $\mathbf{w}' = (w'_1, w'_2, \ldots , w'_n)$, $\mathbf{w}'' = (w''_1, w''_2, \ldots , w''_n)$ are determined in $\mathbf{R}^n$ hyperplane and they define the **n** - demensional parallelepiped A. The variation of w'_i , w''_i , $i = 1, 2, \ldots , n$ values leads to some change in configuration and position of parallelepiped A in $\mathbf{R}^n$. With the set $\{ \xi \}$ of **w** values being final and preset, we have to define $\xi_{i\,min}$, $\xi_{i\,max}$, $i = 1, 2, \ldots , n$ and may normalize ξ_i value to the segment [0, 1]. As a result, $\mathbf{R}^n$ space will represent a single **n** - dimensional cube. Any of its nonempty subspaces may be accepted as a zero approximation of A.

In $\mathbf{R}^n$ space we can introduce a notion of the relative density of information as the relation of the amount of communication information and the respective subspace volume. Changes of subspace A volume, changes of $I_{wy}(\mathbf{A})$ information amount, and shifts of hyperplanes correspond to the change of w'_i , w''_i values to $\pm\Delta w'_i$, $\pm\Delta w''_i$, $i = 1, 2, \ldots , n$ values. Using relative changes of information in different directions of $\pm\Delta w'_i$, $\pm\Delta w''_i$, $i = 1, 2, \ldots , n$ we can build the procedure [3] of successive approximation of **A** to the optimum $\mathbf{A}^*$ value. As a result, we have the determined algebra $\mathbf{A}^* = (A^*, \ \bar{A}^*)$ and the probability model $\mathbf{M_w} = \{ \Omega, \mathbf{A}^*, \mathbf{P}^* \}$ in which

$$I_{wy} \ (\mathbf{A}^*) \ = \ \mathit{max} \ I_{wy} (\ A \ \ ) \ , \ A \subset \Omega \ . \qquad (9)$$

NEW TECHNOLOGIES DESIGN OF EXPERIMENT

Anatoly A. Naumov and Andrew V. Zemnitsky
Novosibirsk State Technical University
Krasnij pr.,71 ,kv.7 , Novosibirsk-104, 630104, Russia

Abstract. The problems of software elaboration, concerning experimenting strategies synthesis and data processing in a real-time systems are considered. The elaborated method is a variation of self organization method, combining with e-invariant imbedding method. It is also considered one of possible approaches of regularization of incorrectly - raised active identification problem. This one is based on a self-organized algorithm. It means cyclical mathematical model synthesis experimenting plan sequential generation and evolutionary selection of optimal plan's points.

Introduction

For more then 30 years from works by J.Kiefer [5],[6] , methods of an active identification of systems were working out on a basis of so-called experimenting strategies optimality and equivalence.
Let us remind of the essence of these fundamental statements. Let the regression model be so:

$$y_i=f(x_i,a)+\varepsilon_i=\underset{\sim}{a}^T\underset{\sim}{\varphi}(x_i)+\varepsilon_i=\underset{\sim}{a}^T(\underset{\sim}{\varphi}_1(x_i),\varphi_2(x_i),\dots,\varphi_n(x_i))^T+\varepsilon_i\ ,$$

$$i=1,2,\dots,n;\ x=(x_1,x_2,\dots,x_k),x_i\in X\in R_k\ ,$$

here a - vector of an unknown model parameters, $\varphi(x)$- basis vector of the model, ε_i- observation accidental error realization. In case of least-square method as a estimating strategy of unknown model parameters , a problem of construction, for example , of D-optimal plan of experiments :

$$\xi=(x_1,x_2,\dots,x_n;\xi_1,\xi_2,\dots,\xi_n)\ ,x_i\in X,i=1,2,\dots,n,$$

$$\xi_i\geq 0\ ,\ \sum_i \xi_i=1.$$

takes the following form:

$$\xi^* = \mathrm{Arg}\ (\inf_{\xi} \mathrm{Det}(D(a,\xi))\ .$$

Theorem of optimality and equivalence for D-optimal strategy of experiment is
THEOREM 1 [6].

D-optimal plan of experiment: ξ^*

1) maximized $\det(M(a^*,\xi))$ (minimized $\det(D(a^*,\xi))$);

2) minimized $\max_x d(x,\xi)$;

3) for this plan $\max_x d(x,\xi)=n$;

statements 1)-3) being equivalent. Here and above $M(a^*,\xi)$ - information matrix; $d(x,\xi)$- so-called efficiency function of experiment; n - number of model parameters.

The investigations , carried by us , had testified that system active identification problem in general and experimenting strategy synthesis problem in particular is incorrectly formulated ones , and the using of Kiefer-Wolfowitz-Vedorov statements analogous to pointed above in practice leads to falling down of efficiency of experimental plans. It may be said ,that the active identification problem in a classic form is a twice incorrectly raised one. If the first level of incorrectness is associated with unsteadiness of model parameters evaluation problem, based on least-square method , then the second one is a consequence of basis spatial vector of mathematical model being a argument of experimenting strategy optimality criteria functional.

Proof of incorrectness

Generally , if plan of experiment is a probabilistic measure $\zeta(x)$, then on a space **Z** of experimenting plans may be introduced metrics:

$$\| \zeta_1(x) - \zeta_2(x) \|_Z = \| \zeta_1(x) - \zeta_2(x) \|_F ,$$

where $\|\zeta_1(x)- \zeta_2(x) \|_F$ - is one of the metrics on a functional space **F** .
For instance , $\| \zeta_1(x)- \zeta_2(x) \|_{Lp}$, , $\| \zeta_1(x)- \zeta_2(x) \|_{L2}$, $\| \zeta_1(x)- \zeta_2(x) \|_C$ etc.

By analogy , on a model's space (e.g. regression models) **M**

$$y=f(x,a)=a^T\varphi(x) \text{ (see above)} , \varphi(x)\in\Phi^n ,$$

(without depreciation of the common character , linear by parameters) one may introduce his own metrics :

$$\| f_1(x,a_1)-f_2(x,a_2) \|_M = \| f_1(x,a_1)-f_2(x,a_2) \|_F ,$$

generally different from metrics on **Z** space.
LEMMA .

Let us assume that $\| \zeta_1(x)- \zeta_2(x) \|_Z = \| \zeta_1(x)- \zeta_2(x) \|_{L2}$ and

$$\|f_1(x,a_1)-f_2(x,a_2) \|_M = \| f_1(x,a_1)-f_2(x,a_2) \|_{L2} ,$$

then for any small $\varepsilon > 0$, for which

$$\| f_1(x,a_1)-f_2(x,a_2) \|_M \le \varepsilon$$

it exists $\delta > 0$ that

$$\| \zeta_1{}^*(x) - \zeta_2{}^*(x) \|_z \ge \delta ,$$

where $\zeta_1{}^*(x)$ and $\zeta_2{}^*(x)$ - are Kiefer's optimal plans of experiment that were built in accordance with optimality criteria from the dispersion matrix of model parameters estimations . They are correspond to the models $f_1(x,a_1)$ and $f_2(x,a_2)$ accordingly.

Proof of this lemma is simple enough and as a consequence we may formally affirm that it's correctly :

$$\text{If } \| f_1(x,a_1)-f_2(x,a_2) \|_M \to 0 \text{ then not } \| \zeta_1{}^*(x)- \zeta_2{}^*(x) \|_z \to 0 .$$

Thus , even in a case it was used Kiefer's experimenting plan sequential generation scheme and it take place coincidence to the " precise " model f(x,a), i.e. $f_i(x,a_i) \to f(x,a)$, i=0,1,2,...,
we may assert that $\zeta_i{}^*(x)$ don't coincide to optimal plan of experiment for the model f(x,a).
In that way it was broken one of Hadamard's[4] correctness conditions - continuous dependence of $\zeta^*(x)$ from f(x,a) . This condition is the so-called " condition of steadiness" .

Hence it appears that the experimenting strategy synthesis problem is incorrectly - raised on a pair of topological spaces **Z** and **M** . This note one more time approve the fact that model synthesis problem for various real phenomena in a various applications (geophysics , spectrometry , econometrics, etc.) appears to be incorrect (not as we would like it to be) and requires regularization schemes for it's decision.

Theoretically , incorrectly - raised experimenting strategy synthesis problem by Hadamard may be converted to correctly-raised by Tihonov [11](or conditionally - correct) if , for example , one determine "correctness" space $\Phi_k \supseteq \Phi^n$, and it's exactly known that $\varphi(x) \in \Phi_k$.

As a rule it comes to selection of deliberately more " wide " model's space. Often it means caring out additional experiments , that secure model , has been built on a basis of experimental data , against inadequatness.

Note , that choosing of " correctness " space Φ_K is rather difficult problem requiring some intuition and thorough comprehension of physical sense of a model synthesis problem as a whole .

In addition we may say that the source of incorrectness of experimenting plan synthesis may appear to be any á priori information (distribution of observation errors, optimality criteria of experiment plan , efficiency function of experiment , etc.).

Former approaches to regularization

For the first time the questions of regularization of system active identification problems , taking into consideration the arising both accidental and systematical errors , had been considered by G.E.P. Box and N.R. Draper [1] , but it may be truly said that it had not used mere term "regularization" by them and it seems probable they had not realized them concerning with this phenomenon. Afterwards S.M.Yermakov and E.V.Sedunov [12] had reinforced the results of Box-Draper.

Regrettably , the Box-Draper's approach doesn't solve all the matters of regularization of incorrectly - raised active identification problems , and in some cases does reinforce this incorrectness.

This approach's essence is to minimize mean-squares error in the following form:

$$J=\int_X E\,[\,g(x,b)-f(x,a^*)]^2\,dx\,,\quad g(x,b)=b^T\psi(x),$$

where $\psi(x)$ - basis vector of a real model; b - parameters vector of this model; E -mathematical expectation operator. In so far as this error may be presented as sum of two errors (accidental and systematical ones)

$$J=V+B=\int_X E\,[f(x,a^*)-E\,f(x,a^*)]^2dx+\int_X[\,E\,f(x,a^*)-g(x,b)]^2dx\,,$$

then the experimenting strategy can be found , for example , by means of minimizing , in first , of systematical error (invariant one about unestimated parameters of a real model) , and ,in second , the problem of minimizing of accidental error can be solved on a set of experimenting strategies , minimizing systematical error.

In first sight it seems the Box-Draper's approach ,compared to traditional Kiefer-Wolfowitz one , introduces in a general functional , which had been put in a basis of experimenting strategy synthesis procedure ,the regularizing addition in the form of systematical error , and thereby the active experimenting problem is drawing from the region of incorrectly - raised ones. However this conclusion is false , because as in traditional approach the basis vector of estimated model (besides the real model basis vector) is inserted in the functional that should be optimizing in a course of second stage of Box-Draper's method , and thereby it doesn't give answers on the questions of incorrectness of experimenting strategies synthesis problems in whole.

In general the following statement is just in subject to the problem considered.

THEOREM 2.

D - optimal efficient experimenting plan can't be invariant one relatively some unknown parameters of a model.

The latter means that the Box-Draper-Ermakov-Sedunov's approach to solving of experimenting strategy synthesis problem is an incorrect one in so far as so-called unbiased planning according to authors , pointed above , is M-invariant in fact ,i.e. invariant in relation to unestimated regression parameters of real model ("interfering" parameters) , and invariance in relation to estimated ones is implied in consequence of which this approach to experimenting strategy synthesis problem is precisely incorrectly - raised , and thereby requires to make more precise definitions and revisions.

Obviously , this statement is followed by the conclusions that the usage of the working-up algorithmical and programming wares based on Kiefer-Wolfowitz-Fedorov's theorem [2],[3], must be fulfilled with some degree of caution , and in some cases to decline the usage of it at all.

New approach to regularization

Let us consider one of possible approaches to solving the incorrectly - raised active identification problem. With this aim let us include into the consideration a model of the following form [10],[7] :

$$f(x,a)=h_1(x)a_1+\ldots+h_n(x)a_n\,,$$

where $$x=z_1h_1(x)+\ldots+z_nh_n(x)\,,\sum_i h_i(x)=1\,,h_i(x)\geq 0\,,i=1,2,\ldots,n,$$

z_i , $i=1,2,...,n$, - simplex elements tops S_j, $j=1,2,...,p$; and $V_j\ S_j = X \in R_k$; the rest symbols have been introduced above.

It is assumed, that $S_j \cap S_i = 0$ (for $i \neq j$), and on the set $\{S_j\}$ there are no isolated simplex groups. In fact , just now introduced model corresponds to regression model on space with a simplex structure X Making an assumption of coinciding of the plan of experiment's points of spectrum and tops of simplex elements , and of a linear form of a local model over every element , we can conclude that this regression model is continuous one , but is not continuously differentiated one. Loosing the smoothness of differentiation property on borders of simplex elements looks not the principal thing , permitting to simplify the procedure of model correcting essentially on set of new data , and its following usage.

In this case the procedure of experiments planning is successive one. Every new point of plants spectrum is placed in mean point of any simplex element , the experiment being realized in this point , and them as a result of comparing the outcoming factors value, measured experimentally , with the modeling value, decision to continue the choosing the experimental points within a given simplex element or not is adopted.

Efficiency of proposed scheme with comparing to analogous one [8] is displayed both on the stage of synthesis and of correcting of mathematical model.

Usage of a model estimator for determination of response value at some point is illustrated by fig.1.

$$\begin{tikzcd}
 & & Z=\{z_i\}\ ,i=1,2,\dots,n \arrow[d] \\
x_0 \arrow[r] & S_{j0},x_0 \in S_{j0} \arrow[r] & \{\delta_i(x_0)\}\ i=1,2,\dots,k+1 \arrow[dl] \\
 & f_{j0}(x_0,a^*_{j0})=\sum\limits_i \delta_i(x_0)y_{ij0} &
\end{tikzcd}$$

Fig.1. Calculation scheme for response function at some point x_0.

It will be noted , the mathematical model usage variation, that has been proposed by us is assumed that the equality $a^*_{ij0}=y_{ij0}$, $x_i=z_i$, $i=1,2,...,n$, is valid .

Belonging of some point X_m to any simplex element may be determined by the solving of the system of equations of the form:

$$\begin{bmatrix} Z_{sj} \\ \hline E_{k+1} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \dots \\ \delta_k \end{bmatrix} = \begin{bmatrix} x_{m1} \\ \dots \\ x_{mk} \end{bmatrix}$$

relatively to vector $\delta^T=(\delta_1,\delta_2,...,\delta_k)$. Here $E_{k+1}=(1,1,...,1)$ $x_m{}^T=(x_{m1},...,x_{mk})$, Z_{Sj} matrix with dimension k x (k+1), created by vector-columns, consisting of simplex top coordinates. In case of resolutions of this system are non-negative quantities, a point x_m belongs to S_j.

For this matter , the mathematical model synthesis and experimenting plan sequential generation scheme just considered is a self-organized algorithm in its essence and thus is one of regularization versions of active identification problem [9].

The active identification self-organized algorithmical scheme is represented on fig. 2.

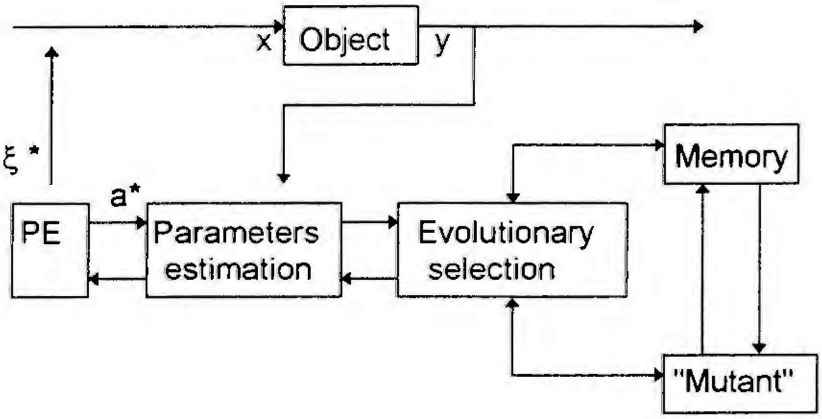


Fig.2. Self-organized algorithm of active identification.

Let us explain what prescription every block of this scheme has. The "PE" block works for generation of the points of plan experiment's spectrum; the problem of reduction of obtained from the studying object experimental data problem of model's estimation constructing (it is possible,by estimating of its parameters) are decided in the block "Parameters estimation". The "Evolutionary selection" block realized one of the self-organization schemes, and as a matter of fact it is a manager block in this structure.

Better steps , arising during the solving of the plan experiment's synthesis problem among all possible ones , that are generated by block-scheme "Mutant" ,are memorized in the block "Memory".

Self-organized algorithm of an active identification ,combined with a method of e-invariant imbedding into so-called "fast space" , it looks of particular efficiency. The essence of this approach is illustrated on fig.3.

$$\begin{tikzcd}
X \arrow[r] & \Omega_x \arrow[r, "f(x,a)"] \arrow[d, leftrightarrow, "V_x"', "V_x^{-1}"] & \Omega_y \arrow[d, leftrightarrow, "V_y"', "V_y^{-1}"] & Y \arrow[l] \\
X_s \arrow[r] & \Omega_{xs} \arrow[r, "f_s(x,b)"] \arrow[d, leftrightarrow] & \Omega_{ys} \arrow[d, leftrightarrow] & Y_s \arrow[l] \\
 & \zeta^* \arrow[r, leftrightarrow] \arrow[d, shift right] & T \arrow[d] & \\
 & Z,\Psi \arrow[u, shift right] & b^* \arrow[l] &
\end{tikzcd}$$

Fig.3. Scheme of e-invariant imbedding method.

The following symbols are used in this scheme:

X,X_s,Y,Y_s - the spaces of input and output system factors;

$\Omega_x,\Omega_{xs},\Omega_y,\Omega_{ys}$ - the structures in corresponding spaces;

$V_x,V_y,V_x^{-1},V_y^{-1}$ - immersion operators and reversal ones for it;

T- model estimating strategy;

Z- the set of admissible experimenting strategies.

Summary

Using nonparametric regression models leads to making experimental design schemes more robust (steady) concerning faults in spatial vector of a model .

Procedure of evolutionary selection of the experimenting plan's points provide practically unrestricted functioning of given model in situation of non-eliminating uncertainty in information.

Imitation of evolutionary selection in a course of decision the problem of prediction and identification proved it's high flexibility for adaptation to the factual processes .

References

1. Box,G.E.P. and Draper,N.R., The choice of a second order rotatable design, Biometrika,50(1965),335-352.
2. Denisov ,V.I. ,Mathematical software of computer system - experimentalist., Nauka ,M., 1977(in Russian).

3. Fedorov ,V.V.and Malyotov,M.B., Optimal designs in regression problems, Math. OF Statist.,3(1972),281-308.

4. Hadamard,J., Sur les problemes aux derivees partielles et leur signification physique , Bull. Univ. Princeton ,1902.

5. Kiefer,J., Optimal experimental design, Jorn.Roy.Statist.Soc.,Ser B ,21(1959),272-319.

6. Kiefer,J.,General equivalence theory for optimum designs (approximate theory). Ann. Statist.,2(1974),849-879.

7. Naumov, A.A., E-invariant imbedding method in the active identification problem . In : Identification , measuring of characteristics and imitation of accidental signals, Novosibirsk, 1991,58-59 (in Russian).

8. Naumov ,A.A., Active express - identification of complicated non stationary dynamic real- time systems . - In : Proceedings of International Symposium "Engineering Ecology-91", Zvenigorod, 1991,283-287 (in Russian).

9. Naumov, A.A., Evolutionary approach to the experimenting strategies synthesis problem and data processing on VLSI. In: Outlooks of the development of computer engineering systems, Riga: RPI, 1989,72 (in Russian).

10. Nudelman ,G.A., Consecutive regression estimation in a system of uninterrupted bit-linear functions. In : Measuring of the characteristics of accidental signals with application of micromachines , Novosibirsk , NSTU, 1988,68-69 (in Russian).

11. Tihonov ,A.N. , About regularization of incorrectly - raised problem.-DAN USSR,Moskow,1963(in Russian).

12. Yermakov, S.M. and Sedunov ,E.V., About undisplaced plans of regression experiments in finite-dimensional functional spaces, Vestnic LSU,1(1974), 12-20 (in Russian).

SIMULATION OF MODULAR DYNAMIC SYSTEMS

A. Rükgauer and W. Schiehlen
Institute B of Mechanics, University of Stuttgart
Pfaffenwaldring 9, D-70550 Stuttgart
email: ru@mechb.uni-stuttgart.de

Abstract. A modular structure for dynamic systems is obtained by functional decomposition. This leads to an interconnected set of dynamic modules. A classification for modular dynamic systems is introduced, including constrained multibody systems with explicit joints, and serveral state space forms. The resulting mechanical differential–algebraic equations are treated by a post–stabilizing projection method. For most efficient simulation, a multi–rate–method is incorporated. The problem of algebraic loops is discussed and a method for its solution is proposed. The simulation program *NEWMOS* supplies all simulation strategies introduced. Several examples from different areas of dynamics show the power of the approach proposed

Introduction

Analyzing large dynamic systems as for instance in mechatronics, one usually has to deal with systems including very different time scales and a tremendous potential of complexity. A modular block simulation approach supported by computer programs as *SIMULINK* [6] among others is well suited to treat standard problems following from the mechatronic system design. However, when the problem size increases, and in particular, when mechanical systems are considered, general purpose block simulators are no longer adequate. An extended strategy supporting modular structures not only for modelling but also for the numerical treatment is preferable. It is the aim of this paper to introduce a simulation approach allowing to exploit the system structure supplied by modular modelling for the needs of efficient simulation by time integration.

Modular structures are inherent to any technical problem. A partitioning of the overall problem by engineering intuition leads to a set of interconnected modules. In Fig. 1 the functional decomposition of a platoon of articulated vehicles, Petersen et al. [8], is shown where two vehicles are linked by a rigid tow bar: Modules for the dynamics of the lateral motion of each of the two vehicles, steering, actuators, and control are derived. The modules are interconnected by both constraint conditions and by signal flow from one module to the other.

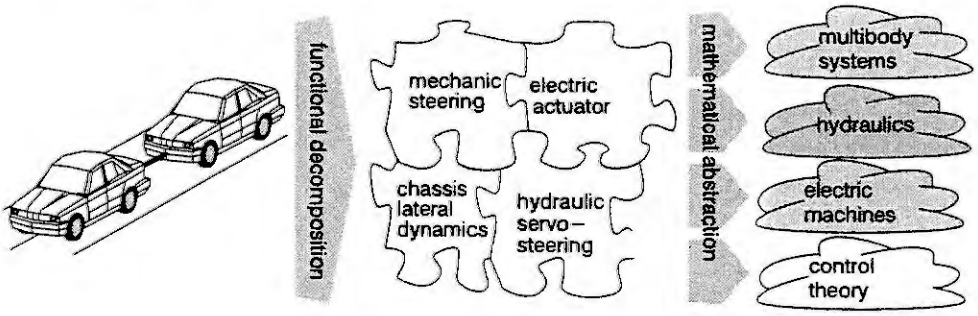


Figure 1: Functional decomposition of a dynamic system

Types of Dynamic Systems and Encapsulation

To describe the system behavior, the transfer function approach widely used in control theory is appropriate: the system is described by an output vector function g as

$$y = g(u, p, t) \tag{1}$$

with the n_u–input–vector u, the n_p–parameter–vector p, the time t and the n_y–output–vector y. The modular system structure may contain q modules. Then, all the inputs and outputs of the system are related to each other by the constant $n_y \times n_u$–incidence–matrix $\boldsymbol{P}$ with $n_u = \sum_{i=1}^{q} n_{u_i}$ and $n_y = \sum_{i=1}^{q} n_{y_i}$ as

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{u}. \tag{2}$$

The dynamics of systems is presumed to be an internal quantity and can be described in various manners. The most general dynamic description used is the explicit, nonsingular and nonlinear state space form which reads as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{3}$$

where the n_x–state–vector $\boldsymbol{x}$ is used. A variation of Eq. (3) is the linear state space form as commonly used in control engineering:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{4}$$

with the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Experimental data and explicitely discretized dynamic systems can be described using the discrete state space form of Eq. (3) as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{f}(\boldsymbol{x}^{(k)}, \boldsymbol{u}^{(k)}, t^{(k+1)} - t^{(k)}) \tag{5}$$

with time index k.

A very important class of dynamic systems is described by mechanical structures in form of multibody systems (MBS). In the following, the semi–explicit, possibly nonminimal mechanical descriptor–form, Hairer and Wanner [5], is used, which reads

$$\begin{aligned} \dot{\boldsymbol{q}} &= \boldsymbol{K}\boldsymbol{z} + \boldsymbol{b} =: \boldsymbol{a}, && \text{(6a)} \\ \boldsymbol{M}\dot{\boldsymbol{z}} + \boldsymbol{k} &= \boldsymbol{q}_e - (\boldsymbol{G}\boldsymbol{K})^T\boldsymbol{\lambda} \text{ and} && \text{(6b)} \\ \boldsymbol{g} &= \boldsymbol{0} && \text{(6c)} \end{aligned}$$

with the n_q–vector $\boldsymbol{q}$ describing the generalized coordinates and the n_z–vector $\boldsymbol{z}$ describing the generalized velocities. Eq. (6a) is referred to as kinemetics relation, Eq. (6b) describes the dynamics. Additional constraints by closed chains are introduced by Eq. (6c). The set (6) of mechanical differential–algebraic equations (MDAE) is of index $k = 3$, Hairer and Wanner [5]. For the numerical treatment, an index reduction must be performed by differentiating the constraint relation (6c) twice:

$$\begin{aligned} \dot{\boldsymbol{g}} &= \boldsymbol{G}\boldsymbol{K}\boldsymbol{z} + \boldsymbol{G}\boldsymbol{b} + \frac{\partial \boldsymbol{g}}{\partial t} = \boldsymbol{0} \text{ and} && \text{(7a)} \\ \ddot{\boldsymbol{g}} &= \dot{\boldsymbol{G}}\boldsymbol{a} + \boldsymbol{G}\boldsymbol{K}\dot{\boldsymbol{z}} + \boldsymbol{G}\left(\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{q}^T}\boldsymbol{a} + \frac{\partial \boldsymbol{a}}{\partial t}\right) + \frac{\partial^2 \boldsymbol{g}}{\partial \boldsymbol{q}^T \partial t}\boldsymbol{a} + \frac{\partial^2 \boldsymbol{g}}{\partial t^2} = \boldsymbol{0}. && \text{(7b)} \end{aligned}$$

To extend the modular block design to mechanical systems, any mechanical system can contain interaction nodes of the form $\boldsymbol{g}_{node} := [\boldsymbol{r}|\boldsymbol{S}]$ with the node position vector $\boldsymbol{r}(\boldsymbol{q}, t)$ and the node rotation matrix $\boldsymbol{S}(\boldsymbol{q}, t)$, see Fig. 2, and Ref. [10]. Now additional joints can be defined using the node data. The tow bar from

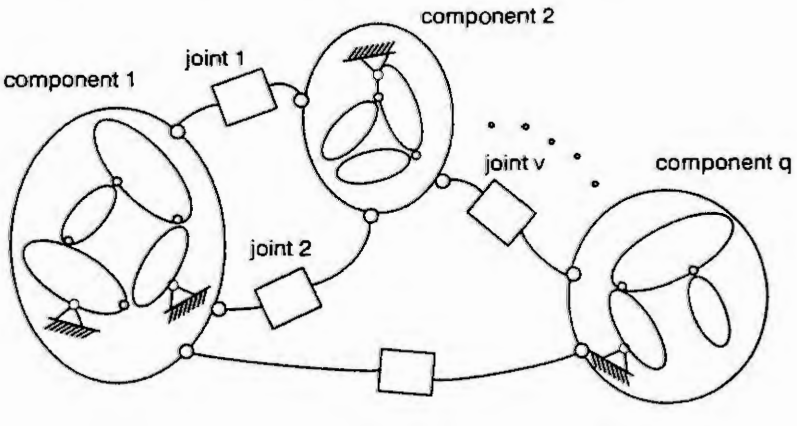


Figure 2: Mechanical systems with joints

the above example of Fig. 1 can be defined using a spherical difference joint (spherical joints with given, fixed distance l) as

$$g_g \;:=\; |\boldsymbol{r}_1 - \boldsymbol{r}_2| - l = 0 \tag{8}$$

using the node vectors $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$. Appropriate index reductions can be found as well, Rükgauer and Schiehlen [9]. The global equations of motion for such modular mechanical systems with q components and v joints show a block-diagonal structure:

$$\underbrace{\begin{bmatrix} \dot{g}_1 \\ \vdots \\ \dot{q}_q \end{bmatrix}}_{\dot{q}} = \underbrace{\begin{bmatrix} \boldsymbol{K}_1 & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{K}_q \end{bmatrix}}_{\boldsymbol{K}} \underbrace{\begin{bmatrix} z_1 \\ \vdots \\ z_q \end{bmatrix}}_{z} + \underbrace{\begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_q \end{bmatrix}}_{\boldsymbol{b}}, \tag{9a}$$

$$g = [g_1, \ldots, g_q | g_{g_1}, \ldots, g_{g_v}]^T, \tag{9b}$$

$$\underbrace{\begin{bmatrix} \boldsymbol{M}_1 & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{M}_q \end{bmatrix}}_{\boldsymbol{M}} \underbrace{\begin{bmatrix} \dot{z}_1 \\ \vdots \\ \dot{z}_q \end{bmatrix}}_{\dot{z}} + \underbrace{\begin{bmatrix} \boldsymbol{k}_1 \\ \vdots \\ \boldsymbol{k}_q \end{bmatrix}}_{\boldsymbol{k}} = \underbrace{\begin{bmatrix} q_{e_1} \\ \vdots \\ q_{e_q} \end{bmatrix}}_{q_e} - (\boldsymbol{G}\boldsymbol{K})^T \boldsymbol{\lambda}. \tag{9c}$$

The structure of the global constraint matrix G is diagonal in the upper part, the lower part is sparsely filled according to the orientation of the mechanical components in the global structure.

For the actual derivation of dynamic models, a formal and abstract interface allowing for the encapsulation of the above dynamic descriptions is used. This interface is implemented in the programming language *C* and is based upon a hierarchical structure, see Fig. 3. Each system is described by the structure `system_descr` which contains the interface functions for initialization (file opening, etc.), termination, output equation, state equation (`eval`), state events (`roots`), the type identifier (`type`) and a corresponding substructure (`type_spec`). In the case of MBS as shown in Fig. 3, the substructure `mbs_descr` includes the n_b node points `bind` of type `mbs_contact_descr`. The `special` parameter may be used to define special properties of the system, e.g. a constant mass matrix $\boldsymbol{M}$. A similar interface definition is introduced for joints and other elements needed for simulation, see below. The data structures shown can easily be generated by code-converters or by hand. Code-converters are currently available from *NEWOPT/AIMS* [2] and from *DSL* [7]. The formal interface definition omitting any information about the actual interconnections between the modules allows for a high degree of code-reusability. All interconnections are introduced at run time.

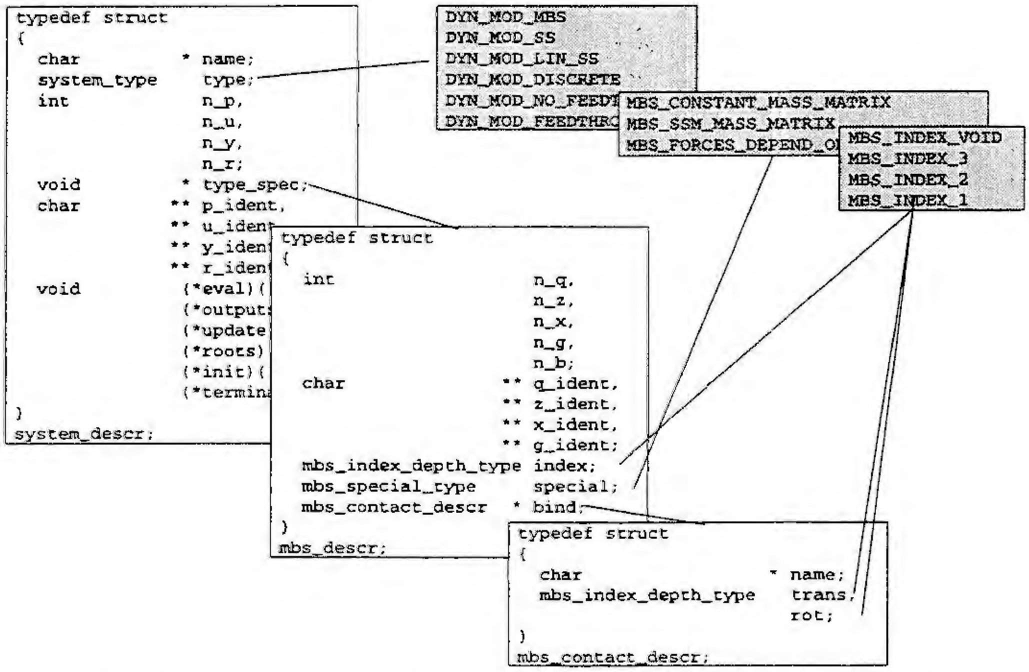
Numerical Methods

For the numerical treatment of the MDAE, a post-stabilizing projection method is used. An extended right hand side by Eqs. (6a), (6b) and (7b) is discretized by any ODE integrator. After every integration step executed, a projection onto the manifold described by the conditions (6c) and (7a) is performed. An energy-optimal projection can easily be found by the defect relations

$$\begin{aligned} \boldsymbol{G}\, d\boldsymbol{q} &= d\boldsymbol{g} \text{ and} && (10a) \\ \boldsymbol{G}\boldsymbol{K} d\boldsymbol{z} &= d\dot{\boldsymbol{g}}. && (10b) \end{aligned}$$

Relation (10b) can be solved directly as Eq. (7a) is linear in z. Although Eq. (6c) can be strongly nonlinear, only a fixed number of 2 iterations must be performed to solve Eq. (10a) due to the fact that the performance of the overall projection resulting from the iteration is directly linked to the index of the system to be solved, Ascher [1]. Employing this fact, a quite efficient integration scheme is obtained, Ref. [9].

To exploit the modular system structure throughout the integration process, a multi-rate strategy, Gear [4], is used. Such an approach is essential for the efficient treatment of large mechatronic systems that include systems at very different time scales. The multi-rate-approach is described in detail in [9], too. As the multi-rate integration introduces additional errors, a multi-rate error control is supplied that adjusts the multi-rate-stepwidth used for inter-system-communication. For better multi-rate-performance, interpolation procedures are used for the reconstruction of the input data.



```
typedef struct
{
  char          * name;
  system_type     type;
  int             n_p,
                  n_u,
                  n_y,
                  n_r;
  void          * type_spec;
  char         ** p_ident,
               ** u_ident
               ** y_ident
               ** r_ident
  void           (*eval)(
                 (*output
                 (*update
                 (*roots)
                 (*init)(
                 (*termin
}
system_descr;
```

```
DYN_MOD_MBS
DYN_MOD_SS
DYN_MOD_LIN_SS
DYN_MOD_DISCRETE
DYN_MOD_NO_FEEDT
DYN_MOD_FEEDTHRO
```

```
MBS_CONSTANT_MASS_MATRIX
MBS_SSM_MASS_MATRIX
MBS_FORCES_DEPEND_O
```

```
MBS_INDEX_VOID
MBS_INDEX_3
MBS_INDEX_2
MBS_INDEX_1
```

```
typedef struct
{
  int                          n_q,
                               n_z,
                               n_x,
                               n_g,
                               n_b;
  char                      ** q_ident,
                            ** z_ident,
                            ** x_ident,
                            ** g_ident;
  mbs_index_depth_type  index;
  mbs_special_type      special;
  mbs_contact_descr   * bind;
}
mbs_descr;
```

```
typedef struct
{
  char                   * name;
  mbs_index_depth_type     trans,
                           rot;
}
mbs_contact_descr;
```

Figure 3: Hierarchical set of data–structures describing dynamic systems

As Eq. (2) describes dependencies between different dynamic system modules, an order of evaluation must be determined before the simulation can be performed. This task is solved using graph methods. A directed graph with nodes representing the dynamic system modules, and edges representing the connections, is designed. Searching for the strickly coherent components, Sedgewick [11], leads to the correct order. However, in case of loops, the search algorithm must be modified in order to mark the loop-closing edges. Then, potential loops and actual loops have to be distinguished. Potential loops are those for which a strickly coherent component is found, while actual loops require in addition that all modules involved show throughputs.

Potential loops are determined and solved by one evaluation of the output equations of all systems contained. Then, the actual loops are known, and the corresponding marked edges are disconnected and replaced by additional inputs for which a constraint condition holds. These additional constraints cannot be solved by DAE–methods as introduced above. A simplified Newton–iteration using Broyden's vector update secant formula, Dennis and Schnabel [3], is adequate to treat this problem. So even complex dynamic systems with feedback and extensive throughputs can be solved efficiently.

Simulation Concepts

The simulation program *NEWMOS* was set up allowing for simulation using the above numerical strategies. Any so called services, referred to in the *NEWMOS*-context for systems, integration codes, joints, and interpolation schemes for multi-rate integration, are loaded at run time as precompiled object code. Also, the interconnections are introduced at run time. To assign the proper simulation topology, an interpreter is supplied allowing the user to introduce the problem in a mnemonic form. In addition, the interpreter allows for the run–time–definition of dynamic systems, so that, in case of rather small problems, no compilation must be performed at all. For the articulated vehicles problem, a code fragment for loading the precompiled vehicle models and the tow bar joint, definition of a steering controller and connection of both vehciles by joint and steering control with second vehicle, and some parameter assignment reads as

```
define ss_system (SteeringPrinciple) with inputs (mu1, mu2) outputs (del, delp)
  parameters (K1, K2) by { y[0] = K1 * mu1 + K2 * mu2; y[1] = 0; };
load system (Vehicle1) of type (spatial_car_model) from file ("car.so");
load system (Vehicle2) of type (spatial_car_model) from file ("car.so");
load joint  (TowBar) of type (spher_diff_joint) from file ("spher_diff_joint.so");
```

```
link(Vehicle1.rear_towbar_node, Vehicle2.front_towbar_node) by (TowBar);
connect(Vehicle2.del, SteeringPrinciple.del);
TowBar.l = 1.8;
```

Several dynamic systems are combined to a so called set which is treated by an integrator as assigned by the user. Several sets are allowed concurrently, resulting in a multi–rate–simulation. Using this approach, one can easily switch between different simulation strategies and dynamic system structures, module tests can be performed independently and finally the whole dynamic problem can be assembled quite easily.

Examples

A 10–body–pendulum is set up using mass points with an interaction node at the center and spherical difference joints. The simulation results for the autonomous pendulum are shown in Fig. 4 (left). To study the multi–rate–simulation strategy, an actor with very high frequency eigendyanmics is added to hieve the first arm from the low equilibrium to the horizontal position, as shown in Fig. 4 (right). In closed form using a stiff integrator, this simulation task for $t_{sim} = 10$s requires 207s. Applying the multi–rate–strategy, the computational cost can be dropped down to 60s.

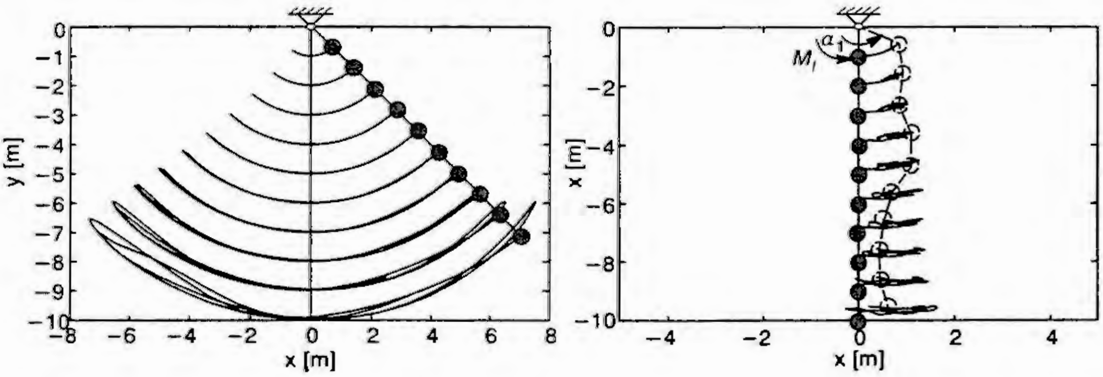


Figure 4: Autonomous (left) and controlled (right) 10–body–pendulum, crossed circles mark initial position, zero initial velocity

The articulated vehicles problem as introduced in Fig. 1 is also treated by a multi–rate–integration. Here, the fast and stiff components as hydraulic servo, electric actuator, and control device are placed in one set, the vehicles are placed in another, as described in detail in Ref. [9]. The results from simulation of an ISO–lane–change at a velocity of $v = 20\frac{\text{m}}{\text{s}}$ are shown in Fig. 5. It turnes out that, for an optimal integration configuration, the results for $t_{sim} = 8$s can be obtained within 80s. Without multi–rate–strategy, a simulation of the given maneouver takes serveral days.

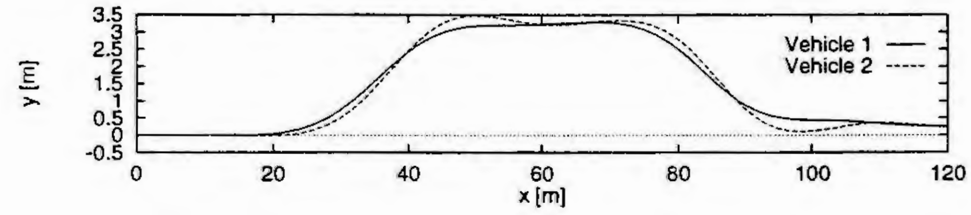


Figure 5: Articulated vehicles undergoing ISO–lane–change

Fig. 6 (left) shows a gearbox with two pulleys connected by a belt. The given model, Fig. 6 (right) includes one potential and one actual loop. As input to the system, step functions acting as torque at the pulley are used. The simulation program detects the loop and supplies the correct solution after one interation due to the fact that all throughputs contained are linear.

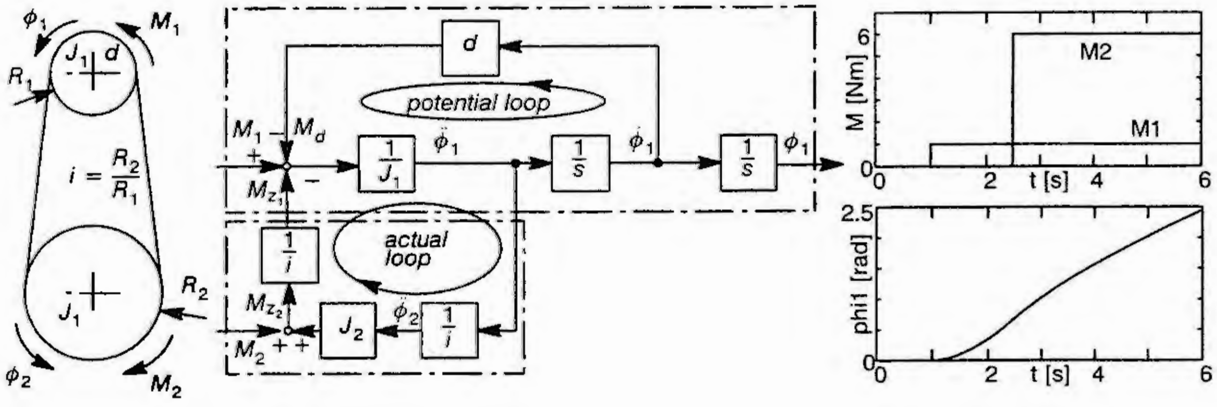


Figure 6: Belt–gear, implementation containing algebraic loop, and simulation results

Summary

A modular modelling and simulation approach has been proposed for the treatment of large dynamic problems as arising in mechatronics. Mechanical systems with additional joints and constraints that allow for a modular description are supported. The the resulting mechanical differential–algebraic equations can efficiently be treated by a post–stabilizing projection method. Most efficiently, the modular dynamic system can be simulated by a multi–rate integration. Due to the modular structure, algebraic loops can occur. A strategy for loop–detection and –treatment is shown. Examples from multibody dynamics, vehcile dynamics, and system dynamics show the validity of the proposed approach. Employing the multi–rate–strategy, big savings in computation time can be obtained.

References

[1] U.M. Ascher: *Stabilization of invariants of discretized differential systems.* Report, Department of Computer Science, University of British Columbia, Vancouver, 1996.

[2] D. Bestle and P. Eberhard: *NEWOPT/AIMS 2.2. Ein Programmpaket zur Analyse und Optimierung von mechanischen Systemen. Anleitung AN–35.* Institut B für Mechanik, Universität Stuttgart, 1994.

[3] J.E. Dennis and R.B. Schnabel: *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall Inc., Englewood Cliffs, 1983.

[4] C.W. Gear: *Multirate methods for ordinary differential equations.* Report UIUCDCS-F-74-880, Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, 1974.

[5] E. Hairer and G. Wanner: *Solving Ordinary Differential Equations II. Stiff and Differential–Algebraic Problems.* Springer Verlag, Berlin, ..., 1991.

[6] The Mathworks Inc., Nattick: *SIMULINK User's Manual*, 1992.

[7] Mechatronik Laboratorium Paderborn (MLaP), Universität-GH Paderborn: *DSL und CAMeL Handbuch*, 1994.

[8] U. Petersen, A. Rükgauer and W. Schiehlen: *Lateral Control of a Convoy Vehicle System.* In *Proceedings of the 14th IAVSD Symposium held at Ann Arbor, Michigan, USA, August, 21-25, 1995*, volume 25 of *THE DYNAMICS OF VEHICLES on roads and tracks*, Lisse, Swets & Zeitlinger, to appear.

[9] A. Rükgauer and W.O. Schiehlen: *Efficient Simulation of the Behaviour of Articulated Vehicles.* In H. Wallentowitz (editor): *Proceedings of the International Symposium on Advanced Vehicle Control*, pages 1057–1070, Aachen, 1996. AVEC'96 - Organizing Committee.

[10] W. Schiehlen: *Technische Dynamik.* Teubner Verlag, Stuttgart, 1986.

[11] R. Sedgewick: *Algorithmen in C.* Addison–Wesley, Bonn, ..., 1992.

DATA MODELS FOR OBJECT ORIENTED MODELING AND SIMULATION

R. Girelli and C. Maffezzoni
Politecnico di Milano
P.zza L. da Vinci 32, 20133 Milano
e-mails: girelli@elet.polimi.it, maffezzo@elet.polimi.it

Abstract. Simulation is becoming crucial in many engineering activities, and models of various complexity are today used for different purposes. As a consequence, there is a strong effort aimed to develop new tools able to simplify and support modeling and simulation.

Object oriented modeling is considered to be the best way to match the natural engineering decomposition of physical systems by components. This approach has been adopted in MOSES, a modeling and simulation environment for complex systems developed in an object oriented database. Models are defined using a fixed data structure in which all the data involved in modeling and simulation are organically related. The principal features of MOSES are described, and the role of the OO database in their implementation is discussed.

1- Introduction

In the last years, production and technological processes are becoming more and more complex. For various reasons (economic, environment preservation, legislation) it is necessary to save energy and materials, or to re-elaborate some products. This means that processes must contain recirculation loops which determinc interactions between many parts of the system and highly complicate the dynamics. Such interactions must be taken into account during the design work.

When designing a complex technological process, what today happens is that control system design "follows" the other design choices. This means that the control system can not be based on real evaluations of the global system, but only on its "presumed" behaviour, which is derived from the behaviour of similar systems. Hence, a closer integration of control engineering with the other engineering disciplines is necessary. In particular, it has been demonstrated that in various applications (see for instance [10]) it is not always possible to separate process design from control system design.

The modern answer to such problems is Computer Aided Control Engineering, (CACE), that [7] "is a specialized form of modeling and simulation with emphasis on design and implementation of feedback control systems". As such, the core of any CACE environment should be modeling and simulation, with particular attention to the integration with the other work process activities.

It is also recognized [3] that the most profound trend in engineering contractor environment is in the way projects are executed and, in particular, the drive to utilize integrated data base technology for the execution of engineering design projects.

MOSES (Modular Object-oriented Software Environment for Simulation) [8] is a modelling and simulation environment developed at Politecnico di Milano with the aim of simplifying and organizing the simulation work process for complex hybrid systems. With reference to models' reusability, the main aspects concern:

1) generality and understandability of the model description,
2) automatic correlation of all the data involved in modeling and simulation activities,
3) easy access to libraries of certified models and library management facilities.

Point 1) is a characteristics of the so called Model Definition Language (MDL), while points 2) and 3) are more related to software and system choices. MOSES adopts a modular and "natural" MDL to describe models, and it is implemented in an object-oriented database, in order to organize and relate all the data involved in modeling and simulation activities. This paper describes the role and relations between MDL and database features in the implementation of MOSES.

2 - Model definition in MOSES

The object oriented approach implements a set of powerful concepts that have been succesfully applied in software engineering, creating the object-oriented programming paradigm. Exploiting the similarities between software and model behaviour, they have been applied also to modeling, creating the so called Object Oriented Modeling (OOM) [11]. MOSES' MDL belongs to the class of object oriented modeling languages, which includes examples like Omola [12], Dymola [6] and NMF [13].

Using OOM to model complex systems, it is possible to use a top-down approach consisting in: decomposition of the global system into components, determination of the interfaces among them and writing of every component model. But more frequently, exploiting the generality of existing models organized in libraries. it is convenient to build the global model with a bottom-up approach. Of course, this requires an easy access to certified model libraries and some efficient managing mechanisms. This aspect is specially considered in MOSES.

In OOM. a high level of modularity is achieved because any part of the reality is described as an "open" model. which interacts with the external world through two kinds of ports: physical terminals. which are associated with power exchange and represent the spatial regions where interactions take place, and control terminals. which are associated with information exchange [14]. The module's internal behaviour is independent of the context where it is used, so a complete encapsulation is obtained. To obtain full modularity. physical terminals in MOSES are strictly standardized: for any physical domain there is only one or a few terminals which correspond to the type(s) of interactions between physical components. For example. in 3-D robotics only one type of mechanical terminal can be used: a connection between mechanical terminals represents a "welding" in the physical system. Control terminals have not any particular structure, because exchange of information is naturally dimensionless; they are causal. so we have input and output ones.

Open models can be aggregated by establishing a one-to-one connection between two identical physical terminals (or control terminals with opposed causality). Abstraction mechanism is implemented by bringing together submodels in an Aggregated Model (AM) that represents all them. Any terminal of a submodel which is not connected to another submodel's terminal is "free" and is reported at the upper level by reproducing it as a free terminal of the AM. This type of connection implements abstraction.

Those models at the lowest level of the aggregation hierarchy are called Simple Models and may be either Continuous-time Simple Models (CSM) or Discrete Simple Models (DSM). A CSM is characterized by a *behaviour* which is described by a set of DAEs, i.e. it is, in general, given a *declarative* form. Some equations (that we call Conditional Equations) can also have different expressions, depending on the value of a set of conditions.

A DSM is used to describe either a discrete-time (synchronous) control system, where the output variables may change only at predetermined, generally equispaced, discrete time instants, or a discrete-event asynchronous system. where the output variables may change only when certain events occur. Since a DSM can exchange only information. it interacts with the external world only through control terminals. Moreover, being a causal system. its behaviour is described by a *procedure* that may consist of equations in explicit form or. more generally. by a software procedure. This is an important choice that allows a real test of the control system. because the procedure simulated in MOSES can be *the same* running on the real controller.

A Plant is defined as a special Aggregate Model that is not submodel of any other. Therefore, a Plant is a "closed" model. i.e. a model which may have only Control Terminals. It is a causal model and is the only kind of model that can be simulated. A Plant is affected by a set of parameters to be defined: this means that such parameters (free parameters) may be assigned/modified afterwards during the analysis process: when the free parameters are specified one produces a *Version* of the plant (see Par. 3).

Once the global model is defined, usually a symbolic manipulation of the system of equations is required. in order to solve some problems due to the adopted general description and improve numerical efficiency [5].

One major difference with other OO languages is that in MOSES it is strictly required that the whole behaviour (equations. variables. terminals) is completely defined in any model, with no permission of using the inheritance mechanism to distribute the behaviour of a model through a class hierarchy. The motivations of this choice are the following ones:

- model libraries are easier to read and more portable, because any model has its complete behaviour "on board";
- generating a hierarchical organisation of model behaviours (in an ordered way) requires that any model library be fully defined at the beginning, so forbidding the natural expansion of libraries with the application experience;
- when using an object-oriented database it is highly convenient to organise modeling data in a fixed structure of classes (i.e. a fixed database schema), so that software behaviour (methods) is directly associated to data. as is usual in object-oriented programming. So. it is easy to match modeling language with data model.

As already pointed out in the Introduction, model reuse can be really effective if models are organized in accessible and certified libraries; furthermore. the simulation environment should have functionalities supporting library management and (possibly) promoting reuse of models. Creating a model of a physical component in MOSES (for example a mechanical body) means creating a Continuous Simple Model and filling the provided "slots" like variables, parameters and equations. Any mechanical body has the same behaviour of this. and the difference between the relative models is only in parameters' values. In MOSES there is a simple way to safely manage this situation: an original model is called *prototype model*, while its copies. in which it is allowed to modify only parameters, are called *instance models*. Since a link is created among them. all the instances can be (automatically) updated when the prototype is changed. Note that the relation between a model and its prototype is similar to an inheritance mechanism, since a model "inherits" the behaviour of the prototype and is specialized by an own value of the parameters. From a software point of view, both prototype and instances are models of the same basic type (for example, Continuous Simple Model): the links among them are easy to manage only because all the environment is implemented in a database.

3- Modeling and simulation data organization

Together with the generality of description, an element that strongly influences the reusability of models is the availability of libraries. For this reason, MOSES appears to the user as a unique library with tree structure, where the leaves are models (either prototypes or components) and structuring elements are called *library categories*. To guarantee data integrity and security while sharing data among different users, some mechanisms typical of multi-user environments have been introduced, like *ownership, locking* and *rights*, which apply to models and to categories as well.

The model database is divided in two parts:

- the public part, which contains general and reliable prototype models of common use;
- the private part, where any user stores his/her own models (either prototypes or instances).

The users are divided in three groups, with different levels of authorization: Programmers, Library Administrators and Analysts. A member of the Programmers' group has the possibility of modifying the database schema adding or changing classes (a situation occurring during the development of the environment or when a major change is needed). The Library Administrators can write in the public part of the model database (i.e., they can administrate the public model library) and can read in the private part. A member of the Analysts' group has the possibility of writing in his models and of reading/writing in other models according to the rights he has.

A very important issue when making simulations in engineering is the automatic correlation and documentation of modeling and simulation data. In MOSES, four structures that allow a quick and efficient access to data have been identified:

- **Model**: it stores the declarative form of a model. If the model is a plant, it can be given also a procedural form. Any procedural form is stored keeping reference to the plant it belongs to.
- **Version**: it refers to a plant model, of which it represents a further specification by assigning the value of free parameters. Versions of the same plant have the same procedural form but different values of some parameters. Versions are very useful in all the cases in which model parameters tuning is needed.
- **Plant state**: it refers to a version, of which it defines the initial values of all the variables. A plant state defines all the data necessary to start a plant simulation.
- **Transient**: it is the result of a simulation, and is constituted by the records of a set of variables, together with that of the exogenous inputs of the model. It refers to the plant state from which the simulation was started.

4- Logical data model and database implementation

In order to organize the simulation activity, the core of the MOSES environment is implemented in a database where all the entities involved in the simulation work process are stored in an ordered way. The use of a database is typically aimed at:

- centralizing the management of data ;
- preserving data consistency, integrity and security while sharing data among different users;
- structuring data and give quick access to them;
- tracing of operations.

Furthermore, the choice of using fixed data structures in a database has forced to do an abstraction effort aimed at understanding how to efficiently represent hybrid models and related simulation data, i.e., which are the fundamental types of models, their properties and similarities.

It can be observed that, since many requirements (like completeness, consistency, non ambiguity) are in common, a database schema design could be a good starting point to define an MDL. Such a schema could also become a standard neutral format for model definition or, at least, it could be easily exchanged among different software programs in a standard format like the emerging STEP/PDES [1], which makes use of EXPRESS to define a data schema for specific application areas.

The conceptual data model has been defined in the form of Entity-Relationships (ER) diagrams; figure 1 sketches the core part of it, i.e., the part related to the definition of the entity "model".

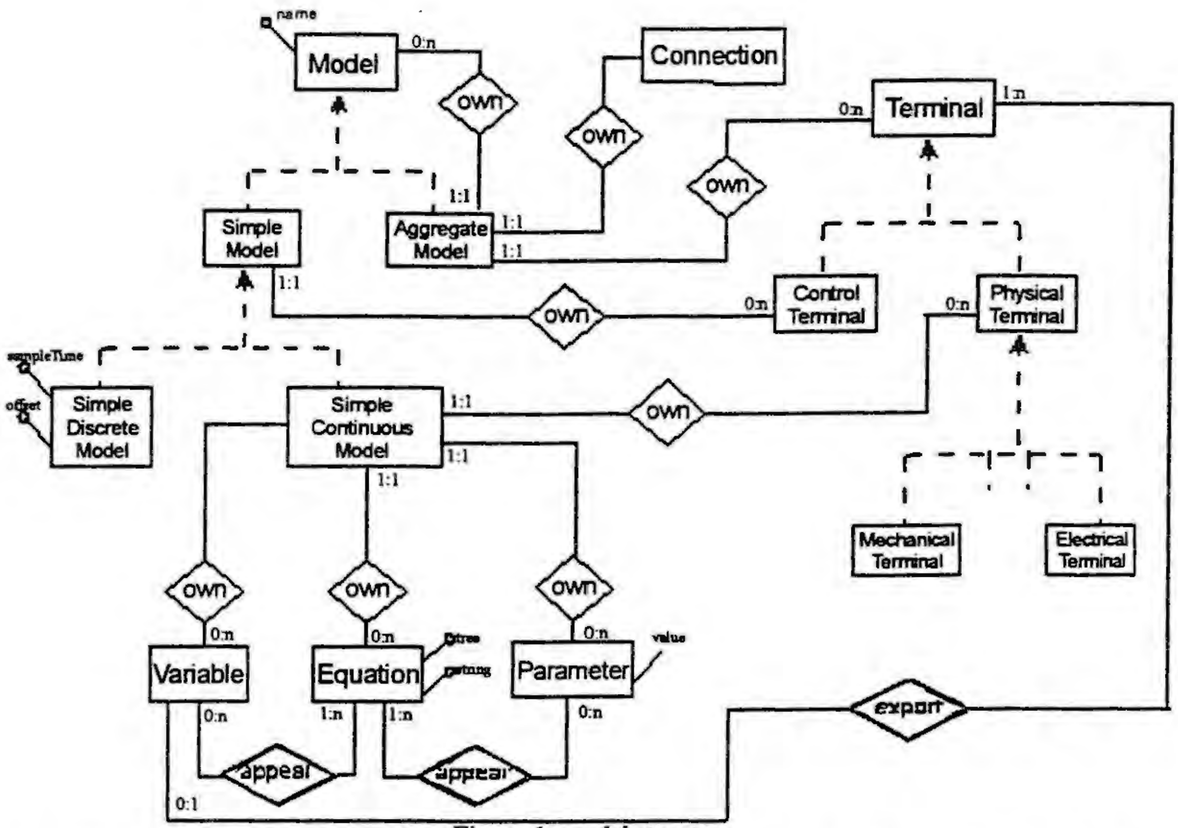


Figure 1: model structure

It is evident from figure 1 that the conceptual data model is very complex and rich of relationships among entities. In order to represent such a schema in a data structure, the object oriented approach has been adopted. The specific reasons for this are the following:

- The OO data model has a very rich semantic, so it can match the application complexity much better than the relational one.
- The OO environment is very flexible and allows fast prototyping, very useful particularly in the first phases of a complex project, and is also much easier to extend and to maintain.
- OO data model associates operations to data structures. In this way, it is possible to specialise not only static properties but also the behaviour of entities. This allows an evolutionary development of the project, in that we can define a kernel of common structures (and relative behaviours) and subsequently specialise them in many application areas, having great benefits in terms of code reuse and maintainability.
- Relationships among data (like existence dependences, corresponding to the "own" relations of figure 1) are needed that are much easier to implement and to manage in an OODB, exploiting the 'part of' relationship peculiar of the OO paradigm; moreover, it is very natural to encapsulate a certain relationship (typically, the "appear" relations of figure 1) within a given entity, so speeding up many queries.
- The data access made by the OO approach is very efficient for navigation through the data, which is often necessary for CAD/CAE applications.
- The OO approach in the Data Model definition allows a strict correspondence between a model or a model item and a data object. Then, code writing is easier and model manipulation (e.g. equation manipulation) is obtained through simple data manipulation [9].

As a consequence of the OO choice, the translation of the conceptual data model into the logical data model has been straightforward; figure (2) shows the highest levels of the class hierarchy, where all the fundamental entities are defined. It is evident that to any element of the MDL there corresponds one database class: since the code is associated to objects, this makes application development more "natural". As it happens in Object Oriented Programming, the class hierarchy is very similar to a taxonomy of the data represented, and this makes it easy to understand. Some exceptions are made in order to maximise reuse of code; for example, class LibraryCategory has been defined as a subclass of AggregateModel in order to reuse the code implementing submodels' management.

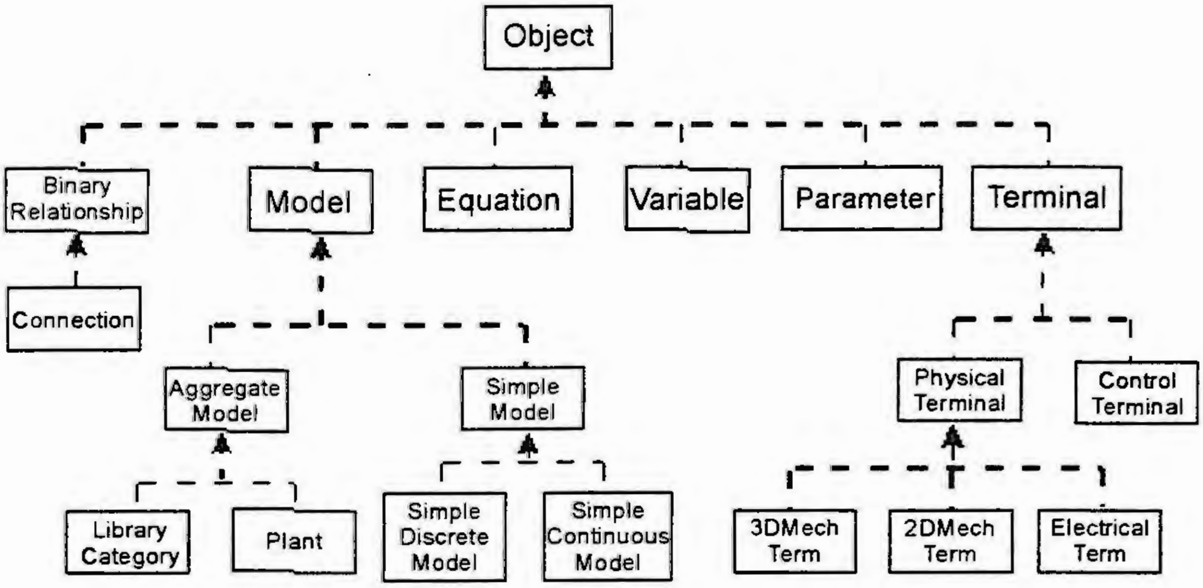


Figure 2: a partial class hierarchy

An important feature of MOSES concerns the symbolic manipulation performed to simplify the procedural form of the global system. It is completely realized in the database, and the resulting procedural form is automatically associated to the originating plant.

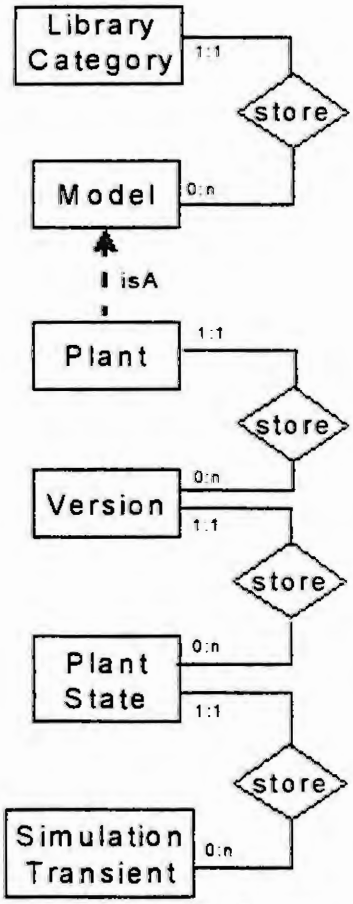


Figure 3: relation between modeling and simulation data

Proper data structures are defined to support the manipulation: equations are represented as binary trees, while very helpful is the implemented Sparse Matrix, which allows to efficiently represent and manage the incidence matrices used in various algorithms (e.g. Block Lower Partitioning, tearing [9]).

To avoid confusion, it is worth remembering that only programmers can modify the class structure, since in MOSES the user is not allowed to define classes at run time. This is one of the major differences between MOSES and other object-oriented environments like OmSim [2] (implementing Omola) and Dymola. In order to maximize modularity in models description, it has been fixed to strictly standardize the structure of models and terminals. This requires a careful design of models and terminals as fixed data structure (note that, in this way, it is possible to associate the behaviour to any object).

In figure (3) the ER diagram describing the relation between modeling and simulation data is reported: it is worth noting that the objects themselves play also the role of storage elements (a plant owns its versions, each version owns its plant states, and so on). This kind of relationship between entities is known as *existence dependence*. Its adoption is convenient both for the database designer and for the user: in fact, its implementation is very straightforward, and results in a logic correlation of all the data relative to a case study, so ensuring automatic tracing of the analysis activity. Existence dependence in our case is particularly useful for supporting deletion operations, because if a plant is deleted, all its versions and plant states are deleted at the same time. Furthermore, we note that this kind of relation is automatically managed by the system, because it is intrinsically implemented in the OO physical data model.

5- Conclusions

The use of an object oriented database to support modular modelling and simulation has been discussed. It is important to guarantee uniqueness, consistency, security and integrity while sharing data among different users. It is essential for the analyst the possibility of using certified model libraries, together with proper mechanisms supporting model consistency.

The choice to represent models in a fixed data structure gives many advantages:

- allows the creation of an open environment, because it is easy to integrate different tools,
- helps the design of a neutral format for hybrid models definition, and
- can be easily translated into standard formats like STEP/EXPRESS.

The object oriented approach facilitates the design of the database schema and allows a good matching between software and modeling items. This in turn makes it easier the application development and manipulation of data structures that is necessary in object oriented modeling.

References

[1]AMT/4/-/7 *A brief introduction to STEP*. CADDETC, Leeds, UK, 1990.

[2] Andersson, M., *Object-Oriented Modeling and Simulation of Hybrid Systems*, PhD thesis ISRN LUTFD2/TFRT--1043--SE. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

[3] Boyette, P. and Bhullar, R. S., *Challenges of tomorrow for the control systems professionals*. ISA Transactions 1994, vol. 33, pp. 197-205.

[4] Bretl R., Maier D., Otis A., Penney J., Schuchardt B., Stein J., Williams E. H., Williams M. The GemStone Data Management System. In Object-Oriented Concepts, Databases and Applications. Kim W. and Lochovsky H. editors. Addison-Wesley 1989.

[5] Cellier, F. E. and Elmqvist, H., *Automated Formula Manipulation Supports Object-Oriented Continuous System Modeling*, IEEE Control Systems, April 1993.

[6] Elmqvist, H., *Dymola - Dynamic Modeling Language, User's Manual*. Dynasym AB.

[7] James, J. and Cellier, F. and Pang, and G. Gray, J. and Mattsson, S. E., *The state of Conputer-Aided Control System -Design*. IEEE Control Systems , Vol. 15 N. 2, April 1995, pp. 6-7.

[8] Maffezzoni, C. and Girelli, R., *Object oriented database support for modular modelling and simulation*. Modelling and Simulation ESM 94, Barcelona, June 1-3, 1994, pp. 354-361.

[9]Maffezzoni, C. and Girelli, R. and Lluka, P., *Generating efficient computational procedures from declarative models*. In: Simulation Practice and Theory (editor: Dekker, L.), Elsevier Science Publisher, N. 4, pp. 303-317.

[10] Maffezzoni, C. and Magnani, G. A. and Quatela, S., *Process and control design for high temperature solar receivers: an integrated aproach*. IEE Transactions on Automatic Control, Vol. AC-30, pp. 194-209, 1984.

[11] Mattsson, S.E. and Andersson, M. and Åström, K.J., *Object Oriented Modeling and Simulation*. In: *CAD for Control Systems*, (Ed: Linkens, D.A.) Marcel Dekker, Inc. New York, chapter 2, pp. 31-69.

[12] Mattsson, S.E. and Andersson, M., *Omola-An object-oriented modeling language*. In: Recent Advances in Computer Aided Control Systems, Studies in Automation and Control (Eds: Jamshidi, M. and Herget, C.J.) Elsevier Science Publisher,Vol. 9, 291-310, 1993.

[13] Sahlin, P. and Sowell, E. *A Neutral Format for Building Simulation Models*. In: Conference proc. Building '89, IBPSA, Vancouver, Can, June 1989.

[14] Wellstead, P.E., *Physical System Modelling*. Academic Press 1979.

PROMOT/DIVA: A PROTOTYPE OF A PROCESS MODELING AND SIMULATION ENVIRONMENT

F. Tränkle, A. Gerstlauer, M. Zeitz, E.D. Gilles
Institut für Systemdynamik und Regelungstechnik
University of Stuttgart
Pfaffenwaldring 9, 70550 Stuttgart, Germany

Abstract. In this contribution, the architecture and the functionality of the modeling and simulation environment PROMOT/DIVA for chemical processes are presented. PROMOT/DIVA is supposed to interactively support chemical engineers in developing, analyzing, and solving mathematical models of individual process units and plants. A key requirement for the implementation of the knowledge-based process modeling tool (PROMOT) is a proper structuring of the chemical-engineering modeling knowledge and the definition of adequate modeling entities in a modeling concept. These modeling entities are implemented as frames with the object-oriented Model Definition Language (MDL) in the knowledge base of PROMOT. The model development with PROMOT will be illustrated by considering the modeling of a continuously stirred tank reactor.

Introduction

The applications of mathematical modeling and simulation in chemical engineering are manifold. Commercially available software tools are used for the stationary design and optimization of chemical plants. Dynamical process simulation is applied to the design and the inspection of process control systems, the startup and shutdown of plants, their behavior in case of operation faults, as well as the design and operation of inherently dynamical processes. Generally speaking, model-based techniques gain more and more importance in chemical engineering in order to improve chemical processes with respect to growing economical and environmental demands.

The current state of the art with respect to dynamical process simulation is represented by process simulation environments such as SPEEDUP [1], GPROMS [14], and DIVA [8]. These environments provide powerful numerical methods to analyze and solve systems of differential-algebraic equations that constitute the process models. Their model libraries contain modeling entities in the granularity of individual process units which can be aggregated to process models by specifying the flow sheet of the considered plant. Unfortunately, these simulation tools give only very limited support for the development of new unit models as well as for the reuse and the documentation of existing unit models [13]. For this reason, the development and validation of adequate process models strongly limits the routine application of model-based techniques in process design as well as process operation in the chemical process industries [11].

To overcome this problem, considerable efforts have been initiated during the last decade to develop general as well as domain specific modeling languages and knowledge-based process modeling tools [11, 15]. These tools are supposed to be integrated into process simulation and design environments to assist the chemical engineer, i. e., to facilitate the modeling process, to suggest modeling assumptions, and to guarantee the consistency of the process models being developed. Despite the advances, the developed knowledge-based modeling tools show several shortcomings. For instance, adequate concepts and language constructs to design, implement, handle, and document large chemical-engineering specific knowledge bases are scarcely available. A general problem is the lack of adequate programming and knowledge representation languages for an easy and efficient implementation of knowledge-based process modeling tools.

In this contribution, an overview of the modeling and simulation environment PROMOT/DIVA for chemical processes is presented. PROMOT/DIVA is supposed to interactively support chemical engineers in developing, analyzing, and solving mathematical models of individual process units and plants. The following section introduces the architecture of PROMOT/DIVA. In the subsequent sections, the underlying modeling concept of the knowledge-based process modeling tool (PROMOT) and the modeling of a continuously stirred tank reactor with PROMOT are illustrated.

The Architecture of PROMOT/DIVA

The modeling and simulation environment PROMOT/DIVA consists of the knowledge-based process modeling tool PROMOT, a pre-processing tool and a code generator [18], the process simulation system DIVA [8], and graphical as well as text-based user interfaces (see Figure 1). PROMOT is supposed to interactively support the modular development of process unit models. Its object-oriented knowledge base contains general chemical-engineering modeling knowledge, from which modelers can interactively browse information and select *modeling entities* with the graphical *model entity browser*. The knowledge base is persistently stored in the *model library*, whose contents can be loaded to the temporary *workspace*. With either the *graphical* or the *textual editor* for the Model Definition Language (MDL), the modeler interactively develops *process unit models* by aggregating and specifying *modeling entities* in the workspace. Once a process unit model is completed, it can be saved to the *model library* for later reuse or exchange with collaborating modelers.

The *code generator* [18] can be called by PROMOT to translate the internal symbolic representation of process unit models selected from the knowledge base to DIVA simulation modules. In the case of partial differential equations (PDEs) or differential-algebraic equations (DAEs) with differential indices greater than one, the mathematical models of the selected unit models have to be pre-processed. The *pre-processing tool* [18] provides algorithms to perform a method-of-lines semi-discretization of PDEs by finite-differences methods. Thus, PDEs are transformed to sets of differential and algebraic equations that can be solved simultaneously with the other parts of the model. Furthermore, the differential index of DAEs can be analyzed. If required, the DAEs can be transformed to index-one-systems.

After pre-processing and code generation, the process unit models are added to the *model library* of the block-oriented process simulation environment DIVA [8]. Each individual process unit model is represented in sparse-matrix form as a set of at least three FORTRAN subroutines and one data file [18]. The FORTRAN subroutines are called during the run-time of DIVA for the initialization of the individual unit model, the evaluation of the model equations, and the output of the process values as simulation data. Optional subroutines may add discrete events and an analytical computation of the Jacobians of the model equations. The individual process unit models can be interconnected to simulation-ready plant models by specifying plant flow sheets, for example in a graphical way [2]. DIVA provides algorithms for the computation of time trajectories, for the analysis of the nonlinear dynamics, and for the parameter identification of the plant models, among others [8]. The simulation data can be displayed with the DIVA *graphics*.

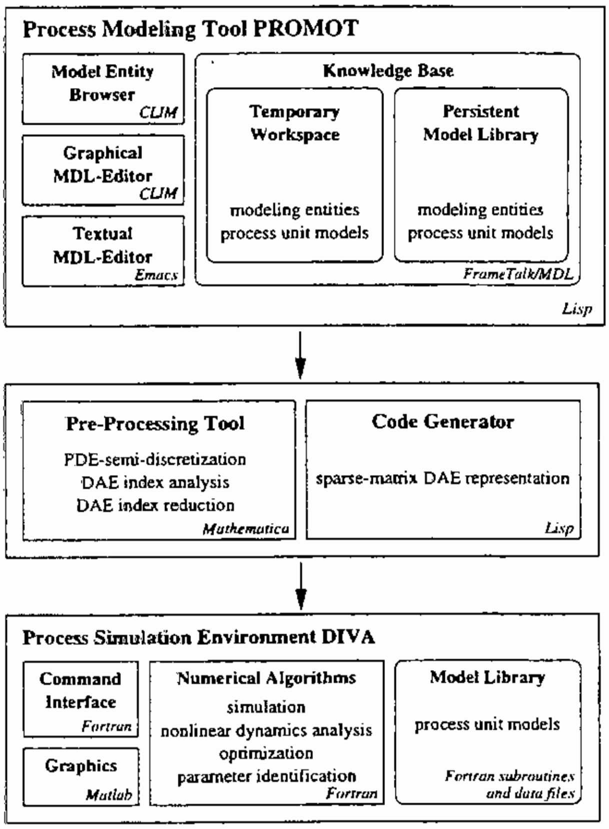


Figure 1: Architecture of the Modeling and Simulation Environment PROMOT/DIVA.

PROMOT and the code generator are implemented as object-oriented program modules in Allegro Common Lisp by Franz Inc. [6]. The modeling entities in the temporary workspace and the persistent model library are defined in the object-oriented Model Definition Language (MDL), which is an extension to the frame language FRAMETALK [16, 17]. With the object-oriented Common Lisp Interface Manager (CLIM) [6], being part of Allegro Common Lisp, the modeling entities are naturally mapped to graphical objects in the *model entity browser* and the *graphical MDL-editor* of PROMOT. The text editor GNU Emacs [5] is used as the *textual MDL-editor*, which is integrated into the graphical user interface of PROMOT with the Emacs communication interface of Allegro Common Lisp [6].

The Modeling Concept of PROMOT

A key requirement for the implementation of a knowledge-based process modeling tool is a proper structuring of the chemical-engineering modeling knowledge and the definition of modeling entities in a modeling concept. First ideas for such a modeling concept were presented in [7, 9, 12, 13]. Subsequently, Marquardt et al. refined and developed this concept further [3, 4, 10]. The modeling concept, on which the knowledge base of PROMOT is based, also originates from the early works cited above. However, during the application of this modeling concept to the modeling of (reactive) distillation processes, it became apparent that several modifications had to be made, in order to allow an easy and efficient representation of the process models [19].

One of the differences in the modeling concept is that in PROMOT elementary and composite modeling entities are defined on three hierarchical levels. The hierarchical levels considered so far are the *process unit level*, the *phase level*, and the *storage level*. On all levels, elementary and composite *structural*, *behavioral* and *material modeling entities* are defined and arranged in specialization and aggregation hierarchies. The specialization hierarchy follows from is-a relationships between the modeling entities, whereas the aggregation hierarchy results from is-part-of relationships between the modeling entities. Figure 2 depicts an extract of the specialization hierarchy of the *structural modeling entities* on the *phase level*.

Structural modeling entities describe the topological structure of a model on each level. There are three main types of structural modeling entities: "devices", "connections", and "terminals" (see Figure 2). Each "elementary device" represents a balance space with a finite extension and has a set of "terminals" that define its interactions with adjacent "connections". In contrast, "elementary connections" are very thin in one spatial dimension and are interconnected with adjacent "devices" via their "terminals" [19]. Examples of such "terminals" are "convective-flow-terminals" and "exchange-terminals" (see Figure 2). *Behavioral modeling entities* (equations, variables, and equation terms) are attached to each structural modeling entity to represent its mathematical model. *Material modeling entities* are used for the description of the pure chemical substances and their mixtures being situated in a structural modeling entity.

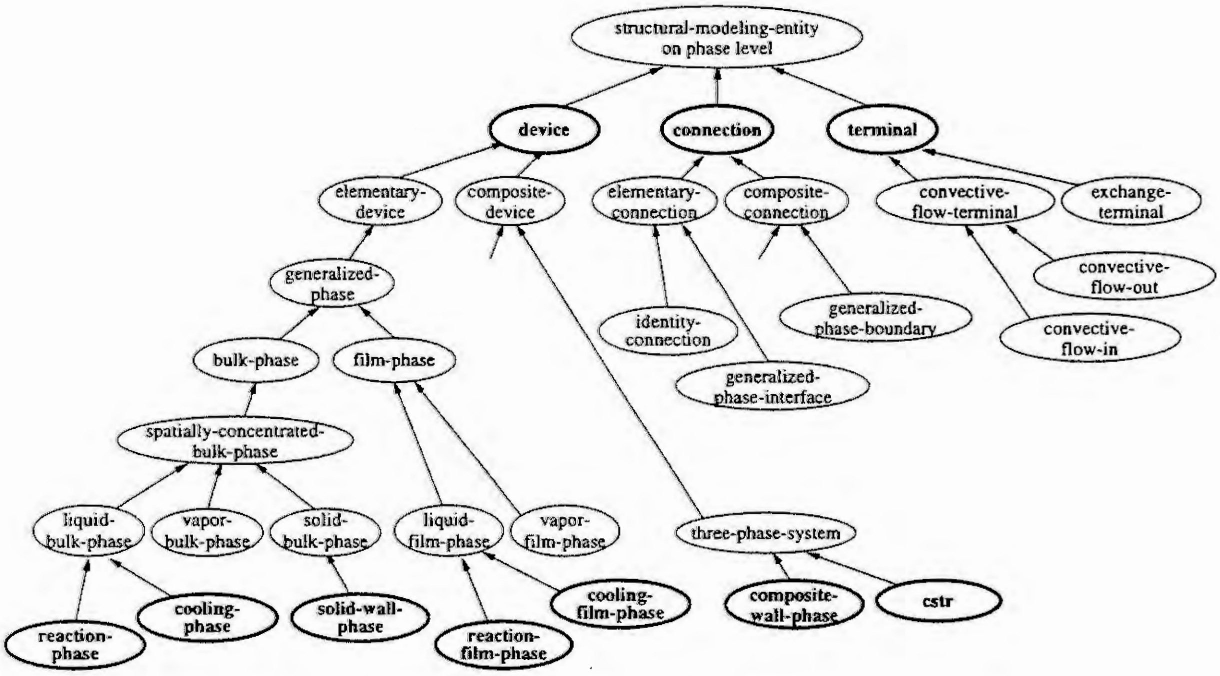


Figure 2: An extract of the specialization hierarchy of structural modeling entities on the *phase level*. The arrows represent *is-a relationships* between the modeling entities.

The "elementary devices" and "connections" on the *process unit level* are "process units" (e. g., tanks, heat exchangers, chemical reactors, and distillation columns) and "pipelines", respectively. Examples of generic "elementary devices" and "elementary connections" on the *phase level* are "bulk-phases" or "film-phases", and "identity-connections" or "generalized-phase-interfaces", respectively [19] (see Figure 2). From these generic modeling entities, fully specified modeling entities may be derived by specialization, e. g., the elementary devices "reaction-phase", "cooling-phase", "solid-wall-phase", "reaction-film-phase", and "cooling-film-phase". Next to the elementary modeling entities, the specialization hierarchy contains generic composite devices (e. g., "three-phase-system") and composite connections (e. g., "generic-phase-boundary") that are aggregated from other composite or elementary modeling entities. By specialization of these composite modeling entities and by aggregation of the elementary modeling entities, a model of a continuously stirred tank reactor (CSTR) will be derived in the next section.

Similar to the structural modeling entities, the behavioral modeling entities are also arranged in specialization and aggregation hierarchies. Elementary and composite behavioral modeling entities describe the dynamical behavior of the structural modeling entities. Examples of behavioral modeling entities are balance equations, phenomenological relations, and physical property correlations. It should be noted at this point, that a major part of the chemical-engineering modeling knowledge lies within the behavioral modeling entities. Hence, a proper modeling concept has to address this point in detail.

The main focus thus far is on the definition and implementation of the structural, behavioral and material modeling entities on the *phase level.* All modeling entities are implemented as frames with the object-oriented Model Definition Language (MDL) in the knowledge base of PROMOT. MDL is an extension to the frame language FRAMETALK [16, 17], with language concepts specific to the representation of chemical-engineering modeling knowledge. Each MDL construct representing a structural modeling entity has a textual as well as a graphical form of presentation. When manipulating the graphical form, the changes are propagated to the textual form and vice versa.

MDL is not only the implementation language of the knowledge base, but also the modeling language of PROMOT. With MDL modelers can successively aggregate composite structural modeling entities from more fine-grained modeling entities, as illustrated in the next section. Each structural modeling entity has a set of text attributes, which are informal representations of the physico-chemical assumptions and simplifications that lead to the mathematical model of the structural modeling entity. According to these attributes, the modeler selects predefined behavioral and material modeling entities from the knowledge base and attaches them to the structural modeling entity.

A Modeling Example

For illustrating the model development with PROMOT, the modeling of a continuously stirred tank reactor (CSTR) (see Figure 3) is considered as a simple example of a process unit in chemical engineering. The CSTR is fed with the chemical substances A and B that react to the substances C and D in an exothermic liquid phase reaction. For cooling purposes, the liquid reaction phase exchanges heat with the enclosing liquid cooling phase across the wall of the tank.

For simplicity reasons, in the following we only consider the aggregation of the reactor model from elementary structural modeling entities and do not show its behavioral and material modeling entities. The reactor model is aggregated from the devices "reaction-phase", "composite-wall-phase", and "cooling-phase" which are connected by simple "identity-connections" (see Figure 2). Each modeling entity is given a unique name: "RPH", "CWPH", "CPH", "ID1", and "ID2", respectively. The modeling entity "composite-wall-phase", representing the wall between the reaction phase and the cooling phase, is modeled as a composite device. It is aggregated from the elementary devices "reaction-film-phase", "solid-wall-phase", and "cooling-film-phase" called "RFPH", "SWPH", and "CFPH", respectively. Figure 4 sketches the graphical representations of the reactor model and the wall model in the graphical MDL-editor, whereas Figure 5 shows the textual representation of the reactor model as a MDL expression in the textual MDL-editor.

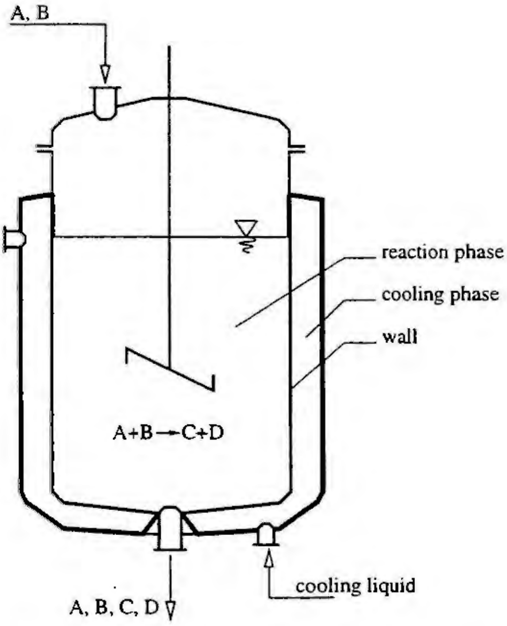


Figure 3: Continuously stirred tank reactor (CSTR) with a reaction, a wall and a cooling phase.

The reactor model is defined as the MDL frame "cstr" which is a sub-class of the MDL frame "three-phase-system" (see Figure 2). "Three-phase-system" is a direct sub-class of "composite-device", which is the root-class of all process unit models in PROMOT. The components of "cstr" are interconnected via "exchange-terminals" as defined by the double-sided arrows in Figure 4 and the MDL construct ":links" in Figure 5. The interconnections of the terminals via the "identity-connections" state the equality of input and output variables. For instance, by interconnecting the "reaction-phase" "RPH" with the "composite-wall-phase" "CWPH" via the "identity-connection" "ID1", the input variable T_{w1} of "CWPH" is set equal to the output variable T_r of "RPH", where T_{w1} is the temperature on the left-hand side of "CWPH" and T_r is the internal temperature of "RPH" .

The reactor model has four convective flow terminals called "rph-in", "rph-out", "cph-in", and "cph-out". The

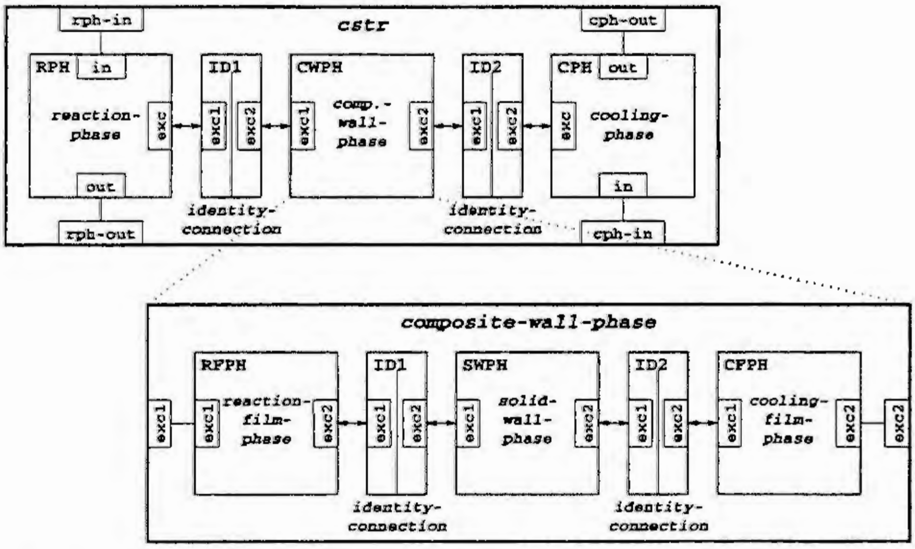


Figure 4: The MDL frames of the reactor model "cstr" and the wall model "composite-wall-phase" in the graphical MDL-editor.

```
(define-device :class "cstr"
  :super-classes ("three-phase-system")
  :documentation "CSTR with a reaction, a cooling and a wall phase"
  :components (("RPH"       :is-a "reaction-phase")
               ("CWPH"      :is-a "composite-wall-phase")
               ("CPH"       :is-a "cooling-phase")
               ("ID1"       :is-a "identity-connection")
               ("ID2"       :is-a "identity-connection"))
  :terminals   (("rph-in"    :is-eq-to "RPH.in")
               ("rph-out"   :is-eq-to "RPH.out")
               ("cph-in"    :is-eq-to "CPH.in")
               ("cph-out"   :is-eq-to "CPH.out"))
  :links       (("RPH.exc"   :is-linked-to "ID1.exc1")
               ("CWPH.exc1" :is-linked-to "ID1.exc2")
               ("CWPH.exc2" :is-linked-to "ID2.exc1")
               ("CPH.exc"   :is-linked-to "ID2.exc2")))
```

Figure 5: The MDL expression defining the reactor model "cstr" in the textual MDL-editor.

terminal "rph-in" is identical to the terminal "in" of the reaction phase, i. e., the reaction feed is the inlet of the reaction phase. Similar identities exist for the terminals "rph-out", "cph-in", and "cph-out". These identities are expressed by the straight lines in-between the terminals in the graphical MDL-editor (Figure 4) and by the MDL construct ":is-eq-to" in the textual MDL-editor (Figure 5).

The pre-defined structural modeling entities "reaction-phase", "cooling-phase", "reaction-film-phase", "solid-wall-phase", and "cooling-film-phase" contain the model equations in the form of behavioral modeling entities (e. g., material balance equations, enthalpy balance equations, physical property correlations, reaction kinetics, and heat transport laws). The model equations of the "identity-connections" are simple identities of the adjacent state variables and fluxes. PROMOT extracts these modeling entities from its knowledge base and aggregates them to a complete mathematical model of the CSTR. This model is a DAE in semi-implicit form with a differential index of one including a set of initial conditions for its dynamical state variables [18]. The code generator translates this model to a DIVA simulation module that can be interconnected with other simulation modules via pipe-lines for the reaction and cooling phase inlets and outlets.

Conclusions

In this contribution, the functionality of PROMOT was illustrated for the example of the modeling of a continously stirred tank reactor (CSTR). It was shown how the model of the CSTR is represented as MDL frames in the graphical MDL-editor as well as the textual MDL-editor of PROMOT. The main focus thus far is on the definition and implementation of the structural, behavioral and material modeling entities on the *phase level* for a representation of the topological structure of a process unit, its mathematical model, and the occuring chemical substances. Future investigations will include the extension of the language concepts and constructs of MDL for an implementation of partial differential equations (PDEs).

Acknowledgment. This research is sponsored by the Deutsche Forschungsgemeinschaft (SFB 412).

References

1. Aspen Technology. *SpeedUp User Manual.* Aspen Technology, Inc., Cambridge, Massachusetts, 1994.
2. M. Bär and M. Zeitz. A Knowledge-Based Flowsheet-Oriented User Interface for a Dynamic Process Simulator. *Comp. Chem. Engng.*, 14:1275–1280, 1990.
3. R. Bogusch, B. Lohmann, and W. Marquardt. Computer-Aided Process Modeling with ModKit. To be published in Proc. Chemputers Europe III, Oct. 29 – 30, Wiesbaden, Germany, 1996.
4. R. Bogusch and W. Marquardt. A Formal Representation of Process Model Equations. *Comput. Chem. Engng.*, 19:211–216, 1995.
5. D. Cameron and B. Rosenblatt. *Learning GNU Emacs.* O'Reilly, 1991.
6. Franz Inc. *Allegro Common Lisp User Guide, Version 4.2, and CLIM 2 User Guide, Version 2.0*, 1994.
7. A. Gerstlauer, M. Hierlemann, and W. Marquardt. On the Representation of Balance Equations in a Knowledge-Based Process Modeling Tool. In *11th Congress of Chem. Engng., Chem. Equip. Design and Automation*, Praha, Czech. Republic, 1993.
8. A. Kröner, P. Holl, W. Marquardt, and E.D. Gilles. DIVA – An Open System for Dynamic Process Simulation. *Comput. Chem. Engng.*, 14:1289–1295, 1990.
9. W. Marquardt. Rechnergestützte Erstellung verfahrenstechnischer Prozeßmodelle. *Chem.-Ing.-Tech.*, 64:25–40, 1992. English translation in Int. Chem. Engg. 34 (1994).
10. W. Marquardt. Towards a Process Modeling Methodology. In R. Berber, editor, *Model-based Process Control, NATO-ASI Ser. E, Applied Sciences.* Kluwer Academic Publ., 1996.
11. W. Marquardt. Trends in Computer-Aided Process Modeling. *Comput. Chem. Engng*, 20:591–609, 1996.
12. W. Marquardt, A. Gerstlauer, and E.D. Gilles. Modeling and Representation of Complex Objects: A Chemical Engineering Perspective. In *6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 219–228, Edinburgh, Scotland, June 1993.
13. W. Marquardt and M. Zeitz. Rechnergestützte Modellbildung in der Verfahrenstechnik. In *I. Troch, editor; Modellbildung für Regelung und Simulation. Methoden – Werkzeuge – Fallstudien. VDI-Berichte 925, 307-341*, VDI-Verlag, Düsseldorf, 1992.
14. M. Oh and C.C. Pantelides. A Modeling and Simulation Language for Combined Lumped and Distributed Parameter Systems. In *International Conference on Process Systems Engineering PSE'94*, pages 37–44, Kyongju, Korea, 1994.
15. C.C. Pantelides and H.I. Britt. Multipurpose Process Modelling Environments. In L.T. Biegler and M.F. Doherty, editors, *Conference on Foundations of Computer-Aided Process Design*, pages 128–141. CACHE Publications, 1995.
16. C. Rathke. *Using the CLOS Metaobject Protocol to Implement a Frame Language*, chapter 8, pages 129–143. CRC Press, Inc., Boca Raton, Fla., 1996.
17. C. Rathke and F. Tränkle. An Extensible Frame Language for the Representation of Process Modeling Knowledge. Submitted to IEA-AIE, Atlanta, Georgia, June 10–13, 1997.
18. S. Räumschüssel, A. Gerstlauer, E.D. Gilles, and M. Zeitz. Ein Präprozessor für den verfahrenstechnischen Simulator DIVA. In G. Kampe and M. Zeitz, editors, *Simulationstechnik, 9. ASIM-Symposium in Stuttgart, Germany*, pages 177–182, Braunschweig/Wiesbaden, 1994. Vieweg Verlag.
19. F. Tränkle, A. Gerstlauer, M. Zeitz, and E.D. Gilles. Application of the Modeling and Simulation Environment PROMOT/DIVA to the Modeling of Distillation Processes. Accepted to PSE'97, ESCAPE–7, Trondheim, Norway, May 26–29, 1997.

NETSIM–NETCALC
A Complete Papermill Simulation Package

Eugen Brenner
University of Technology Graz
Steyrergasse 17/IV, A-8010 Graz, Austria
brenner@iti.tu-graz.ac.at

Jürgen K. Weinhofer
Johannes Kepler University Linz
Altenbergerstraße 69, A-4040 Linz, Austria
juergen@regpro.mechatronik.uni-linz.ac.at

Abstract. The goal of this project was the development of a tool to support the designers of papermills. Their primary goal in return is to keep the quality of the final paper in a desired range. Therefore, the developers are especially interested in the streams of material, consisting of suspended, dissolved, and vapor state phases. Due to environmental and economical reasons, parts of these streams circulate in the papermill. Algebraic loops, which can hardly be solved by hand, are the result. Therefore, a graphical user interface (NETSIM) and a calculation program (NETCALC) were developed, which communicate through a standardized text interface. These programs support the interactive design of the plant network and the complete calculation out of this graphical representation. The number of different material streams is not limited and desired values for mass flow, volume flow, or consistency of streams can be given at arbitrary points of the network.

1 Introduction

Papermills are a typical example in the area of process engineering. A plant is built from a number of units like pulpers, cleaners, or disc filters. These units are connected with pipes for the transport of the material. Typically, raw wood and old paper are crushed and pulped (suspended) in water. The chemical pulp is then cleaned in various stages and treated with chemicals to obtain the desired quality. These stages also require the addition of water to obtain the necessary consistency for each stage of treatment. Finally, the paper or board is produced by means of sieves or pressure wires. This stage delivers large quantities of water in different qualities, which in turn are cleaned and reused in the earlier stages of the treatment. This fact is responsible for the high complexity of the calculation, as it delivers a system with a large number of interconnected meshes.

When offering the construction of a plant, the firm has to guarantee the quality of the final product under usual working conditions within a defined production range, e.g. between 50% and 120% of the nominal production. Within these production limits, certain aggregates require a constant volume flow in order to work properly, others require the consistency of the flow to remain constant [1]. As various suppliers deliver a number of aggregates with varying operating limits, one of the tasks is to find the ideal set of aggregates. Therefore, it is necessary to calculate the system with different parameters and with minor modification in the network. This requires easy to handle facilities to alter aggregate parameters or the network setup, which can only be delivered by means of a graphical user interface.

2 Package requirements

Depending on the quality of raw materials and of the desired paper quality, developers have to consider not only primary fibers of different sizes, but also foreign contraries like ash or dirt and chemical ingredients. This requires the calculation of a larger number of phases, each one representing one of the components of the material stream.

Until recently, most of the calculations were done by hand, sometimes with the help of computer programs, that allowed only the calculation of small sections of the network. Usually, the definition of desired values at arbitrary points within the network was not possible and had to be iterated by means of starting values.

The ultimate goal for the design of the package was to improve the working facilities of the user as far as possible. The plant designer should work in an intuitive manner. He selects aggregates out of a catalogue, places them on the drawing plane and connects them with pipes. He then selects a set of parameters and desired values of mass flow, volume flow, or consistency for every aggregate. As different aggregate suppliers often use different symbols for the same type of aggregate, the user should be free to independently connect drawing and content, where content is constituted out of type definition and parameter set. In some cases, aggregates even combine the functionality of other aggregates to form a new unit, characterized by a new symbol. The results of these steps and the additional calculation are values of mass flow, volume flow, or consistency and are labeled to the pipes, where the international environment requires the output to be shown in different units. Additional text and graphics are added to form a document, which could be included in offers or operating instructions.

These requirements can only be fulfilled with a graphical user interface, which is best built on base of an existing set of libraries and/or an existing drawing program.

As the resulting network delivers a set of meshed nonlinear equations, only iterative methods can be used. To improve flexibility and development time, the user interface and calculation program were defined as two separate parts, connected only by ASCII-text files.

3 Aggregate modelling

In order to cover a broad range of elements produced by different suppliers, the base characteristics of the aggregate types had to be identified. To give an example, let us consider a cleaning element. This aggregate should remove dirt particles from the material stream. Hence, it consists of two outputs, cleaned mass flow and reject, and one or two inputs, depending on the supplier. Typical parameters given are cleaning efficiency EF, reject rate UEM, and the relative parameters ETA_i. Mathematically, this element can be described by

$$EF = \frac{O\_2c}{I\_1c} \qquad UEM = \frac{O\_2ms}{I\_1ms} \qquad ETA_i = 1 - \frac{O\_1k_i}{I\_1k_i} \quad i \geq 2,$$

where O_1 denotes the first output, O_2 the second output, I_1 the first input, ms the suspended mass flow, ms_i the individual phase of ms, v the volume flow, $c = ms/v$ the consistency, and $k_i = ms_i/ms$ the individual phase ratio.

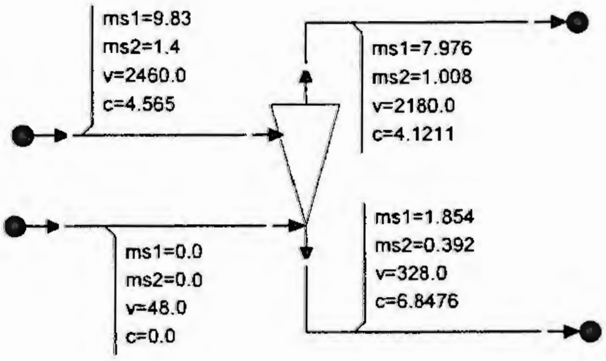


Figure 1: Heavy particle cleaner

Figure 1 shows a heavy particle cleaner with input and output values. The consistency is calculated by $c = (ms1 + ms2)/v$ and given as a percentage or per mil value. The higher portion of heavy particles in the reject can be seen clearly. To reduce the amount of wood pulp in the reject, cleaners are combined to form a multi-stage cleaning section as shown in figure 2. Only a small amount of particles, in this case light weight dirt, is finally dumped as waste.

About a dozen basic elements were defined in a similar manner, allowing to describe most basic elements or at least their main characteristics.. These plain elements are objects in the sense of object-oriented programming, each representing a type definition, a user-definable set of parameters, a user-definable set of values to be used in a label list, and a graphical symbol which again can be defined by the user. To allow the handling of more complex aggregates or sections of complete networks, these basic elements may be grouped together to form a new hierarchical element, represented by a new symbol. As the basic elements remain accessible, this combination does not require a parameter set for its own. Another possibility to define elements is to use their input-output matrix equation explicitly, which allows a wide range of elements to be defined.

For technological reasons, the user requires the definition of values within the network like mass flow, volume flow, or consistency. For example, cleaning stages work best when the volume is kept constant, screening elements or certain types of sieves require the consistency within a certain range, or the overall

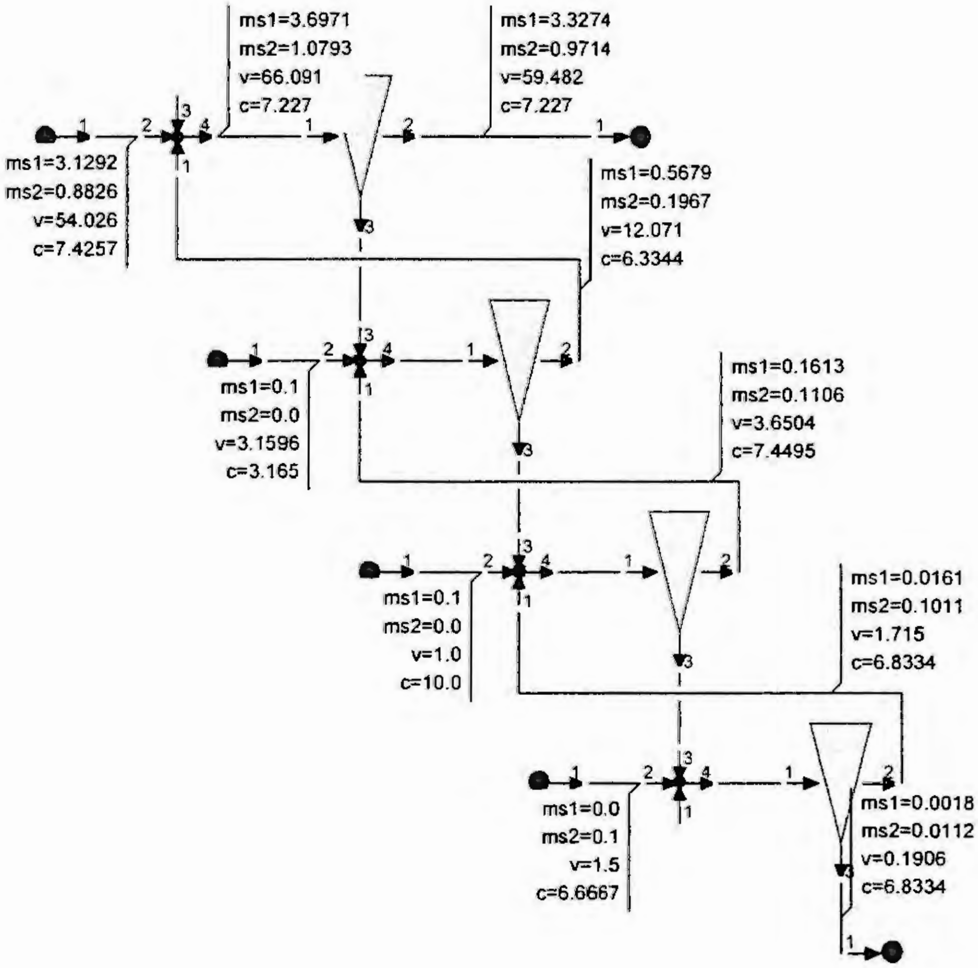


Figure 2: Four-stage cleaning section

production requires the amount of wood pulp to be defined in the last stage of the whole network. Therefore, the description of the elements had to be extended by a number of target values. These designer's goals require the same amount of free variables that have to be iterated to fulfill the requirements. To improve the calculation speed we additionally allowed the definition of starting values for the iteration as described in section 5.

For easier programming and to keep the user interface as independent as possible from the actual application we decided to split the package into two independent parts, the user interface NETSIM and the calculation program NETCALC.

4 User interface

NETSIM is a general purpose interface for various kinds of block-oriented simulators with either stationary or dynamic results. The development over several years included hydraulic simulation [3], process interfacing [6] and finally wood pulp production [1]. Basically the program is a drawing tool. Functions like rectangle, ellipse, polyline, or polygon drawing can be combined with a broad selection of fill styles, line types and widths or any kind of text. Every element is an object that can be scaled, rotated, flipped, etc., and whose properties can be changed. Objects can be combined to an arbitrary depth, where any compound object can include others. NETSIM was developed based on Borland's OWL-libraries and is fully object-oriented to allow quick and easy adaptation to different requirements and multi-language support. It follows the philosophy of "Select a Tool and Apply" as far as possible, therefore not only minimizing the training phase but also the relearning period when using the tool only every now and then. The interface follows the usual MS-Windows conventions and uses the available printing and clipboard facilities, therefore allowing importing and exporting of drawings.

Only a minor set of extensions is necessary to build a full simulator interface.

1. We distinguish two modes: one to define aggregates, the other to define networks. The network mode treats aggregates as special kind of compound objects, that are inseparable and supplied with additional methods like connecting or accessing the set of parameters. These can only be altered in aggregate mode.
2. Special kinds of lines can be used to connect inputs and outputs of aggregates. These lines are objects themselves and therefore are supplied with a parameter set.
3. Aggregates in the sense mentioned above and their interconnecting lines form a full graphical representation of a network. Nodes are formed by aggregates, edges are built by lines. Each object has its own set of parameters, describing their type and mathematically relevant values.
4. The interface dumps this description into a file, starts the simulation program, and waits for the results.
5. Depending on the type of simulation program either stationary or transient results are then taken from the simulator output and written into the label flags.

With these properties the interface is nearly independent from the actual simulation program, as type definitions or parameters are bound to objects and not compiled into the program anywhere. To sum up the properties, NETSIM allows free design of symbols for building blocks e.g. "aggregates" which can be combined to form complex networks. It allows for hierarchical combination of subnets to single blocks that can be used as elements like plain aggregates. Parameter sets of aggregates can be defined and used in any combination as required. Various options allow the choice of simulator input and output format, conversions from SI-units to commonly used units and an arbitrary selection of output data. As the drawing not only is the simulator input but also includes the output in the user-requested way, it

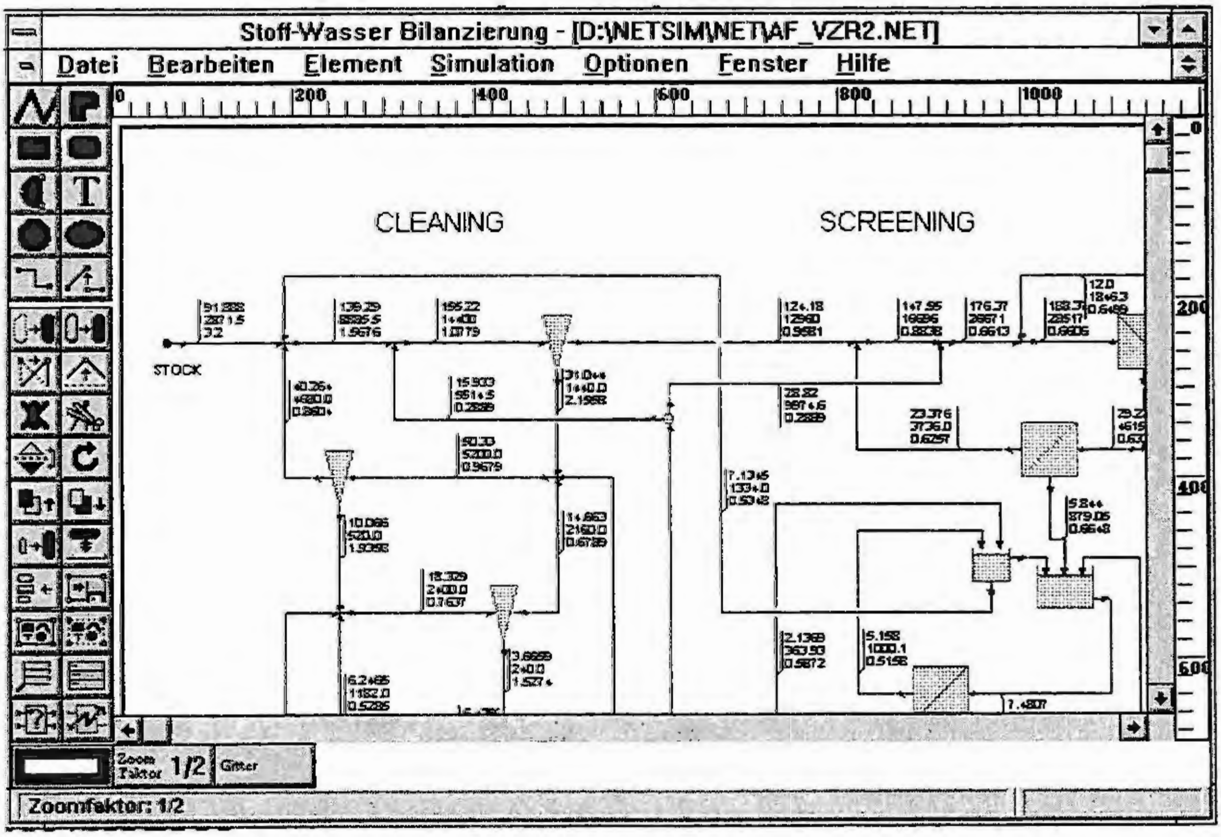


Figure 3: User interface NETSIM

is actually its own documentation, which is also supported by the possibility of adding arbitrary text to supply complete information and documentation for the user.

Figure 3 shows a screen dump of the interface with drawing and manipulation tools, status line and selection fields.

5 Calculation program

The tasks of NETCALC include the preparation of the input data, the calculation of the streams, the variation of the input streams to reach the designer's goals, and the construction of the output data. For the computer representation the object-oriented programming language C++ was chosen. Each aggregate is now one object and all objects are located in a tree structure. This setup improves not only the data representation but also the calculation algorithm and expandability. For the calculation of the streams the papermill net is searched with a Top-Down-algorithm for algebraic loops. Thereafter, these loops and the designer's goals are solved with either a Newton-algorithm or a Steepest-Descent-algorithm. The operator overloading ability of C++ was especially helpful for calculating the partial derivatives for both algorithms.

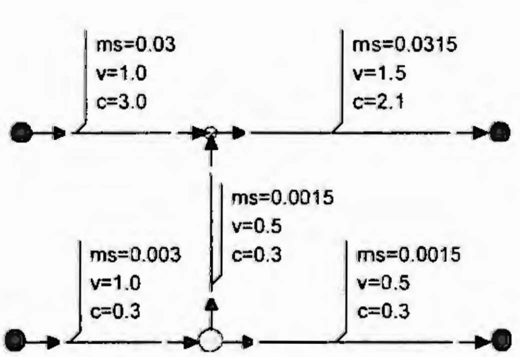


Figure 4: Example network for convergence behavior

One of the most challenging problems is to fulfill designer's goals. For every goal that is defined, one value in the system has to be a free variable. This can either be an input value of the system or a branch value like in figure 4. In this example, the designer's goal is a consistency of 2.1 in the upper output while there are two input streams available with consistencies of 3.0 and 0.3 respectively. The solution seems to be straight forward, but when looking at iterative solutions the problem becomes apparent. Depending on the starting value of the branching flow, the iterative solution converges to the correct result or does not deliver a result at all. This is still no problem with networks like the one in the example. In larger networks however, up to several dozens of meshes have to be solved, where one wrong starting value can prevent the finding of a solution.

One major improvement for this problem was the definition of a constant branching ratio and a variable deviation of the volume mass to be iterated. This dramatically improved the convergence behavior and therefore the number of necessary iterations. This is also a good solution for practical work, as these values can be easily determined by rough hand calculations or from similar networks. This solution in many cases is close enough to deliver a feasible solution even on the first try. In our example above with a ratio of 0.5, any deviation smaller than 1.5 would deliver the solution. Actual networks containing several hundred elements, more than 50 meshes and about 30 designer's goals were successfully calculated.

Figure 5 shows a complete network that simulates a paper production line.

6 Conclusions

The simulation package shown here demonstrates the enormous influence of modern technology on a wide range of areas. In our case, the calculation time for a small plant like the one shown in figure 5 was reduced from some weeks to one or two days, depending on the experience of the plant designer. Some of the models could never be calculated before and brought a much deeper understanding of what is actually going on in some of the networks. Extensions currently under development deal with thermal behavior and should be able to calculate the energy flow in these plants, or the inclusion of dissolved phases that represent chemicals necessary for the wood pulp production.

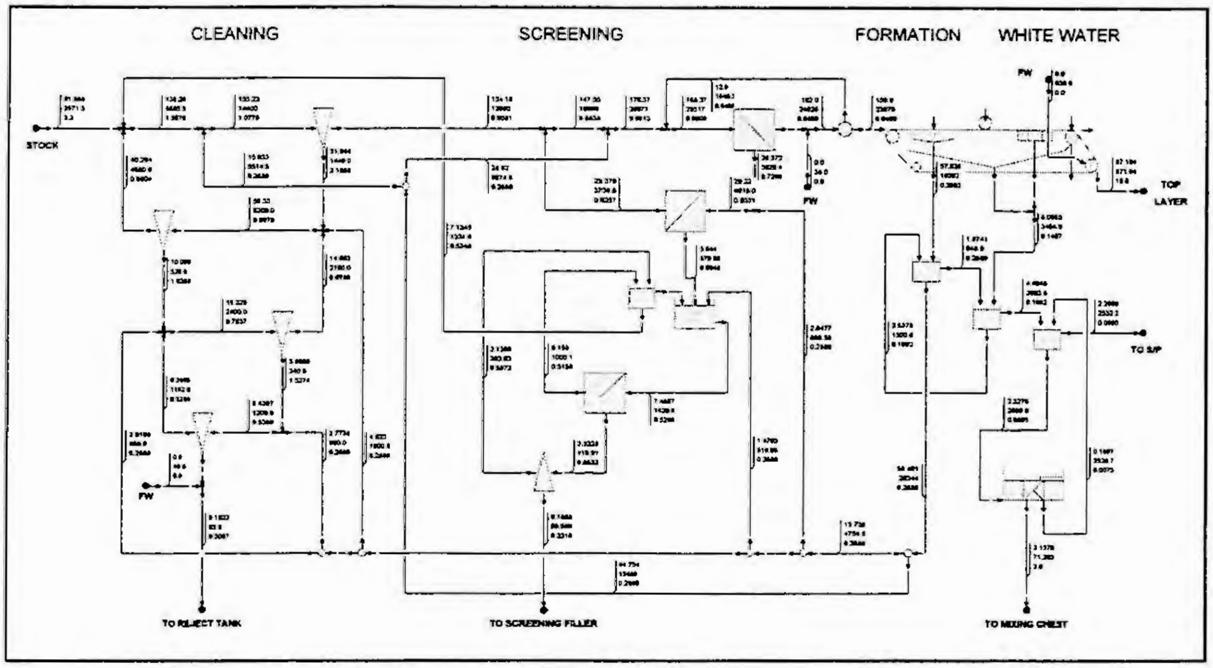


Figure 5: Complete papermill network

A user interface that is nearly independent of the actual simulation program allows a wide range of block-oriented programs to be interfaced as shown in [4] or [5], where various kinds of simulators were included. [2] shows the usage of 5 different simulators in one interface program and demonstrates the possibility to integrate even existing simulators.

Acknowledgment. Special thanks have to be given to Mr. Dorneger of Andritz AG, who brought in the technological knowledge, and Mr. Unterweger, who was responsible for the implementation of the user interface.

References

[1] Brenner, E., Dorneger, R., Weinhofer, J. and Unterweger, D., NETSIM - Ein Stoff-Wasserbilanzierungssystem mit integrierter graphischer Benützeroberfläche. Technischer Bericht, Institut für Technische Informatik, TU Graz, 1995.

[2] Brenner, E. and Unterweger, D., WEDSIM - Ein Windows EDitor SIMulator. Technischer Bericht, Institut für Technische Informatik, TU Graz, 1995.

[3] Brenner, E. and Weiß, R., Ein interaktives Entwurfs- und Simulationssystem für Hydraulikanwendungen. In: Simulationstechnik, (Eds.: Breitenegger, F., Troch, I. and Kopacek, P.) Vieweg, Braunschweig, September 1990, 377 - 384.

[4] Egger, A., Entwurf und Implementierung einer generischen Benutzeroberfläche für verteilte Simulationssysteme. Diplomarbeit, Institut für Technische Informatik, TU Graz, 1992.

[5] Natter, A., Integrierte graphische Simulationsumgebung unter MS-Windows. Diplomarbeit, Institut für Technische Informatik, TU Graz, 1995.

[6] Unterweger, D., Ein objektorientierter Fließbildgenerator. Diplomarbeit, Institut für Technische Informatik, TU Graz, 1992.

COMPUTER AIDED ANALYSIS OF SOME NONLINEAR PHENOMENA IN RELAY SYSTEMS

A. Moeini and D. P. Atherton
School of Engineering, University of Sussex, Brighton BN1 9QT, U.K.
email address: atherton@sussex.ac.uk

Abstract: The paper describes a software package written in MATLAB to determine exactly any limit cycles and their stability in a relay feedback system. The software uses analytical methods based on the Tsypkin approach and several examples of its use are given.

1 Introduction

Many control systems, for example, on-off temperature control systems, employ relay-type elements and in recent years relays have been intentionally used in feedback systems to determine suitable parameters for controllers. This latter approach, known as auto-tuning, has received appreciable attention during the last decade [1].

The possibility of limit cycles in a feedback loop containing a nonlinear element can be investigated approximately using the describing function method and this approach can also be used when the nonlinearity is a relay. The relay, however, is a unique nonlinear element in that the output does not depend upon the precise value of the input at each instant of time. The output of the relay only changes when the relay input passes through the switching levels. It is this unique feature of relays which enables the determination of limit cycles in relay feedback systems to be done using a special method which gives exact results for the limit cycle and also for its stability. The approach, known as the A function method, is briefly outlined in Section 2.

Based on the A-function method, software called Relay has been written to find limit cycles in the basic relay feedback control system shown in Fig. 1. This is one of several special purpose software packages which have been developed to work in the MATLAB environment. The first of these was the Control Kit [2] used for teaching undergraduate students which was developed in co-operation with Rapid Data Limited. The philosophy behind these packages is that they may be used to investigate specific aspects of simple feedback control systems and can be used by a person with no knowledge of the software . Section Three gives a description of the software package developed. Section Four gives some simple examples illustrating the use of the software and this is followed in Section Five by some examples showing more interesting facets of periodic modes that may take place in relay systems.

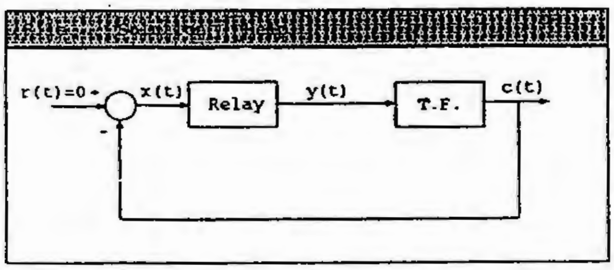


Figure 1 Basic relay system with main menu when the software is invoked

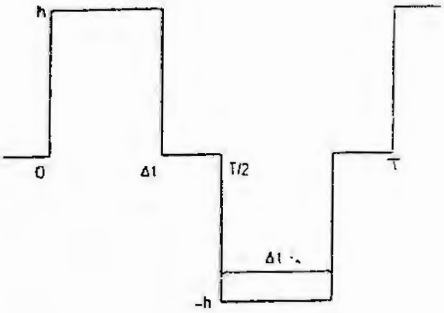


Figure 2: Relay output

2. Outline of the Method

When the relay in Fig. 1 has a dead zone then the basic form of odd symmetric limit cycle will result in the relay output waveform shown in Fig. 2. It is easy to show that the waveform has a Fourier series.

$$y(t) = \frac{2h}{\pi} \sum_{n=1(2)}^{\infty} \frac{1}{n}\left[\sin n\omega\Delta t \cos n\omega t + (1 - \cos n\omega\Delta t)\sin n\omega t\right] \tag{1}$$

This results in the output $c(t)$ having the form

$$c(t) = \frac{2h}{\pi} \sum_{n=1(2)}^{\infty} \frac{g_n}{n}\left[\sin n\omega\Delta t \cos(n\omega t + \varphi_n) + (1 - \cos n\omega\Delta t)\sin(n\omega t + \varphi_n)\right] \tag{2}$$

where $G(jwm) = g_n \exp(j\varphi_n) = U_G(n\omega) + jV_G(n\omega)$ (3)

The A locus of a transfer function [3] is defined by

$$A_G(\theta,\omega) = \operatorname{Re} A_G(\theta,\omega) + j \operatorname{Im} A_G(\theta,\omega) \tag{4}$$

where

$$\operatorname{Re} A_G(\theta,w) = \sum_{n=1(2)}^{\infty} V_G(n\omega)\sin n\theta + V_G(n\omega)\cos n\theta \tag{5}$$

and

$$\operatorname{Im} A_G(\theta,\omega) = \sum_{n=1(2)}^{\infty} (1/n)\left[V_G(n\omega)\cos n\theta - V_G(n\omega)\sin n\theta\right] \tag{6}$$

Using these expressions it is easy to show that

$$c(t) = (2h/\pi)\left[\operatorname{Im} A_G(-\omega t,\omega) - \operatorname{Im} A_G(-\omega t + \omega\Delta t,\omega)\right] \tag{7}$$

and similarly

$$\dot{c}(t) = (2\omega h/\pi)\left[\operatorname{Re} A_G(-\omega t,\omega) - \operatorname{Re} A_G(-\omega t - \omega\Delta t,\omega)\right]. \tag{8}$$

For the autonomous system $-c(t)$ is the input to the relay and for it to generate the assumed output of Fig. 2 it must satisfy

$$\left.\begin{aligned} -c(0) &= \delta + \Delta \\ -\dot{c}(0) &> 0 \end{aligned}\right. \tag{9}$$

and

$$\left.\begin{aligned} -c(\Delta t) &= \delta - \Delta \\ -\dot{c}(\Delta t) &< 0 \end{aligned}\right. \tag{10}$$

at the switching instants, where δ is half the dead zone and Δ half the hysteresis of a relay with dead zone and hysteresis.

Substituting in eqns (9) and (10) the expressions for $c(t)$ and $\dot{c}(t)$ results in the two conditions

$A_G(0,\omega) - A_G(\omega\Delta t,\omega)$ must have

$I.P. = -\pi(\delta + \Delta)/2h$ and $R.P. < 0$ (11)

and

$A_G(0,\omega) - A_G(-\omega\Delta t,\omega)$ must have

$I.P. = -\pi(\delta - \Delta/2h)$ and $R.P. < 0$ (12)

which must be satisfied for a limit cycle to exist.

When the relay has no dead zone, ie δ =0, the single condition

$A_G(0,\omega)$ must have

$I.P. = -\pi\Delta/4h$ and $R.P. < 0$

results.

The A_G locus for any transfer function can be evaluated from the series definitions of eqns (5) and (6) or by the use of z transforms. Once the frequency of the limit cycle when the relay has no dead zone has been found from eqn. (13) or its pulse width and frequency from eqns (11) and (12) the waveform $c(t)$ can be obtained. Since these equations are only necessary conditions for a limit cycle when they have a solution, it is important to obtain the waveform $c(t)$ to see that it satisfies the sufficient conditions for the assumed limit cycle, namely that it does not cross the relay switching levels to cause switchings other than those assumed in giving the output shown in Fig. 2.

A further and interesting feature of the approach is that when an acceptable limit cycle solution is obtained it is possible to do additional calculations to show whether the solution is for a stable or unstable limit cycle.

3. The Software

The program called **Relay** has been written to determine limit cycles in the autonomous feedback system shown in Fig. 1. The plant is linear time invariant and with transfer function $G(s)$, which allows the possibility of a time delay. The relay is symmetric with or without dead zone and hysteresis so that only odd-symmetric limit cycles are considered. When the program is invoked in the MATLAB environment, the block diagram of Fig. 1 is given for the user. Double clicking on the relay block opens up a window which allows the user to enter the form of the relay and its parameters. Similarly, when double clicking on the transfer function, TF, block a window is opened which enables the user to enter the parameters of the numerator and denominator polynomials of the transfer function and any time delay. The **Solution** pull-down menu has two options, namely **Graphical** or **Analytical**. In the graphical approach the user can display A loci relevant to the solution for any possible limit cycle. For the analytical approach the nonlinear equations which give any limit cycle solution are solved using the nonlinear algebraic solution routine available in MATLAB. When a limit cycle solution has been obtained the user can display the limit cycle waveform and also evaluate whether the limit cycle is stable or not. Since the waveform of any predicted limit cycle is available, the user can check its validity by observing that it only crosses the relay switching levels in the appropriate manner for generating the assumed relay output waveform.

4. Some Simple Examples

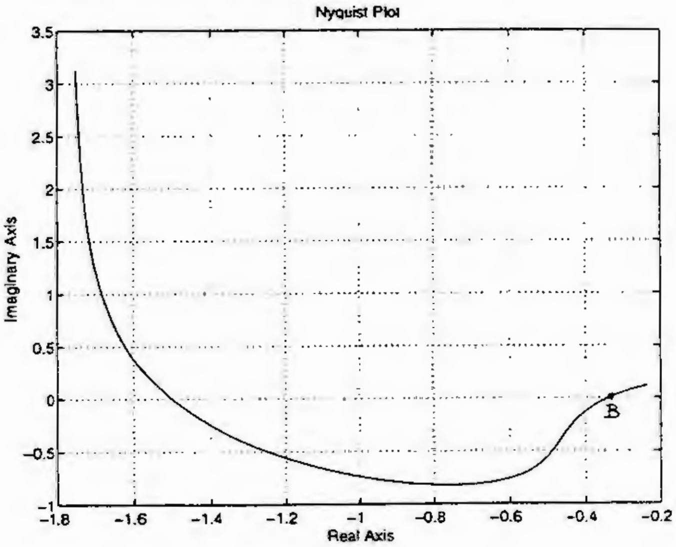


Figure 3: Nyquist plot of $G(j\omega)$ for Example 1

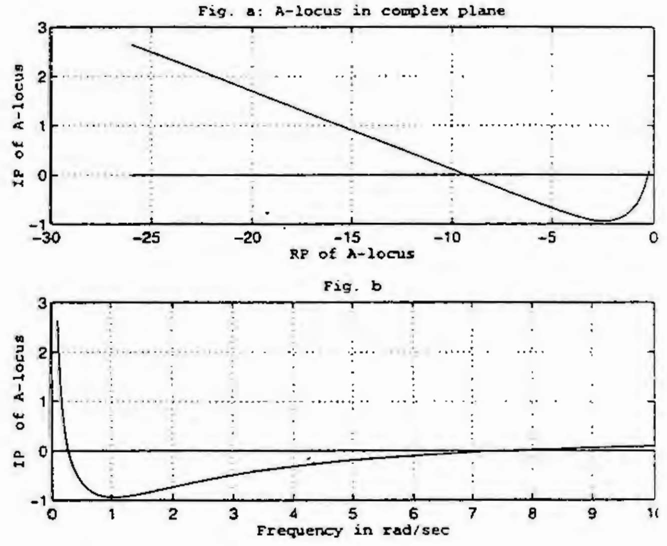


Figure 4: A locus plots for Example 1

Here some straightforward examples are investigated before more complex problems are considered in the next section.

Example 1

Consider a feedback loop with an ideal relay with output ±1 and a transfer function which is both unstable and contains a time delay given by

$$G(s)=\left(\frac{8s^2+10s+2}{3s^3+3s^2-6s}\right)e^{-0.2s}$$

The Nyquist plot of $G(j\omega)$ is shown in Fig. 3 and according to the describing function theory the limit cycle given by point B on the Nyquist locus is a stable one. The calculations show it has an amplitude of 0.43 at a frequency of 7.67 rads/sec. The A locus giving the solution for the limit cycle is shown in Fig. 4 and the nonlinear algebraic equation solver yields the exact limit cycle solution which has the form shown in Fig. 5. It is seen to be almost a triangular waveform with an amplitude of 0.532 and a frequency of 7.62 rads/sec. As expected the accuracy of the describing function method is much better in predicting the frequency of the limit cycle than its amplitude, although it should be noted that the describing function solution is an estimate for the fundamental of the limit cycle waveform. If it was known that the limit cycle was roughly triangular then the describing function approximation for its peak value would be 0.54.

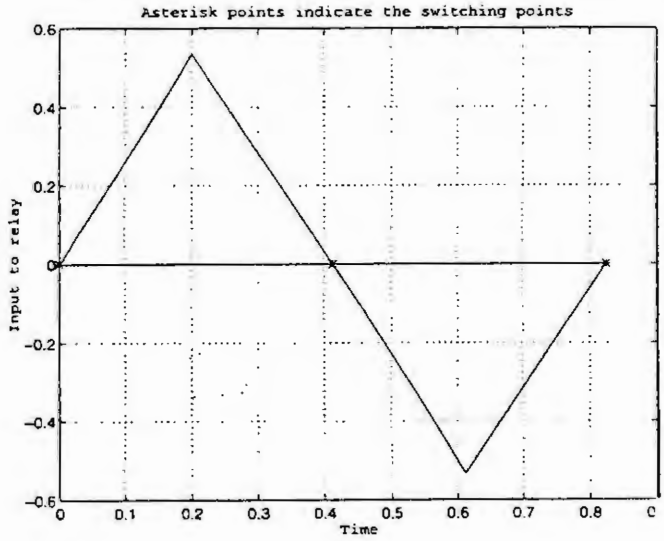


Figure 5: Limit cycle waveform in Example 1

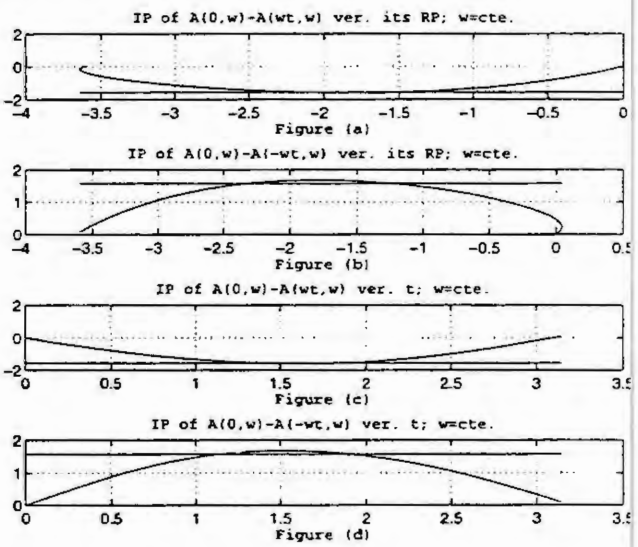


Figure 6: A loci graphs for Example 2

Example 2

Consider a feedback system with a relay having an output ±1 and dead zone ±1 and a transfer function $G(s)$ of $5/s(s^2+3s+1)$. According to the describing function method there are two limit cycles at a frequency of 1.000 rads/sec, one stable and one unstable. Choosing the **Solution** menu in the software and clicking on the submenu **Graphical** a window appears which requests a guess for the frequency of a limit cycle. Entering 1.0 rads/sec and clicking on **Plot** yields the four graphs of Fig. 6. Fig. 6(a) is a plot of the A locus expression in eqn. (11) for the given frequency with $\omega\Delta t$ varying between O and π with the line $-\pi(\delta+\Delta)/2h$ also included. Fig. 6(c) is a plot of the imaginary part of the A locus of eqn. (11) versus Δt for the given frequency. Figs 6 (b) and (d) present similar information for eqn. (12). When the graphs of Fig. 6(c) and (d) intersect the horizontal lines for the same value of Δt a valid solution frequency has been found. This will be seen to be roughly the situation in Fig. 6 for values of Δt around 1.1 and 1.9. When the **Analytical** submenu is chosen initial guesses of (1, 1.9) and (1, 1.1) for the frequency and pulse width of the expected limit cycles may be used. Figs 7 and 8 show the plots of the two limit cycles obtained, the former showing the stable limit cycle which has the larger amplitudes and is more sinusoidal with parameters (0.988, 1.967) and the latter the unstable limit cycle with parameters (0.736, 0.716).

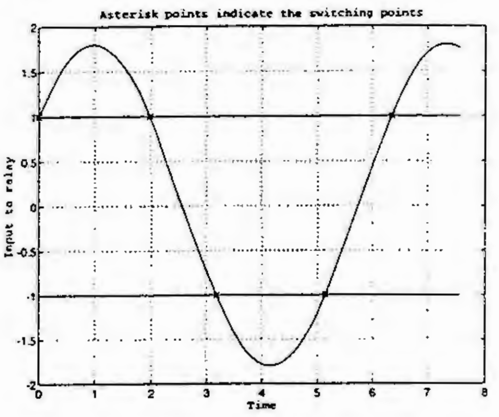


Figure 7: Stable limit cycle solution for Example 2

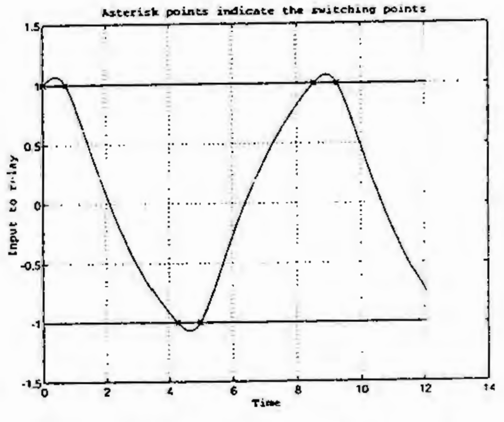


Figure 8: Unstable limit cycle solution for Example 2

5. Multiple Limit Cycle Solutions

In this section a more complicated example is considered which yields several limit cycle solutions. A relay with output ± 1 and hysteresis of ±0.5 is taken and the linear plant has a transfer function

$$G(s)=\frac{100\left(s^2-0.14s+2.2\right)}{s(s+1)^2\left(s^2+1.2s+25\right)}$$

The Nyquist diagram of the plant, $G(j\omega)$, and the negative reciprocal of the relay describing function $-1/N(a)$, are shown in Fig. 9. Two solutions are predicted with frequencies and amplitudes of 5.162 rad/s and 3.400, and 0.816 rad/s and 5.889 respectively. The A locus for the transfer function is shown in Fig. 10 and there are five intersections, numbered 1 to 5, with $-1/N(a)$. Since the intersection at point 4 is for a positive real part only four limit cycle solutions appear to exist and using the analysis feature of the program their frequencies are $\omega_1=0.9541$ rads/s, $\omega_2=1.4223$ rads/s, $\omega_3=1.6762$ rads/s and $\omega_5=5.1634$ rads/s. The solution waveforms for the lowest and highest frequency limit cycles are shown in Fig. 11.

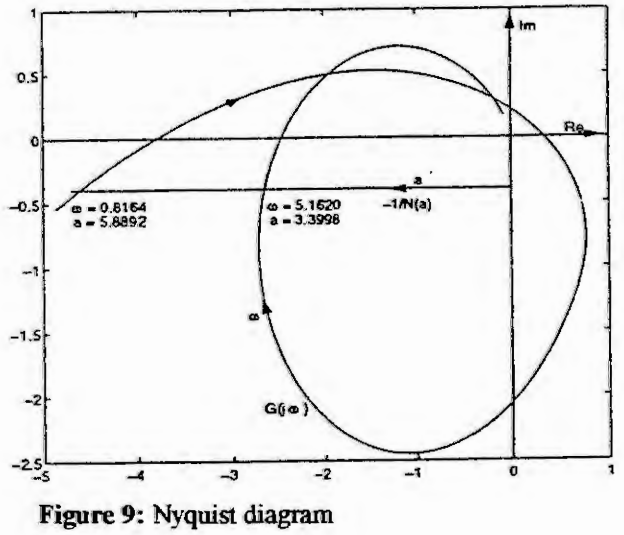


Figure 9: Nyquist diagram

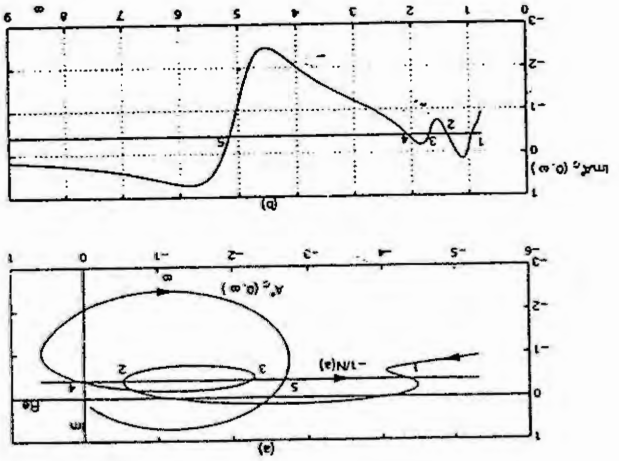
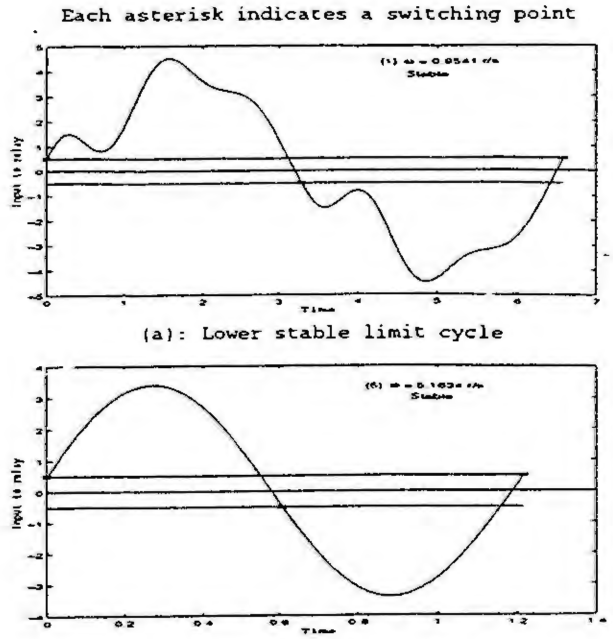


Figure 10: A locus

The analytical solutions for the two remaining solutions 2 and 3 gave the limit cycle waveforms of Fig. 12. It can be seen that these are invalid solutions because they would cause an output with more pulses per period than assumed in the analysis. They indicate the possibility of a limit cycle with 3 pulses per half period but no such stable limit cycle solutions could be found by the software or in simulation.



(a): Lower stable limit cycle

(b)Higher stable limit cycle

Figure 11: Stable limit cycles

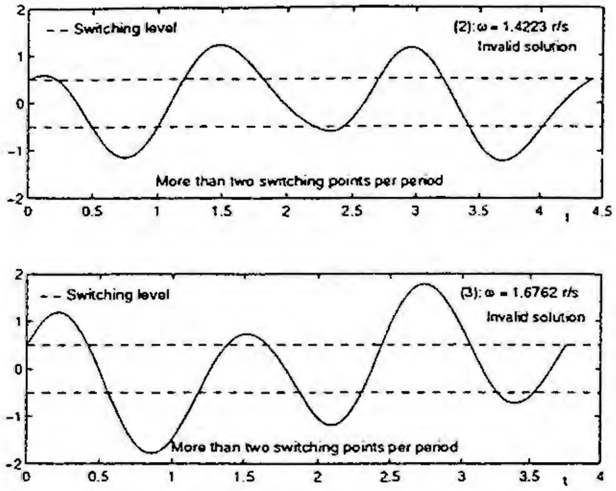


Figure 12 Invalid solutions

Method	ω	Max($x(t)$)	Eigenvalues
A-function	0.9541	4.5650	1, 0.40, 0.17 0.17, 0.00
	5.1634	3.4436	1, 0.98, 0.69 0.69, 0.22
DF	0.8164	5.8892	
	5.1620	3.4998	
Simu-lation	0.9540	4.5520	
	5.1590	3.4382	

Table 1: Multiple stable limit cycles

6 Conclusions

The paper has described software written in MATLAB for the study of limit cycles in relay feedback systems. It is based on the theoretical approach of Tsypkin, has an easy to use graphical interface and requires no knowledge of MATLAB for its use. Several examples have been given to illustrate use of the software.

7 References

[1] K. J. Åström and T. Hägglund, "Automatic Tuning of Simple Regulators", *Proceedings of the 9th IFAC world congress*, Budapest, 1984.

[2] D. P. Atherton, O. B. Sorenson, and A. Goucem, "Teaching Control Engineering Using Implementations of Matlab", *Proceeding IFAC, ACE'94*, Tokyo, Japan, pp. 291-294, 1994.

[3] D. P. Atherton, "Conditions for Periodicity in Control Systems Containing Several Relays", *Paper 28E, 3rd IFAC Congress*, London, 1966.

Modeling of Looper, AGC System and Development of a Dynamic Simulator of Hot Strip Steel Mill

C. J. Park, J. H. Kwak, W. H. Lee and K. T. Lee
POSCO(Pohang Iron & Steel Co. Ltd.) Technical Research Lab.
790-785, Pohang, Kyungbuk, Korea
Fax:82-562-279-6499, E-mail:pc807532@smail.posco.co.kr

Abstract. POSCO has developed a general purpose dynamic simulator of hot strip steel mill. The simulator is a tool for developing the process control system in an industrial system solution business, by making the most use of control and simulation techniques fostered in steelmaking business. This simulator has, not only a powerful numerical analysis function, but an easy-to-use graphic user interface which readily enables to simulate dynamic system. This paper analyzes looper, AGC(Automatic Gauge Control) system and presents the features of the simulator as its application.

1. Introduction

In hot strip steel mill, looper and AGC systems are very important systems to maintain constantly the speed and tension of each stand. The AGC is one of the most advanced control system in the steel industry. But, the AGC and looper can bring about interaction conditions and any problem for the tension between each stand. And it can lead to bad effects at thickness control and strip quality. Thus, it is needed to analyze and model the exact algorithm of these systems.

A computer simulation program that can analyze the dynamic behaviors of hot strip steel mill was developed. It is a tool to present the characteristic of hot rolling process and design the steel plant, control system. With using this simulation program, the stability and accuracy of strip thickness control system were evaluated for various disturbances.[1)-3)]

In this paper, the exact algorithm of looper and AGC system was analyzed and the simulation program and the results of simulation with actual mill data of POSCO #2 hot strip mill(fig.1) were described.

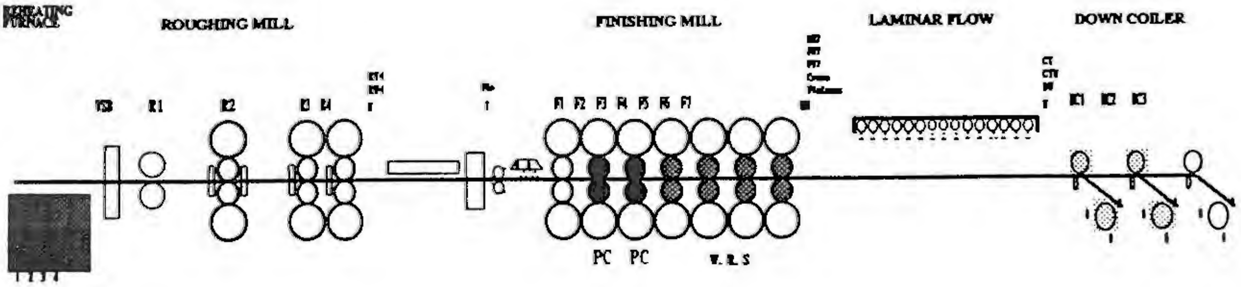


Fig. 1 Layout of POSCO #2 hot strip mill

2. Modeling of looper system

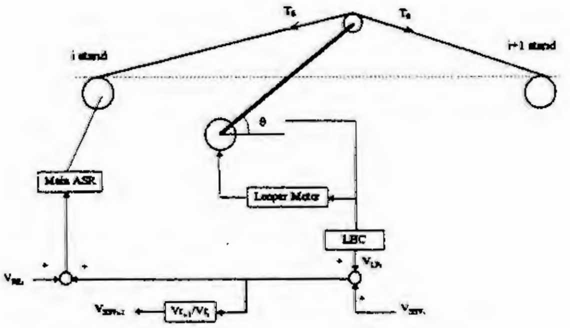
2.1 Outline of looper system

Loopers located between stands in a hot strip finishing mill perform at least two important functions. First, preventing the changes in width and thickness of a strip by regulating the interstand tension at a desired value. Second, preventing the formation of strip loop between stands and providing stable rolling operation by maintaining the looper angle at a desired value. A control input of looper control system is torque of looper arm, work roll speed of former stand and a control output is looper angle, the tension of strip. The angle of looper is controlled by roll speed and the tension is controlled by looper motor's torque. The configuration for the looper system is shown in fig. 2.

2.2 Looper control system

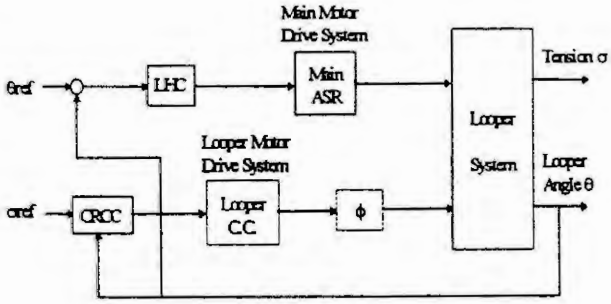
The looper control system has roughly two control modes. The looper motor current reference calculation controller(CRCC) calculates looper motor current reference by input the tension reference from the SCC (Supervisory Computer Control). Then, the current reference is used to the current controller of the looper motor drive system. The another controller is looper height controller(LHC). The LHC calculates the speed correction value of main motor from the looper angle deviation between the reference looper angle and the actual looper angle.

In POSCO #2 hot strip mill case, it performs the looper stable control with looper motor CRCC and LHC. The block diagram for the looper control system is shown in fig. 3.



θ:Looper angle[rad], T_f:Tension[N], Vr:Roll speed[mm/s], V_{RE}:Voltage reference[V], V_{LP}:Voltage compensation by looper LHC[V], V_{SSV}:Sucessive compensation[V]

Fig. 2 Control system of Looper



σ:Unit tension[N/mm^2], LHC:Looper Height Control,
CRCC : Current Reference Calculation Controller,
ϕ:Torque factor, C.C:Current Controller,
Main ASR: Main Motor Drive System

Fig. 3 Looper block diagram of POSCO 2 hot strip mill

3. Modeling of AGC system

POSCO #2 hot strip mill has roll force AGC and monitor AGC etc. at present. Roll force AGC controls exit thickness by using eq. (1) difference between reference roll force set at SCC and actual roll force sensored by the load cell at each stand. The load cell installed in the stand gives measurements of the force transmitted through the mill housing. Next, monitor AGC controls exit thickness by using eq. (2) difference between reference thickness set at SCC and actual thickness sensored by the x-ray at exit stand. To measure the thickness error of the finished product, x-ray gauges are permanently mounted at the exit of the finishing mill. The error signal is used by the monitor gauge control function to distribute the correction to the finishing mill stands. This control is performed in two ways: a high-gain control is used on the initial head part and a lower-gain continuous control is used for the remainder of the coil. This control is an integral feedback properly compensated for transport delay.

$$\Delta s_{RF} = -\frac{\alpha}{M}\Delta P \tag{1}$$

$$\Delta s_{Mon} = -\left(K_P + \frac{K_I}{s}\right)\frac{M + (1-\alpha)Q}{M}\Delta h \tag{2}$$

where, Δs_{RF}: roll gap feedback value of roll force AGC, Δs_{Mon}: roll gap feedback value of monitor AGC, Q: plastic coefficient(=$\partial P/\partial h$), α:control gain, M:mill stiffness, P:roll force, K_p:proportional gain, K_I:integral gain, s:Laplace index, h:exit thickness

4. Development of the simulator and conditions of simulation

We modeled hot strip steel mill and the dynamic simulation program was developed. Fig. 4 illustrates the overall structure of simulation program that was developed by MATLAB with SIMULINK. By using this simulation program, the dynamic characteristic of hot rolling process was analyzed and accuracy of thickness control system was evaluated as a function of disturbances. Table 1 shows rolling conditions used in this simulation. The data were collected from an actual 7 stand hot strip mill of POSCO.

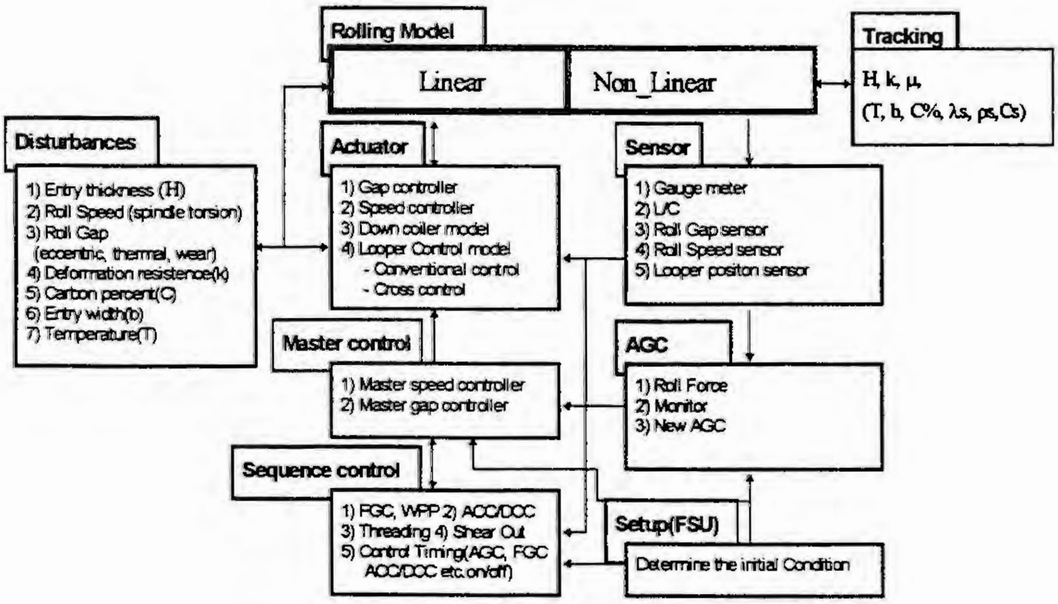


Fig. 4 Structure of dynamic rolling simulation program

Table. 1 Rolling data used in simulation

Item Stand	1	2	3	4	5	6	7
Entry thickness (mm)	19	12.1	8.4	6.3	5.0	4.2	3.7
Exit thickness (mm)	12.1	8.4	6.3	5.0	4.2	3.7	3.3
Strip width (mm)	946	946	946	946	946	946	946
Entry unit tension (N/mm^2)	0.0098	0.7056	1.1074	1.617	2.058	2.6264	3.1948
Exit unit tension (N/mm^2)	0.7056	1.1074	1.617	2.058	2.6264	3.1948	3.6848

5. Results of simulation and discussion

5.1 Dynamic characteristics of hot rolling process

For the simulation, step function of disturbance was used to understand dynamic characteristics of hot rolling process. Fig. 5 shows the thickness deviation for each stand. Thickness deviation of 0.019mm, which was equivalent to 0.1% of base thickness of 19.0mm, was assumed. Thickness deviation value was reduced gradually as rolling proceeded. Fig. 6 is the forward unit tension deviation for each stand. Forward unit tension deviation was decreased to negative direction as the rolling proceeded. Thickness deviation occurred when the starting point of disturbance arrives at relevant stand and increased small amount when the forward unit tension changed.

5.2 Looper simulation

Fig. 7, 8 show the result of looper simulation. Like the fig. 3, Looper controller performs two control methods of CRCC and LHC. Fig. 7 is the simulation result when the proportional gain(K_P) of LHC is 0.001. Where, looper angle presented great overshoot, but rolling was stabilized because the looper angle was converged the constant value. Fig. 8 is simulation result when K_p of LHC is 0.0005. At this case, the overshoot was smaller than previous case.

5.3 AGC simulation

Fig. 9, 10 show the result of AGC simulation by change of AGC gain α. Designed AGC is roll force AGC and Monitor AGC. Where, we suggest the selection method of control gain. Fig. 9 shows simulation result of exit thickness deviation with $\alpha_1=0.6$, $\alpha_2=0.8$ at F1, F2 stand. In this case, thickness deviations had great oscillation and absolute value reduced a little. But, thickness deviations were smaller than fig. 5 without AGC. Fig. 10 shows $\alpha_1=0.8$, $\alpha_2=0.6$. In this case, thickness deviation and oscillation rate were smaller than previous case. Thus it is noted that the control effect was very well if the control gain of the former stand was greater a little than the later stand's.

6. Results

In this paper, a general purpose dynamic simulator was developed to reproduce the hot rolling process and

GMS -- A Graphical-based Modeling System

Ma Yongguang and Wang bingshu
Research Inst. of Simulation & Control Technology
North China Electric Power University , Box 93
204 Qingnian Rd. Baoding, Hebei, P.R.China

Abstract. In the life cycle of modeling and simulation, the most time consuming and difficult job is modeling. The difficulties arise in two major cases: 1)The simulated systems with a large number of interactions and levels of hierarchy are in large scale and wide scope, dynamic and uncertain. 2)For different simulation goals and applications, quite sophisticated and accurate models are required. For example, a diagnostic simulation system will require a model that can accurately replicate the dynamic, static behavior of the real system. At the same time, the effectiveness and validity of simulation highly depend on the performance of the models. Efficient approaches that simplify the modeling process are clearly desirable. Using simulation language, modeling and simulation tools, simulation support environment can eliminate modeling time, improve the efficiency of modeling.

In this paper, a graphic-based modeling software (GMS) is presented. GMS is an integrated modeling system. With its friendly graphical user interface, modeling engineer can represent a system and generate the model auotomatically by drawing schematics using components found in icon class libraries. It provides a graphics-based model developing environment with which the user can generate, test and optimize simulation models of complex systems. It reduces or eliminates the need to modeling engineer in using programming languages and skills in simulation theory. In addition, this paper discusses the design goals, open architecture of GMS, and how to generate and optimize simulation models automatically by using knowledge-base system in the modeling activities which little interaction from modeling engineer is demanded. The features of GMS are given in this paper. The paper concludes with descriptions of current research status of GMS and its development efforts.

keywords: simulation, modeling,object-oriented

1. Introduction

In more and more simulation applications, sophisticate and accurate simulation models are requested. however, modeling is the most time consuming, difficult and labor-intensive endeavor in the lift cycle of simulation and modeling, especially that for complex large scope system, for example, the modeling of a full scope simulation model of a power generation unit. So far as soon, using simulation environment, tools and modeling software kits is the best way to simplify the modeling process, eliminate the modeling time consuming and reduce the labor consuming. The research of modeling software becomes an important branch of simulation studies.

In order to enhance and extend the ability of modeling software, the approach of combining graphical technique intelligent and knowledge engineering with the modeling softwares is focused on for more than a decade. By taking advantages of intelligent user interface formed as nature language interface, interactive dialogue interface or graphical interface, the differences between system presentation by users and the final model presentation may be decreased. And a simplified, straight forward and efficiency user interface makes the user can cope with complex simulation inputs without the need for significant simulation experience and simulation theory. In addition, by embedded the knowledge technique onto simulation software, it is possible that partial endeavor of model generation and optimization can be done automatically by the software, it will benefit the users from standardizing model developing process, reducing errors, shortening test, improving the quality and maintainability of the model software. Combining graphical technique, knowledge-base, object-oriented programming with simulation and modeling technique is the main feature of GMS. It is the major goals of GMS to provide users an integrated modeling, test, model optimization and simulation environment with its friendly graphical user interface and also offers the modeling and simulation life cycle support .

2. Major goals of GMS

It is the general goals of GMS that provide user a general purpose real-time modeling software environment, with which user can create simulation models directly and test them interactively through a friendly graphical user interface, thus greatly enhancing the efficiency of model generation, test and maintenance. In other words, GMS should benefit its user in reducing model software design, development and maintenance costs, improving quality and maintainability of models. Goals in details are as follows.

I) To develop a simple, friendly, convenient and accordant graphical user interface, allow user to represent a system by drawing schematics using components found in icon class libraries. Based on the schematics GMS creates executive models, which can be parameterized, modified and tested interactively. Thus the model development process can be facilitated and standardized.

2) To provide support of whole model development life-cycle includes model generation, test, maintenance, etc. user can perform all of the modeling work in GMS environment.

3) Virtual panel is an utility of GMS. Using the utility outlooks of real panels or the instruments of power plant can be resembled on a screen, all the operation and indication done on the real one can be realized with virtual panel. By using virtual panel, user can easily setup and change the simulation frame, monitor the static and dynamic behaviors of models, which makes model test more efficient.

4) Documentation generating utility generates the documents for every model develop phases, and get hardcopy output when it is request.

5) On-line help is essential, it allows user to get help information interactively at any hierarchy. It guide the user to use the modeling software correctly.

6) Model generating automatization certainly is the best approach to enhance the efficiency of model development. Usually, a simulation model of complex system consists of a large number of model-modules with inputs and outputs, to resolve the cross-reference of these inputs and outputs manually is error prone and time consuming, GMS is designed to do the work automatically.

7) Model optimization is important,and also difficult. Only expert in modeling have the ability to do this work. It is an advanced goal of GMS to realize model automatic optimization by using knowledge-base technique.

8) Component oriented modeling environment allows user generate models directly without the need of editing, compiling any sourse code program thus eliminating the time-consuming associated with these activities. In additional, people who has no significant simulation experience can do the model development work with GMS.

Each of above goals represent an important aspect of GMS. At the same time some common requirements such as open-architecture, compatibility, portability of a software etc. are necessary for GMS.

3. The architecture of GMS

Object-oriented methodology is used in design of GMS, which has the advantage of encapsulation and inheritance. Object-oriented software has the feature of modularity and hierarchical structure. Modularity is accomplished through the encapsulation of procedures and data into software object. Thus the complexity of software is reduced greatly. The architecture of GMS is shown in fig. There are nine main blocks: user interface, icon class generator, graphics editor, documentation generator, virtual panel, knowledge manager, optimizer, model generator and model manager.

1) The user interface is an interactive system which allows user to access the system very easy. The system asks questions to the user in order to get information of a target model that user want to generate. For different users, two kinds of interactive way are considerate. Graphical menu lets new user easy master the software. Expert command line lets skilled user get high input and response speed.

2) The icon class generator generates the icons of components and puts them into icon class libraries. The icons customize the graphical representation of components, they are encapsulated that consist of a data structure and a subroutine defining the structure and dynamic behavior of the component. They can be reuse during later modeling activities.

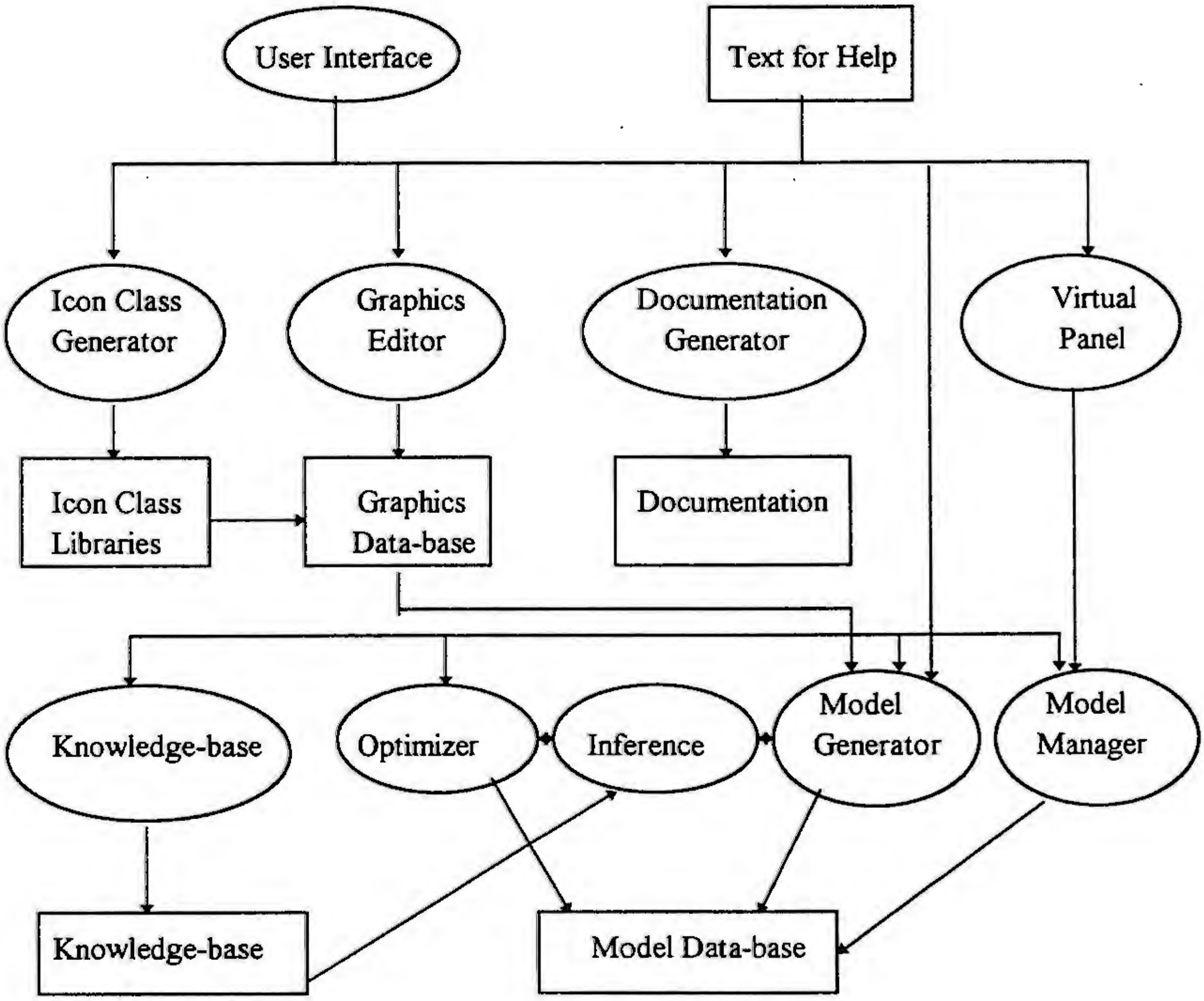


Fig 1. The block diagram of architecture

3) The graphic editor is a tool which supports the model engineers creating schematic drawing of a specific system. Model engineer only needs to select the components found in icon class libraries on drawing and make connections between the components by simply drawing line between them. For a large and complex system, the zoom, pan, and extend of schematic drawing is possible. It also supports the user editing multiple schematics at the same time and inter-schematic browsing. It allows the user making interconnection between the components in difference schematic.

4) The model generator generates executable model automatically through translating the schematics according to the data and topology of the simulated system. The generator makes interconnections of inputs and outputs between model blocks, parameterizes the block coefficients according to the components characteristics, although sometimes, communication with inference may be required to solve some uncertain affects. All of these process are dynamic and invisible.

5) The model manager is a kernel block of GMS. It performs the task of management of executable model. Firstly, modification of model may be quite frequent, resulting from changes in the simulated system. Modularity is an important feature of the simulation model generated by GMS. It allows the user to concentrate on the essential detail of modification. The way of direct modification of simulation model may be necessary and useful. Model manager allows the user directly modify the simulation model through interaction with the user interface. Secondly, a simulation model consist of a large number of model modules that represent components of procedures. These modules may have very different time-constant, to schedule them in different executive period can improve the efficiency of execution. The modules which have not minimal period can have two or more executive phases, to schedule them into reasonable executive phases may balance the processor load. Model manager allows user schedule simulation model through interaction with the user interface. Thirdly, the access to executable model and the executive control of executable model such as go, freeze, fast time, replay etc. are necessary functions of model manager.

6) The virtual panel provides the user a tool by which user can setup the experiment frame and test the model. By resembling the outlooks of real panel and instruments installed, virtual panel on CRT can perform the operation and monitor of simulation model. In GMS, user has the ability to generate required virtual panel and use it.

7) The documentation generator is an aids tool of GMS. It just generates model development documentation that represent the modeling activities of each of step in its life-cycle.

8) The optimizer is an advanced software block of GMS. By comparing the data generated by simulation model with the design data or real-time data of simulated system, the optimizer analysises the errors and validity of simulation model according to the difference between these data. In most cases, the errors are interrelated, difficult to find their reasons, They could be found only by experts who have both knowledge in application domain and experience in simulation. In GMS, knowledge-base is adopted in optimizing the model, the optimizer begins with the result of data analyses, by making use of knowledge found in knowledge-base, starts a inference process, to find the factors which lead to the errors, then optimize the simulation model automatically or give the advises to the user adjusting model parameters.

9) The knowledge-base manager is a handler of knowledge-base. It allows the user to add new knowledge to the knowledge-base or remove obsolete knowledge form knowledge-base. While adding new knowledge to the knowledge-base, it needs to parse the inputs according to the input description format, check if same knowledge already exist, justify if there is any contradiction to other knowledge in the knowledge-base, and translate them into mathematical description that is acceptable by inference, then put it into knowledge-base. To remove and search knowledge are easier than add, but the operating efficiency must be considerate.

ON THE SUPREMAL CONTROLLABLE GRAFCET OF A GIVEN GRAFCET

J. Zaytoon[1], **C. Ndjab**[1] **and J.M. Roussel**[2]
[1] L.A.M., Faculté des Sciences de Reims
Moulin de la Housse, B.P. 1039, F-51687 Reims Cedex 2
[2] L.U.R.P.A., E.N.S. Cachan
61 Av. du Président Wilson, F-94235 Cachan Cedex

Abstract. This paper presents a formal approach for the synthesis of a supremal controllable Grafcet in the frame of the supervisory control theory. The supremal Grafcet represents the minimal possible restriction of the behaviour of a given Grafcet, subject to a number of specifications related to the controlled plant. The different steps of the synthesis approach, which are introduced in this paper, are used to match the semantics of Grafcet with the semantic model of the supervisory control theory.

Introduction

Grafcet [2, 4] is an international standard used for the specification and implementation of logic controllers. The main contribution of Grafcet is that it allows a clear modelling of inputs and outputs, and of their relations. It also allows modelling of concurrency and synchronisation. Many approaches have recently emerged [6, 8, 10] to provide formal verification possibilities to Grafcet. A more challenging problem is that of providing a formal framework for the automatic synthesis of an optimal Grafcet, starting from a given Grafcet and a number of user requirements. Only one approach is documented in the literature to treat this problem [1]. This approach, which has been applied to a real-sized system, can only handle a sub-class of Grafcet in which the logical expressions of Grafcet transitions are limited to single events, representing the edges of input variables. Furthermore, this approach does not consider the constraints induced by the controlled plant.

The supervisory control theory, introduced by Ramadge and Wonham [7], provides a framework for the automatic synthesis of supervisory controllers from their specifications. Despite its theoretical appeal, there are very few control logic synthesis applications based on this theory. This is due to two classes of key problems:

- the interpretation associated to the plant and the supervisor models is not appropriate for most real systems. The logical model proposed assumes a plant that generates events spontaneously and the sole control mechanism available to the supervisor is the ability to prevent the occurrence of some events called controllable events. In fact, real time systems usually react to commands as inputs with responses as outputs.
- automata or/and formal languages proposed by the theory are difficult to manipulate by control system designers. High level specification models (such as Grafcet) are therefore required for practical applications.

This paper presents an approach that is based on the use of Grafcet as a control model in the framework of the supervisory control theory. All the features of Grafcet as defined in [4] are taken into account. The objective of this approach is twofold; i) establishing a scheme for the application of the supervisory control theory for real automated systems, and ii) providing the Grafcet model with a formal support for automatic synthesis of a supremal Grafcet that represents the minimal possible restriction of the behaviour of a given Grafcet and that satisfies the given safety and liveness requirements. The six steps of this approach (Fig. 1) are necessary to match the semantic distance between Grafcet model (based on conditions, events, logic operators, double time scale interpretation, synchronism, reactivity, possibility of simultaneous actions and simultaneous transition firings), and the formal model of the theory which is asynchronous, based on a specific interpretation of events and of controller-plant interaction. The sections of the paper are dedicated, each, to one of these steps, according to the step's order shown in Fig. 1. The reader may wish to refer to [11] for an illustrative example.

Modelling

Grafcet is used to specify the required control behaviour. Plant behaviour as well as the safety and liveness user requirements are modelled using automata. The automaton representing the plant behaviour corresponds to a spontaneous event generator "G" [7]. Controllable events Σ_c are associated to the activation and deactivation of Grafcet outputs. The activation and deactivation of an output z are given, respectively, by the events ↑z and ↓z which belong to Σ_c. Uncontrollable events Σ_u are associated to the rising and falling edges of Grafcet inputs. The rising edge of an input x is given by ↑x, and the falling edge by ↓x.

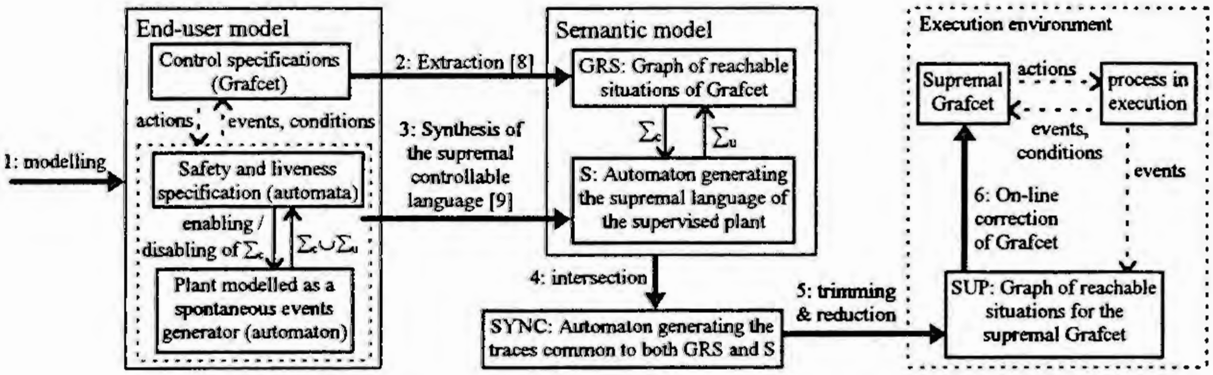

End-user model
Control specifications (Grafcet)
actions
events, conditions
1: modelling
Safety and liveness specification (automata)
enabling / disabling of Σc
Σc∪Σu
Plant modelled as a spontaneous events generator (automaton)
2: Extraction [8]
3: Synthesis of the supremal controllable language [9]
Semantic model
GRS: Graph of reachable situations of Grafcet
Σc
Σu
S: Automaton generating the supremal language of the supervised plant
4: intersection
SYNC: Automaton generating the traces common to both GRS and S
5: trimming & reduction
Execution environment
Supremal Grafcet
actions
process in execution
events, conditions
events
6: On-line correction of Grafcet
SUP: Graph of reachable situations for the supremal Grafcet

Fig 1. Steps for the synthesis of supremal Grafcet

Extraction of the graph of reachable situations of Grafcet

This step consists in generating the graph of reachable situations of Grafcet (GRS) by applying the algorithm given in [8]. This algorithm takes into account the possibilities of interpreted parallelism of Grafcet, the use of edges and step variables in the receptivities of the transitions, and the reachability of a stable situation. The GRS reflects the required semantics of Grafcet in terms of synchrony, reactivity and determinism. From a mathematical point of view, the generated graph is a uniform and completely specified Mealy machine, defined by a 5-tuple: GRS= (X, Z, Y, T, y_0) where:

- X is the set of Grafcet inputs.
- Z is the set of Grafcet outputs.
- Y is the set of states, each of which represents a different reachable situation of Grafcet. To each state y is associated the set $Z_y \subseteq Z$; this set includes the outputs that are active during the state y.
- $T: f(X) \times Y \rightarrow Y$ is a partial function representing the transitions (evolutions) between Grafcet situations; f(X) is a combinatorial expression combining both inputs (binary values) and input edges (events). An event-based algebra has been established [3] to evaluate such an expression.
- y_0 is the initial state.

Synthesis

The supremal language of the supervised plant is next obtained according to the classical synthesis algorithm [9]. The supervisor realisation which is used here is that of a discrete event system S, in which the enabling/disabling action of the supervisor is implicit in the transition structure of S. Therefore, the transition structure of S corresponds to the maximum non blocking allowable behaviour of the controlled plant with respect to the imposed safety and liveness specifications. This means that the part of Grafcet that will be allowed to execute should be confined within the language that can be generated by S. From a mathematical point of view, the generated supervisor is given by a 4-tuple: $S=(\Sigma, Q, \delta, q_0)$. Here Σ is a set of events, Q is the set of states, $\delta: \Sigma \times Q \rightarrow Q$ is a partial function called the transition function, and q_0 is the initial state.

Intersection

The sequences of events which can be generated both by the basic event-model of Grafcet and by the supremal language of the supervisor are extracted by intersection of the two corresponding languages (synchronous composition of the two automata). The synchronised composition of the automata S and GRS yields the automaton SYNC, defined by the 5-tuple (Σ, ST, REG, TR, $state_0$) where:

- $\Sigma = \Sigma_c \cup \Sigma_u$.
- ST is the set of states of SYNC. A state corresponds to a distinct value of the tuple $(q, y, x_1, x_2, \ldots, x_n)$ where q is a state of S, y is a state of GRS, and x_1 to x_n correspond to the logical values (0 or 1) of the inputs of Grafcet.
- REG is a partition of ST into subsets, called regions. Each region includes the states of SYNC which can be visited by partial trajectories composed only of controllable events. To each region $r \subset REG$, will be associated the two following sets :

- NAC(r) is the set of outputs which are forbidden in the region r.
- Z(r) is the set of outputs that are activated by the state of GRS which corresponds to region r.

- TR: $\Sigma \times ST \rightarrow ST$ is a partial function representing the transitions of SYNC.
- $state_0$ is the initial state given by (q_0, y_0, initial value of x_1, initial value of x_2, ... , initial value of x_n).

The automaton SYNC is generated using an algorithm that includes both an initialisation step and an inductive search procedure.

The initialisation step creates the region $r_0 = \{state_0\} \subset REG$, where $Z(r_0)=Z_{y0}$, and $NAC(r_0)=\varnothing$. Starting from $state_0$, the inductive search procedure goes through the automata GRS and S, and applies the following four expressions. In these expressions, the notation $P : p \xrightarrow{\sigma} p'$ is used to express the possibility for a process P to undergo transition from state p to state p' in response to the event σ. Similarly, the notation $P : p \xrightarrow{\sigma} \backslash$ is used to express the fact that when the process P is at state p, no state transition is possible in response to the event σ.

$$S : q \xrightarrow{\uparrow z} q' \quad \Rightarrow$$
$$\begin{cases} ((q,y,x_1,x_2,\ldots,x_n) \xrightarrow{\uparrow z} (q',y,x_1,x_2,\ldots,x_n)) \ \& \ (\text{add } (q',y,x_1,x_2,\ldots,x_n) \text{ to the region of } (q,y,x_1,x_2,\ldots,x_n)), & \text{if } z \in Z_y \\ (q, y, x_1, x_2, \ldots, x_n) \xrightarrow{\uparrow z} \backslash, \quad \text{if } z \notin Z_y \end{cases} \tag{1}$$

$$S : q \xrightarrow{\downarrow z} q' \quad \Rightarrow$$
$$\begin{cases} ((q,y,x_1,x_2,\ldots,x_n) \xrightarrow{\downarrow z} (q',y,x_1,x_2,\ldots,x_n)) \ \& \ (\text{add } (q',y,x_1,x_2,\ldots,x_n) \text{ to the region of } (q,y,x_1,x_2,\ldots,x_n)), & \text{if } z \notin Z_y \\ (q, y, x_1, x_2, \ldots, x_n) \xrightarrow{\downarrow z} \backslash, \quad \text{if } z \in Z_y \end{cases} \tag{2}$$

$$S : q \xrightarrow{\uparrow xi} q' \quad \Rightarrow$$
$$\begin{cases} ((q, y, x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \xrightarrow{\uparrow xi} (q', y', x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)) \ \& \\ \quad (\text{situate } (q',y',x_1,x_2,\ldots,x_{i-1},1,x_{i+1},\ldots,x_n)), \quad \text{if } \exists t: (y \xrightarrow{t} y' \ \& \ f_t(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) = 1) \\ ((q, y, x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \xrightarrow{\uparrow xi} (q', y, x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)) \ \& \\ \quad (\text{situate } (q', y, x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)), \quad \text{otherwise} \end{cases} \tag{3}$$

$$S : q \xrightarrow{\downarrow xi} q' \quad \Rightarrow$$
$$\begin{cases} ((q, y, x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \xrightarrow{\downarrow xi} (q', y', x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)) \ \& \\ \quad (\text{situate } (q',y',x_1,x_2,\ldots,x_{i-1},0,x_{i+1},\ldots,x_n)), \quad \text{if } \exists t: (y \xrightarrow{t} y' \ \& \ f_t(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) = 1) \\ ((q, y, x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \xrightarrow{\downarrow xi} (q', y, x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)) \ \& \\ \quad (\text{situate } (q', y, x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)), \quad \text{otherwise.} \end{cases} \tag{4}$$

The first and the second expressions indicate that a transition corresponding to a controllable event will be included in SYNC if the plant can execute it, and provided that this event is not in contradiction with the output function (of Grafcet) in the current situation. The downstream state of the newly introduced transition of SYNC is added to the region of its upstream state. The first expression reflects the fact that the supervised plant S can execute an event corresponding to the activation of an output of Grafcet only if this output is active in the current situation of GRS. Similarly, the second expression states that process S (and the plant) can execute an event corresponding to the deactivation of an output of Grafcet only if this output is not active in the current situation of GRS. The third and fourth expressions define concurrency in case that an uncontrollable event is generated by the plant. Here, the occurrence of such an event updates the value of the corresponding input variable (of Grafcet); Grafcet situation may also advance concurrently, provided that GRS can execute a transition whose logical condition is true when the event occurs. The occurrence of an uncontrollable event drives SYNC into a new region. The function *situate(state)* is therefore used to determine whether *state* belongs to an already created region; if not, this function creates a new region that includes *state*. For each newly created region, $Z(r)=Z_y$, and $NAC(r)=\varnothing$.

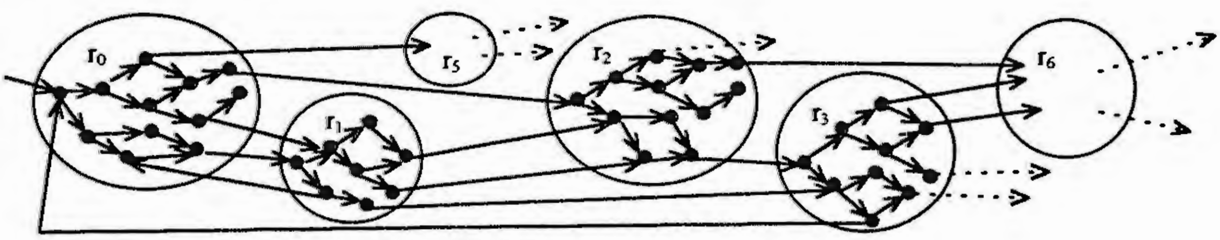


Fig. 2. Semantic model of the composed automaton SYNC

Figure 2 depicts the general form of the semantic model underlying the composed automaton SYNC. The intra-region transitions correspond to controllable events and the inter-region transitions represent uncontrollable events. States with no downstream transitions are blocking states.

Trimming and reduction

The objective of this step is to generate the maximum non-blocking execution model by trimming and reducing the automaton SYNC. The resulting automaton SUP represents the most permissive sub-set of the behaviour of Grafcet which is both non-blocking, and satisfies the supervisory specifications; that is, it gives the graph of reachable situations of the supremal Grafcet. Each state in SUP corresponds to a distinct region of SYNC, and transitions in SUP correspond to the inter-region transitions (changing value of input variable). The grouping of the states of a region in SYNC into a single state in SUP is due to the fact that the behaviour of the automaton SYNC is plant oriented (asynchronous), whereas the behaviour of SUP is Grafcet oriented (synchronous). According to the system point of view of SYNC, controllable events are executed when the plant accepts them; due to the asynchronous nature of the plant and to the delays communication systems, such events cannot take place at the same time instant even though the Grafcet issues the corresponding outputs simultaneously. On the other hand, the automaton SUP is generated to reflect the behaviour of the supremal Grafcet. Therefore this automaton executes the actions from controller point of view which implies that, for a given situation of Grafcet, all parallel outputs are produced simultaneously.

The following two procedures are introduced to be used later on by the trimming and reduction algorithm. The first procedure is invoked when the algorithm is required to remove a region from SYNC, and the second is used for trimming a single transition. The list TRIM contains uncontrollable transitions that have been trimmed in SYNC but haven't yet been guaranteed to be unreachable during the real execution.

```
Procedure Remove-region(r);
Begin
  1) remove the region r from REG, and remove the states and transitions of this region, respectively, from ST
     and TR; /* elimination of the region r and of all the elements inside r */
  2) move upstream transitions of region r from TR to TRIM; /* upstream transitions are memorised to be
     treated later on (to guarantee that they can never be reachable during execution) */
  3) remove downstream transitions of region r from TR; /* these transitions are unreachable during system
     execution */
  4) For each of these removed downstream transitions (of sub-step 3):
        4.1) if the transition is in TRIM, then remove it from TRIM; /* this transition is guaranteed to be
             unreachable, hence it is unnecessary to treat it furthermore */.
        4.2) if the downstream region has no other entering transition then Remove-region(downstream); /*in
             the case where the downstream region (of the removed transition) becomes unreachable, this region
             must be removed by calling the current procedure recursively */;
End.
```

```
Procedure Trim(transition);
Begin
  1) remove transition from TR;
  2) if transition is labelled $\uparrow z$ then  2.1) trim all the transitions of the same region labelled $\uparrow z$;
                                                2.2) add z to NAC(current);
     /* if the trimmed transition corresponds to the activation of an output z, then this output must be forbidden
     everywhere in the current region; this is due to the fact that the order of successive controllable events in the
     region can not be guaranteed in real execution since the Grafcet produces these actions simultaneously.
     Thus the correction of Grafcet will consist in forbidding the execution of this output in the situation
     corresponding to the current region (the region that includes transition). Forbidding outputs in this way
     represents a restriction on the behaviour of Grafcet, and hence it cannot be in contradiction with the
     supervisory specification. */
  3) if transition is labelled $\downarrow z$ then Remove-region(current);
     /* if the trimmed transition corresponds to the deactivation of an output z, then this deactivation, as in the
     case of $\uparrow z$ above, must be avoided in the whole region. However, in the case of $\downarrow z$, we cannot impose the
     activation of the output z, because this corresponds to the introduction of supplementary behaviour that may
     be contradicting with the supervisory specification. Since both cases (activating and deactivating the output)
```

```
    are to be avoided, the whole region is removed (by invoking the previous procedure) to avoid conflicts in
    real execution */
  4) if transition corresponds to an uncontrollable event then
      4.1) put transition in TRIM;
      4.2) Remove-region(downstream) if the downstream region has no other entering transition;
      /* Here, the trimming of an uncontrollable transition consists in memorising it, because further
      treatment is necessary to guarantee that this transition may never execute in the real process. The
      downstream region must be removed if it becomes unreachable due to the trimming of this transition. */
End.
```

The trimming and reduction algorithm is composed of an *initialising step*, a *reduction step* and a final *aggregation step*. The *initialising step* consists in removing the blocking states (states with no downstream transitions) from ST and trimming the upstream transitions of these states. This trimming results in the elimination of certain transitions and regions of SYNC (see procedure Trim(*transition*) above), and therefore certain states become unreachable. The *reduction step* is next used to remove these unreachable states together with their downstream paths and transitions. It is also used to guarantee the non-reachability of the trimmed uncontrollable transitions in the case of real execution. This step is performed for each region in a recursive manner, as follows.

```
Reduction(region)
Begin
  1) invoke the procedure Trim(transition) for each controllable downstream transition of unreachable states of
    the region; /* The trimming procedure (given above) results in removing further transitions and regions */
  2) trim the uncontrollable downstream transitions of unreachable states (in the region) and trim these states.
  3) for each of the transitions trimmed in 2:
      3.1) if this transition is in TRIM, then remove it from TRIM; /* this transition is guaranteed to be
        unreachable, hence it is unnecessary to treat it furthermore */.
      3.2) if this transition leads to an unreachable region (having no other entering transition) then remove
        this downstream region by invoking the procedure "Remove-region";
  4) Select a transition in TRIM which leaves the current region and whose upstream state can not lead
    (through any partial trajectory) to another transition in TRIM which also leaves the current region;
  5) remove the selected transition from TRIM and trim all the transitions leading to its upstream state;
    /* the application of the procedure trim(transition) insures that this state, and the removed transition may
    not be reached in the real process */
  6) if the previous sub-steps (1 to 5) of the current iteration result in trimming new transitions then invoke the
    reduction procedure recursively for the same region in order to purge these transitions;
  7) for states of the current region that have both controllable and uncontrollable downstream transitions,
    remove the uncontrollable downstream transitions;
    /* In this stage, all the trajectories within the current region end with an uncontrollable transition that
    leaves the region. Therefore, all the other uncontrollable transitions leaving intermediate states of the
    trajectory will be eliminated, because they cannot be executed in reality. In fact, Grafcet semantics imposes
    that outputs are produced concurrently and that even if the plant cannot start the execution of these outputs
    simultaneously (because of its asynchronous nature), it will nevertheless receive these outputs consecutively,
    prior to their consequent executions */
  8) for each of the transitions removed in 7:
      8.1) if this transition is in TRIM, then remove it from TRIM; /* it is guaranteed to be unreachable */.
      8.2) if this transition leads to a region with no other entering transition, remove this downstream region;
End.
```

The final *aggregation step* of the trimming and reduction algorithm generates the automaton SUP whose states correspond, each, to a region of SYNC, and whose transitions correspond to the inter-region transitions of SYNC. To each state of SUP is associated the three sets NAC(state), Z(state) and ACT(state) where:

- Z(state) contains the outputs belonging to the set Z of the corresponding region.
- ACT(state) contains the outputs whose activations (↑z) are associated to local transitions of the corresponding region. When the state is active, these outputs must be produced by the Grafcet.
- NAC(state) contains the outputs that must be forbidden in the current state. These outputs consist of those belonging to NAC of the corresponding region, together with those ones whose deactivations (↓z) are associated to local transitions of this region.

On-line correction of Grafcet

The automaton SUP corresponds to the required behaviour of the supremal Grafcet. During execution, this automaton is used to condition the execution of Grafcet outputs, in such a way as to confine the behaviour of Grafcet to the required supremal behaviour. The outputs that must be produced by the supremal Grafcet are given by the set *Current* which is updated during execution. This set is initialised to *ACT(initial state of SUP).* When an event occurs in the plant, the automaton SUP advances in parallel with the Grafcet. In each new state of this automaton, *Current* is calculated by the expression: $Current = (Current \cup ACT(state)) - NAC(state)$. The on-line correction of Grafcet is achieved by deactivating the outputs belonging to *Z(state)-Current.* This allow to avoid behaviours which are not included within SUP. The disabled outputs can be highlighted so that the control designer can distinguish those outputs which are maintained or restricted by the on-line correction.

Conclusion

The work presented in this paper allows to benefit both from the user-friendliness and simplicity of Grafcet, and from recent advances in discrete-event system theory. The main contribution of this work is twofold. First, it associates formal semantics to Grafcet and to its interactions with the controlled plant in the frame of supervisory control theory. Second, it provides an approach for the synthesis of the supremal controllable Grafcet. For an illustrative example, the reader may wish to refer to [11]. Our current research work aims at establishing an algebraic reformulation of this approach so to alleviate the computational difficulties caused by high dimensionality of practical discrete event processes on the one hand, and by the synchronous and parallel nature of Grafcet on the other.

References

1. Charbonnier, F., Alla, H. and David, R., The supervised control of discrete event dynamic systems: a new approach. In: Proc. 34th Conference on Decision and Control, New-Orleans, USA, 1995.

2. David, R., Grafcet : a powerful tool for specification of logic controllers. IEEE Transactions on Control Systems Technnology, 3 (1995), 253-268.

3. Denis, B., Lesage, J. J. and Roussel, J. M., A Boolean algebra for a formal expression of events in logical systems. In: Proc. IMACS-MATHMOD'94 Symposium, Vienna, 1994, 859-862.

4. International Electrotechnical Commission, Preparation of function charts for control systems. Publication 848, Geneva, 1988.

5. Lhoste, P., Faure, J. M., Lesage, J. J. and Zaytoon, J., Comportement temporel du Grafcet. European Journal of Automation, 31 (1997), (to appear, in French).

6. Marcé, L. and Le Parc, P., Defining the semantics of languages for programmable controllers with synchronous processes. Control Engineering Practice, 1 (1993), 79-84.

7. Ramadge, P. J. and Wonham, W. M., The control of discrete-event systems. Proceedings of the IEEE, 77 (1989), 81-97.

8. Roussel, J. M. and Lesage, J. J., Validation and verification of Grafcets using state machine. In: Proc. IMACS CESA'96 Symposium on discrete events & manufacturing systems, Lille, France, 1996, 765-770.

9. Wonham, W. M. and Ramadge, P.J., On the supremal controllable sublanguage of a given language. SIAM J. Control Optimization, 25 (1987), 637-659.

10. Zaytoon, J., De Loor, P. and Villermain-Lecolier, G., Using a real-time framework to verify the properties of Grafcet. In: Proc. 3rd IFAC/IFIP Workshop AARTC'95, Ostend, Belgium, 1995, 233-238.

11. Zaytoon, J., Ndjab, C. and Carré-Ménétrier, V., On the synthesis of Grafcet using the supervisory control theory. In: Proc. IFAC Conference on Control of Industrial Systems, Belfort, France, 1997.

MODELLING OF RIVERS FOR LEVEL CONTROL

J. Chapuis[1] **and R. Sachs**[2]
[1]Swiss Federal Institute of Technology
CH–8092 Zurich
[2]Rittmeyer AG
CH–6302 Zug

Abstract. Water level control is a primary control task of river power plants. In this article we develop two types of linear models for controller design that describe the unsteady flow behavior of open channels near a nominal discharge. The first one is based on a spatial discretization and linearization of Saint-Venant PDE yielding state space models. The second type of models results after transforming the river into an equivalent rectangular channel and solving its linearized PDE. The two types of models are validated with FLORIS[1], a simulation package that solves the nonlinear PDE.

Introduction

To improve the performance of water level control of rivers significantly, more sophisticated controller design methodologies than PID-controllers must be employed. Modern methods like Model-Predictive-Control techniques postulate mathematical models of the process to control. For level control tasks we are interested in which way changes of the discharge affects the water level. Variable river cross-sections and friction lead to a strong nonlinear dynamics that mainly depends on the flow. The level control system has to deal with a wide range of flows from extremely low to extremely high water situations. Because of slowly varying flow changes we consider models that linearizes the nonlinear dynamics near a given operating discharge Q_0. Further we assume that the river section ends with a dam at each side (Fig. 1). The water flow entering and leaving the river is controlled by the upstream and downstream power plant. In Switzerland almost all rivers are surveyed. The known data for the rivers are the cross-section profiles and the stationary water surface profiles for different discharges.

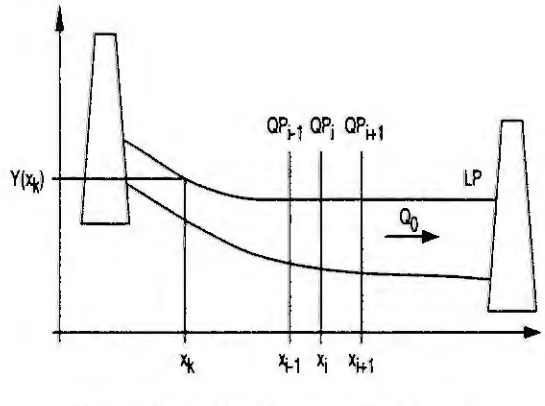


Fig. 1: Considered river configuration.

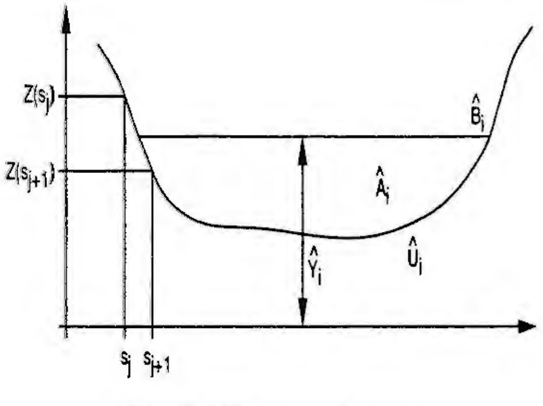


Fig. 2: River cross-section.

A cross-section profile QP_i of the river is a tuple

$$QP_i = (x_i, \Psi_i)$$

containing the longitudinal location x_i and a set Ψ_i, whose elements again are coordinates

$$(s_j, Z(s_j)) \in \Psi_i$$

[1] Developed at the Laboratory for Hydraulics, Hydrology and Glaciology, ETH Zurich, CH–8092 Zurich, http://www.vaw.ethz.ch.

describing the river bed stage $Z(s_j)$ at the cross-section position s_j (Fig. 2). For a stationary discharge Q_0 the longitudinal water stage profile LP is given by a set of discrete points

$$(x_k, Y(x_k)) \in LP$$

The starting point of our models are the PDE of Saint-Venant [1, 2]

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\frac{\partial H}{\partial x} + gA(S_f - S_0) = 0 \tag{2}$$

where Q, H, A, S_f and S_0 are the discharge, the water depth, the cross-section area, the friction slope and the river bed slope. The Saint-Venant equations assumes:

- constant water depth and velocities within a cross-section (one- dimensional flow)
- and a small river bed slope.

Determination of characteristic river parameters

The equations of Saint-Venant uses characteristic quantities like wetted cross-section areas, channel-widths and friction coefficients. These values must be first constructed from initially given geometrical and hydraulic data.

With the cross-section profile data and the corresponding water stage we compute by interpolation the wetted cross-section area $\hat{A}$, the wetted river bed perimeter $\hat{U}$ and the river width $\hat{B}$ at the water surface.

To model the friction slope S_f for turbulent flow condition we are using Strickler's formula [3]:

$$S_f(x) = \frac{V^2}{k_{str}^2 R^{4/3}} \tag{3}$$

where V, $R = A/U$, k_{str} are the water velocity, the hydraulic radius and the Strickler coefficient.

Substituting the depth H with the stage Y and combining (3) with the dynamic equation of Saint-Venant (2) in steady state

$$\frac{\partial}{\partial x}\left(\frac{Q_0^2}{A}\right) + gA\frac{\partial Y}{\partial x} + gAS_f = 0$$

we get the Strickler coefficient

$$\frac{1}{k_{str}^2} = \left(\frac{A}{U}\right)^{4/3}\left(\frac{1}{g}\frac{\partial(\ln A)}{\partial x} - \frac{A^2}{Q_0^2}\frac{\partial Y}{\partial x}\right) \tag{4}$$

In [4] the author used the simplified friction model $S_f = \Gamma V^2$ where the loss coefficient Γ is a time independent constant. This approach leads to badly reproduced filling time constant.

The river can now be parameterized at nominal discharge Q_0 by the quantities $\hat{B}(x)$, $\hat{A}(x)$, $\hat{U}(x)$ and $\hat{k}_{str}$.

Linear state space models

Linearizing (1) and (2) at the operating point $Q = Q_0$, $H = \hat{A}/\hat{B}$ and $U = \hat{U}$, setting $\Delta U = 2\Delta H$ and introducing

$$\epsilon(x) = \frac{5}{3} - \frac{4\hat{H}(x)}{3\hat{U}(x)} \qquad \eta(x) = \frac{Fr^2(x)}{1 - Fr^2(x)} \qquad Fr(x) = \frac{Q_0}{\hat{B}(x)\hat{H}(x)\sqrt{g\hat{H}(x)}}$$

where Fr is the Froude number, we get after scaling of all variables with Q_0, H_0, B_0 and Γ_0

$$\begin{aligned}
\tau &= g\Gamma_0 \frac{Q_0}{B_0 H_0} t & \chi &= g\Gamma_0 x \\
q &= \frac{\Delta Q}{Q_0} & q_0 &= \frac{Q_0}{Q_0} = 1 \\
h &= \frac{\Delta H}{H_0} & \hat{h} &= \frac{\hat{H}}{H_0} \\
\hat{b} &= \frac{\hat{B}}{B_0} & \hat{\gamma} &= \frac{1}{\Gamma_0} \frac{\hat{U}^{4/3}}{k_{Str}^2 (\hat{B}\hat{H})^{4/3}} = \frac{\hat{\Gamma}}{\Gamma_0}
\end{aligned}$$

the following partial differential equations

$$\begin{bmatrix} \frac{\partial h}{\partial \tau} \\ \\ \frac{\partial q}{\partial \tau} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{\hat{b}} & 0 & 0 \\ \\ -\frac{1}{\hat{b}\hat{h}^2 \eta} & -\frac{2}{\hat{b}\hat{h}} & \frac{2\hat{\gamma}\epsilon}{\hat{b}\hat{h}^2} & -\frac{2\hat{\gamma}}{\hat{b}\hat{h}} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial h}{\partial \chi} \\ \frac{\partial q}{\partial \chi} \\ h \\ q \end{bmatrix} \tag{5}$$

Further (5) can be discretized in $2n$ equidistant sections each of length $l = g\Gamma_0 L_c / n$ where L_c is the total river length (Fig. 3). The discrete quotients can be chosen in such a way that the state vector $\chi = [h_1 \; q_1 \; h_2 \; \ldots \; h_{2n-1} \; q_{2n} \; h_{2n+1}]^T$ describes alternatively discharge and depth changes at the section boundaries. We obtain a linear state space model of order $2n + 1$. For the time derivatives of the depths h_i $(i = 1(2)2n + 1)$ and discharges q_{i+1} $(i = 1(2)2n - 1)$ we get

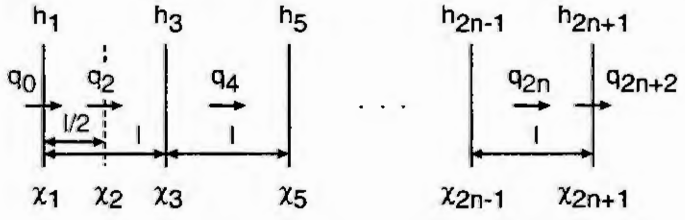


Fig. 3: Spatial discretization of river.

$$\frac{dh_i}{d\tau} = \alpha_{1,i}(q_{i-1} - q_{i+1}) \quad \text{and} \quad \frac{dq_{i+1}}{d\tau} = \alpha_{2,i+1} q_{i-1} + \alpha_{3,i+1} h_i + \alpha_{4,i+1} q_{i+1} + \alpha_{5,i+1} h_{i+2}$$

where

$$\begin{aligned}
\alpha_{1,i} &= \frac{1}{\hat{b}_i l} 2^{\delta_{1,i}} 2^{\delta_{2n+1,i}} \\
\alpha_{2,i+1} &= \frac{2}{\hat{b}_{i+1} \hat{h}_{i+1} l} 2^{\delta_{2,i}} \\
\alpha_{3,i+1} &= \frac{1}{\hat{b}_{i+1} \hat{h}_{i+1}^2} \left(\frac{1}{\eta_{i+1} l} + \hat{\gamma}_{i+1} \epsilon_{i+1} \right) \\
\alpha_{4,i+1} &= -\frac{2}{\hat{b}_{i+1} \hat{h}_{i+1}} \left(\frac{1}{l} + \hat{\gamma}_{i+1} \right) \\
\alpha_{5,i+1} &= -\frac{1}{\hat{b}_{i+1} \hat{h}_{i+1}^2} \left(\frac{1}{\eta_{i+1} l} - \hat{\gamma}_{i+1} \epsilon_{i+1} \right)
\end{aligned}$$

The cases for $i = 1, 2, 2n + 1$ due to the special location of the discharges q_0 and q_{2n+2} are handled with the Kronecker-Delta symbol $\delta_{k,j}$.

Transformation to equivalent channel

If one is only interested in the hydro-dynamical behavior at a specific location x_{eq} along the longitudinal river axis, the river section can be transformed approximately into a channel with constant parameters [4]. This equivalent channel is an ideal rectangular channel whose dynamics for small discharge deviations can be described by the nominal parameters Q_0, A_0, B_0, Γ_0 and L_c. To transform a given river section in its equivalent channel the following conditions must be fulfilled:

$$B_0 L = \int_{-L_c}^{0} \hat{B}(x)dx \tag{6}$$

$$\frac{L_c}{\sqrt{g\frac{A_0}{B_0}} + \frac{Q_0}{A_0}} + \frac{L_c}{\sqrt{g\frac{A_0}{B_0}} - \frac{Q_0}{A_0}} = \int_{-L_c}^{0} \frac{dx}{\sqrt{g\frac{\hat{A}(x)}{\hat{B}(x)}} + \frac{Q_0}{\hat{A}(x)}} + \int_{-L_c}^{0} \frac{dx}{\sqrt{g\frac{\hat{A}(x)}{\hat{B}(x)}} - \frac{Q_0}{\hat{A}(x)}} \tag{7}$$

Equation (6) states that in both cases the area of the water surface must be the same. This guarantees equal filling time constants. In (7) the sums of propagation times of upstream and downstream traveling waves must be equal. Equations (6) and (7) can be solved for A_0 and B_0. The loss coefficient Γ_0 is calculated assuming an equal mean friction slope over the whole channel length

$$\frac{\Gamma_0}{A_0^2} = \frac{1}{L_c} \int_{-L_c}^{0} \frac{\hat{\Gamma}(x)}{\hat{A}^2(x)} dx$$

Solving Saint-Venant equations for the equivalent channel

Because of the uniform geometrical structure of the equivalent channel, (5) reduces to

$$\begin{bmatrix} \frac{\partial h}{\partial \tau} \\ \\ \frac{\partial q}{\partial \tau} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \\ -\frac{1}{\eta} & -2 & 2\epsilon & -2 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial h}{\partial \chi} \\ \frac{\partial q}{\partial \chi} \\ h \\ q \end{bmatrix} \tag{8}$$

with

$$\epsilon = \frac{6H_0 + 5B_0}{6H_0 + 3B_0} \approx 1.67 \; (B_0 \gg H_0)$$

and where we set $\hat{B} = B_0$, $\hat{H} = A_0/B_0$, $\hat{\Gamma} = \Gamma_0$. After transformation of (8) in the frequency domain we can solve the resulting linear differential equations with various boundary conditions and obtain transcendental transfer functions:

1. Transfer function from a inflow change to the depth change at location $\chi \in [-l_c, 0]$ with the boundary condition $q(0, s) = 0$:

$$G_{in}(s) = \frac{h(\chi, s)}{q(-l_c, s)} = e^{\alpha(\chi + l_c)} \frac{\beta \cosh(\beta\chi) + \alpha \sinh(\beta\chi)}{s \sinh(\beta l_c)} \tag{9}$$

2. Transfer function from an outflow change to the depth change at location $\chi \in [-l_c, 0]$ with the boundary condition $q(-l_c, s) = 0$:

$$G_{out}(s) = \frac{h(\chi, s)}{q(0, s)} = e^{\alpha\chi} \frac{\alpha \sinh(\beta(\chi + l_c)) + \beta \cosh(\beta(\chi + l_c))}{s \sinh(\beta l_c)} \tag{10}$$

with

$$\begin{aligned} \alpha(s) &= \eta(s + \epsilon) \\ \beta(s) &= \sqrt{\eta(\eta + 1)s^2 + 2\eta(\eta\epsilon + 1)s + \eta^2\epsilon^2} \end{aligned}$$

Usually the water level is controlled near the downstream dam. Assuming as a special case $\chi = 0$ (channel exit) we can develop power series of (9) and (10) by evaluating the residuals at the complex poles $\pm k\pi j$ for $k \geq 0$:

$$G_{in}(s) = e^{l_c\eta(s-\epsilon)} \left(\frac{\eta\epsilon}{s \cdot \sinh l_c\eta\epsilon} + \sum_{k=1}^{\infty} \frac{k^2\pi^2}{k^2\pi^2 + l_c^2\eta^2\epsilon^2} \cdot \frac{(-1)^k 2l_c^{-1}\left(s + 2\frac{\eta\epsilon+1}{\eta+1}\right)}{s^2 + 2\frac{\eta\epsilon+1}{\eta+1}s + \frac{k^2\pi^2+l_c^2\eta^2\epsilon^2}{l_c^2\eta(\eta+1)}} \right) \tag{11}$$

$$G_{out}(s) = -\eta - \frac{(1+\coth l_c\eta\epsilon)\eta\epsilon}{s} - \sum_{k=1}^{\infty} \frac{k^2\pi^2}{k^2\pi^2 + l_c^2\eta^2\epsilon^2} \cdot \frac{2l_c^{-1}\left(s + 2\frac{\eta\epsilon+1}{\eta+1}\right)}{s^2 + 2\frac{\eta\epsilon+1}{\eta+1}s + \frac{k^2\pi^2+l_c^2\eta^2\epsilon^2}{l_c^2\eta(\eta+1)}} \tag{12}$$

Both transfer functions show integral behavior and an infinite number of oscillatory modes. The time delay of (11) corresponds to the wave propagation time from the channel entry to its exit. The direct feed through term in (12) indicates that a discharge change at downstream causes an immediate change of the water depth.

Model reduction

The following calculations are based on real data of the river Rhein between the two power plants at Augst-Wylen and Birsfelden. Evaluation of the geometrical and hydraulic data for three different flow conditions (extremely low to extremely high water situation) leads to the model parameters as shown in Tab. 1 and 2.

Q_0 $[m^3/s]$	270	1030	5500
L_c $[m]$	7889	7889	7889
B_0 $[m]$	213.70	216.38	222.72
A_0 $[m^2]$	1262	1295	1753
H_0 $[m]$	5.91	5.98	7.87
Γ_0 $[s^2/m^2]$	1.7912e-4	1.1612e-4	0.6346e-4

Tab. 1: River parameters.

Q_0 $[m^3/s]$	270	1030	5500
l_c	13.86	8.99	4.91
Fr	0.03	0.10	0.36
η	0.0008	0.0109	0.1461
ϵ	1.6318	1.6317	1.6227

Tab. 2: Derived channel constants.

For controller design purposes the order of (11) and (12) must be reduced, e.g. by cutting off high frequency poles. Evaluation of the sums for $k = 1 \ldots n$ results in rational transfer functions $G_{Zu,n}(s)$ and $G_{Ab,n}(s)$ of order $2n + 1$. Fig. 4 shows the frequency spectra of G_{in} and $G_{Zu,8}$ for three different discharges.

Simulation results

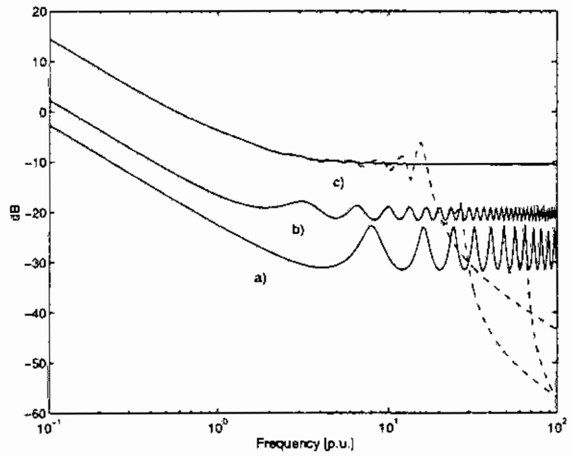
In the following we compare step responses of the obtained state space models as well as the transfer functions with the simulation package FLORIS. FLORIS determines the unsteady one-dimensional flow of river networks by solving the nonlinear PDE of Saint-Venant. It uses as input data the cross-section profiles and the Strickler coefficients. The profiles can be subsequently recalculated with FLORIS and show a good correspondence with the initially given data (Fig. 5).

For the simulation we assume $Q_0 = 1030\,m^3/s$ and a 10% discharge step at the upstream boundary for $t = 0$. Fig. 6 and Fig. 7 shows the step responses of the depth at the downstream boundary compared to the solution of FLORIS. The chosen orders for both types of model are 7 and 21.

The transfer function models in Fig. 6 have pronounced oscillatory behavior because of neglecting high frequency poles. Increasing the model order doesn't improve accuracy. The state space models in Fig. 7 reproduces very well the initial time delay but need a very high order for good approximations. This is due to the sampling theorem leading to aliasing effects for low orders.

Conclusions

Starting from given geometrical and hydraulic river data we developed linear models for controller design purposes. The achieved accuracy depends on the chosen model order. Due to the strong process nonlinearity the models are valid only near a nominal discharge. The models reproduce very well the phenomenon of translatory wave propagation.



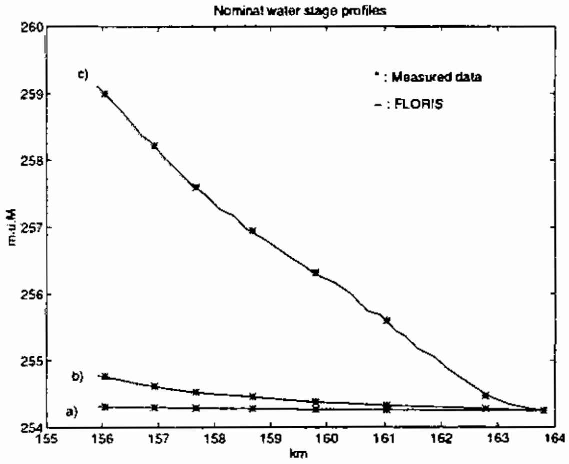
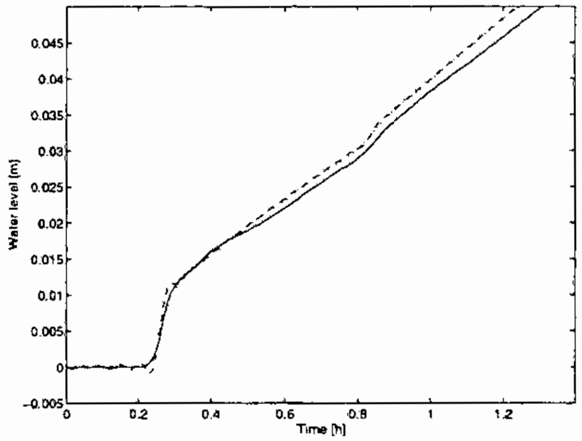


Fig. 4: —— $|G_{in}(j\omega)|$; - - - $|G_{in,8}(j\omega)|$
a) $Q_0 = 270\,m^3/s$
b) $Q_0 = 1030\,m^3/s$
c) $Q_0 = 5500\,m^3/s$

Fig. 5: Measured and computed water stage profiles
a) $Q_0 = 270\,m^3/s$
b) $Q_0 = 1030\,m^3/s$
c) $Q_0 = 5500\,m^3/s$



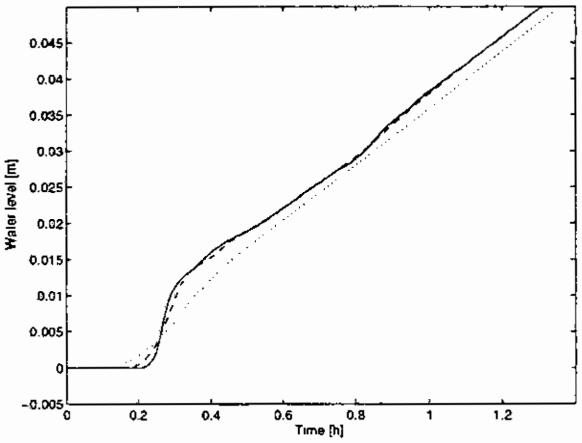


Fig. 6: Step responses of transfer function models for $Q_0 = 1030\,m^3/s$ and $\Delta Q = 103\,m^3$:
—— FLORIS; . . . $n = 3$; - - - $n = 10$

Fig. 7: Step responses of state space models for $Q_0 = 1030\,m^3/s$ and $\Delta Q = 103\,m^3$:
—— FLORIS; . . . $n = 3$; - - - $n = 10$

References

[1] J.A. Cunge, F.M. Holly, and A. Verwey, *Practical Aspects of Computational River Hydraulics*, Pitman, 1980.

[2] A. Chadwick and J. Morfett, *Hydraulics in Civil and Environmental Engineering*, Chapman & Hall, London, 1993.

[3] *FLORIS — Benutzerhandbuch, Version 2.0.3*, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie, ETH Zürich, 1992.

[4] M. Lahlou, *Modelisation des canaux hydrauliques et application au réglage de niveau*, thése. EPF Lausanne, institut d'électronique industrielle, 1994.

MATHEMATICAL MODELING OF ELECTRICALLY STIMULATED MUSCLE

J. Schultheiss, L. del Re and F. Kraus
Automatic Control Laboratory Zürich
Physikstr. 3, CH-8092 Zürich

Abstract: Electrically stimulated muscle is usually modeled using Hammerstein type models, mainly because of the easy later controller design. However, these models suffer from a limitation: they do not include any hysteresis effects which have been reported by several researchers. In this paper we present a possibility to include hysteresis in such models. Although significant increase in model accuracy has been achieved, Hammerstein type models remain inadequate to completely describe the behavior of electrically stimulated muscle.

Introduction

Functional electrical stimulation (FES) is a technique which aims to restore an ability to handicapped persons. It does so by generating nerve signals in places where the natural stimulus is missing due to malfunctions of the nervous system. Today electrical nerve and muscle stimulation has a broad field of application which includes cardiac pacemaker, phrenic pacemakers against respiratory insufficiency, motor nerve stimulation for the paralyzed, electrical stimulation for urinary or anal incontinence, improvement of spasticity, visual prostheses for the blind, auditory prostheses for the deaf, etc. In all these cases electrical stimulation is used to compensate for inadequately functional, missing or lost body functions and hence this technique is called *functional electrical stimulation.*

Our group has focused on gait restoration for the paralyzed. In recent years applied researchers have become more and more interested in using feedback systems in order to cope with the large inter-subject variability in response to electrical stimulation. As with any closed loop system the achievable performance and/or the robustness merely depends on the accuracy of the model. So far force dynamics of electrically stimulated muscle have been described by so called Hammerstein type models [2, 5, 10, 6, 4]. However, these models were unable to account for any hysteresis effects which were reported by several investigators [1, 7, 8].

The aim of this paper is to present a model which is suitable for the latter control and can easily be adopted for the individual patient. Hence, we continue with the structure of the Hammerstein type model and extend it to include hysteresis effects. In the following section we describe the experimental setup. Next we will describe how we extended the classical Hammerstein type model to account for hysteresis effects. Eventually, simulation results will be presented and compared to experimental data.

Experimental setup

All measurements were recorded on healthy able bodied subjects. During the experiment the subject was sitting on a table so that the shank could swing freely. The knee angle was measured with a goniometer which was attached to the shank as well as to the thigh (see Figure 1). The data was recorded at a sampling frequency of 320 Hz with 16 times oversampling yielding an effective sampling frequency of

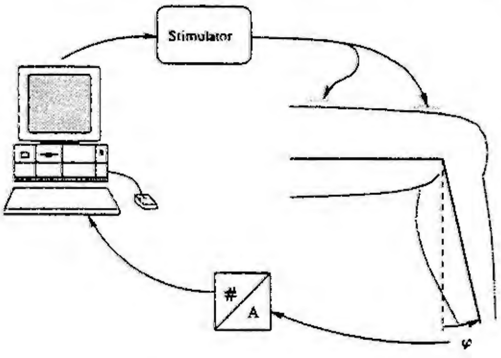


Figure 1: Experimental setup

20 Hz resp. 50 ms sampling time. Prior to the digital acquiring the data was filtered with an analog 4th order butterworth filter with a cut-off frequency of 32 Hz. The electrical stimulus was generated using a device developed by the group of Prochazka and was delivered through self adhesive surface electrodes. To vary the intensity of the applied electrical stimulus we varied only the pulse width. Pulse frequency was double the sampling frequency (40 Hz) and the pulse amplitude was set to 50 mA.

Before each experiment a "normalization" procedure was carried out. Normalization of input as well as of output signals was necessary in order to obtain reproducible shapes of the input nonlinearities as the absolute pulse widths necessary to stretch the leg may vary substantially.

First, the resting angle was measured and subsequently subtracted from every measurement. Hence, the resting position was set to zero. After the knee angle offset has been determined, the subject was asked to voluntarily stretch the leg. In the case of patients this task could be accomplished with the help of any aids. The subsequent measurements were scaled so that the knee angle between resting position and fully stretched leg lied in the interval [0, 1]. Note, that during swing back of the shank, knee angles may become negative as well.

When the shank was back in the resting position electrical stimulation was switched on and continually increased until the knee angle was 80% of the maximal knee angle determined before. The normalization procedure was done at around 80% of the maximal angle in order to calibrate the system in an operational range.

Modeling of electrically stimulated muscle

Basically the behavior of electrically stimulated muscle can be described as a pendulum which is driven by a momentum caused by the force generation of the stimulated muscle. Obviously, the most difficult problem is the modeling of the force generation. This is due to

- Different electrode positions on the skin and hence stimulation of different muscle fibers. This effect can occur even during a session because the electrodes which are placed on contracting muscles can move slightly and hence tbe electrical field is focused on different nerves.
- Varying skin resistance to electrical current. Hence the same amount of current today will produce a different force compared to yesterday.
- Fatigue effects. Force generation will decrease with time.

We therefore concentrate first on the modeling of movements where no force (except gravity) acts on the lower leg.

Dynamics of passive swing back

To determine the dynamics of passive swing back three different measurements have been compared to each other. Figure 2 shows the three different step responses as they have been conducted, *i.e. not scaled* whereas in Figure 3 the data is *scaled* such that the maximum of each measurement (occurs

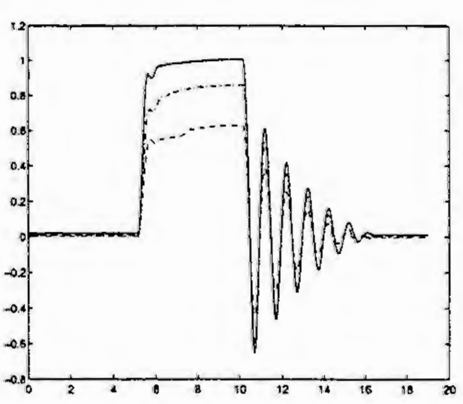


Figure 2: Step responses *not* scaled

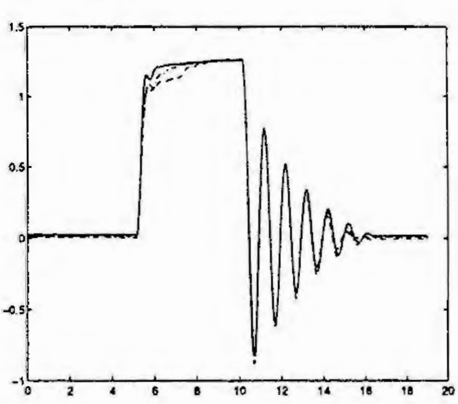


Figure 3: Step responses scaled

after 10 seconds) corresponds to each other. From Figure 3 one can conclude that the passive lower leg dynamics can be modeled as a linear system. With the data shown in Figure 2 we can determine a third order autoregressive (AR) model to estimate the passive dynamics of the shank. In Figure 4 a pure simulation is shown which has been accomplished by setting the first three initial values of the simulation equal to the measured data.

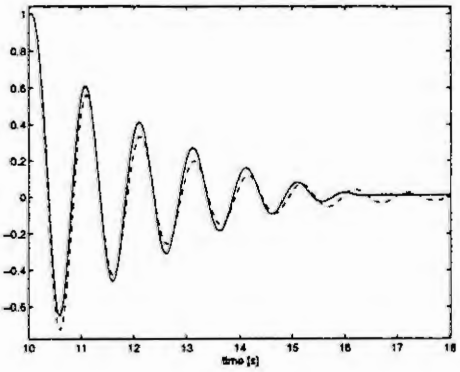


Figure 4: Simulation of passive swing with a 3rd order AR model

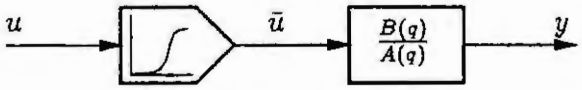


Figure 5: Hammerstein type model structure

It is therefore not surprising that in the literature, the most commonly used model structure is the so called Hammerstein type model [2, 5, 10, 6, 4]. Hammerstein type models consist of a static input nonlinearity followed by a linear transfer function (see Figure 5). These models allow a simple linear controller design since the effect of the static nonlinearity is generally compensated by premultiplication of the control signal with the inverse of the estimated nonlinearity. The controller is then designed under the assumption of a linear input-output relationship.

However, these models suffer from several limitations when applied to FES. Most significantly, with the standard static input nonlinearity it is impossible to account for any hysteresis.

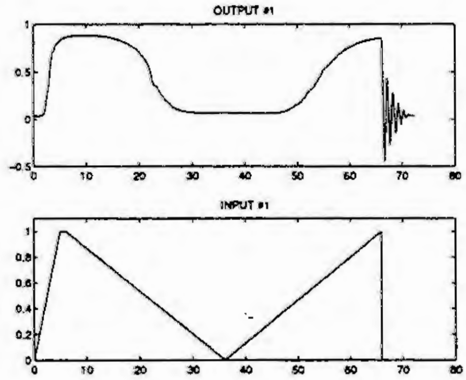


Figure 6: Ramp input

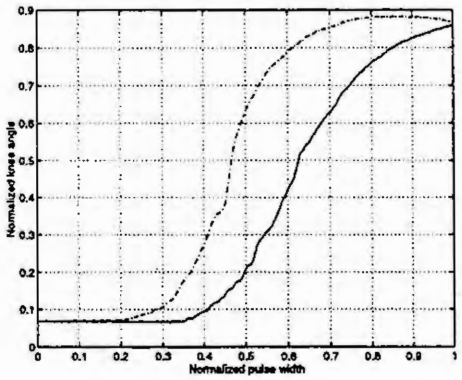


Figure 7: Standard hysteresis representation

In Figure 6 we see the result of an experiment in which the stimulus intensity has slowly been decreased and later increased. One can clearly see the nonlinear behavior of electrically stimulated muscle. The general form of the input nonlinearity is similar to the recruitment characteristics shown in [5]. Furthermore, one can see the hysteresis when increasing (solid) and decreasing (dashed) stimulus intensity (Fig. 7). Such hysteresis has also been noted by other investigators [8, 7, 1].

Nonlinear identification using neural networks

A general and well known approach to nonlinear identification is the use of neural networks. Neural networks are known to be particularly suitable for finding nonlinear mappings. A common pitfall however is to over-parameterize the network. Another drawback in neural networks is that the resulting weights are often difficult to interpret. In [3] a special neural network structure has been suggested which evades these problems and can be adopted very well to the problem here.

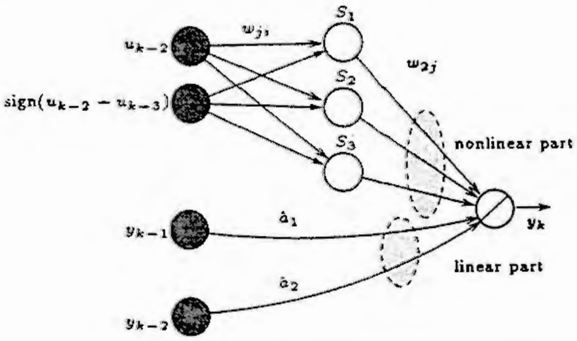


Figure 8: Simple neural network structure

In order to obtain interpretable results a rather simple network structure has been chosen. The structure of the network in Figure 8 is of similar complexity as the Hammerstein type model commonly used in literature. Note that the output function of the last neuron is a linear transfer function.

The one-step ahead prediction y_k of the neural network (Fig. 8) can be written as

$$y_k = \hat{b}_0 \, f(u_{k-2}, \text{sign}(u_{k-2} - u_{k-3})) + \hat{a}_1 \, y_{k-1} + \hat{a}_2 \, y_{k-2} \tag{1}$$

where the nonlinear exogenous input is approximated by the "nonlinear" subpart of the neural network shown in Figure 8. This subpart of the neural network can be computed according to the classic theory of neural networks (in this figure the number of neurons is $N_1 = 2$ and $N_2 = 3$).

$$\begin{aligned} f(u_{k-2}, \text{sign}(u_{k-2} - u_{k-3})) &= f_T(w_{20} + \sum_{j=1}^{N_2} w_{2j} \cdot S_j) \\ S_j &= f_T(w_{j0} + \sum_{i=1}^{N_1} w_{ji} \cdot I_i) \qquad j \in [1..3] \\ I_1 &= u_{k-2} \\ I_2 &= \text{sign}(u_{k-2} - u_{k-3}) \end{aligned}$$

The function $f_T(x)$ in this context is usually called *transfer function* and is defined as

$$f_T(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2}$$

The hyperbolic tangent sigmoid transfer function (2) has been chosen as either input as well as output data lies in the interval [-1, 1]. As learning technique an error back-propagation method was used, the so-called Levenberg-Marquardt algorithm [9, p. 284 ff]. This algorithm shows a superior learning efficiency when compared to standard error back-propagation techniques.

The direct estimation of the input nonlinearity together with the linear dynamics is difficult since the dynamics are weighted with a much higher weight in the learning procedure. Therefore the nonlinear part f is estimated separately. Regrouping (1) we obtain

$$\underbrace{y_k - \hat{a}_1 y_{k-1} - \hat{a}_2 y_{k-2}}_{v_k} = \hat{b}_0 \, f(u_{k-2}, \text{sign}(u_{k-2} - u_{k-3})) \tag{3}$$

Defining $v_k = \frac{1}{\hat{b}_0}(y_k - \hat{a}_1 y_{k-1} - \hat{a}_2 y_{k-2})$ we can see f as a mapping

$$f : \mathbf{x} \in \Re^2 \longrightarrow \Re : [u_{k-2}, \text{sign}(u_{k-2} - u_{k-3})] \xrightarrow{f} v_k$$

The neural network has the two inputs $[u_{k-2}, \text{sign}(u_{k-2} - u_{k-3})]$ and the target v_k to estimate this mapping. The time series v_k can be obtained by filtering the measured output y_k with the dynamics of the passive swing back.

Results

Let us now apply the methodology described above to the problem of the electrically stimulated muscle. First of all, we should think of appropriate input signals to persistently excite the system. As we know from theory, a good input signal would be white noise or a pseudo random binary signal (PRBS). However, with white noise input signals slower dynamics are counted with much less weight than faster dynamics. On the other hand step input signals as shown in Figure 2 do not contain enough information to properly determine a model. Therefore a compromise has been made. The input signal has been produced by a random signal whose individual values have been prolonged randomly. An example of such an excitation signal is shown in Figure 9 (lower plot).

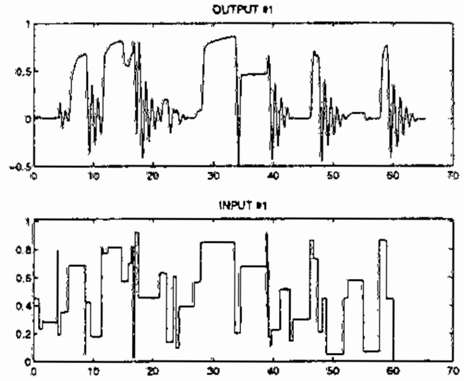


Figure 9: Identification signals

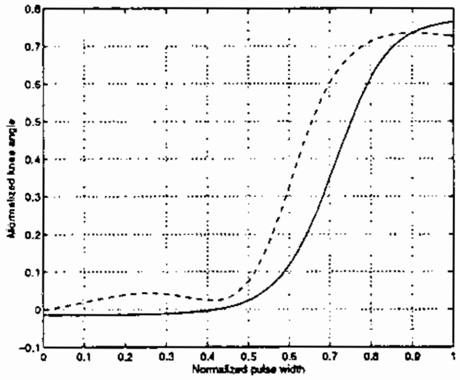


Figure 10: Estimation of input nonlinearity

Using the filtered data v_k to train the neural network we get an approximation of the input nonlinearity which is shown in Figure 10. The shape of the estimated nonlinearity is similar to the recruitment characteristics (see Figure 7) and is what we would expect based on physiological knowledge. The differences between the estimated input nonlinearity shown in Figure 10 and the input nonlinearity shown in Figure 7 merely rely upon the fact that the one mentioned previously is obtained from dynamical data whereas the latter is obtained from almost static measurements. Nevertheless, similarities are evident.

Whereas the prediction delivers almost perfect results (Fig. 11) the pure simulation (Fig. 12) reveals the weaknesses of the model. In both figures solid lines show the experimental data and dashed lines show the simulation. Although swing phases are modeled correctly the simulation overshoots when stimulation is initiated (*i.e.* when the knee angle is increasing).

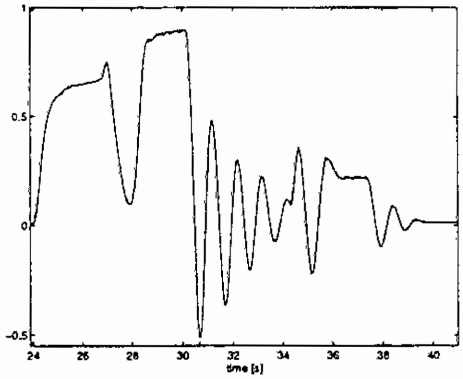


Figure 11: Detail of prediction with validation data

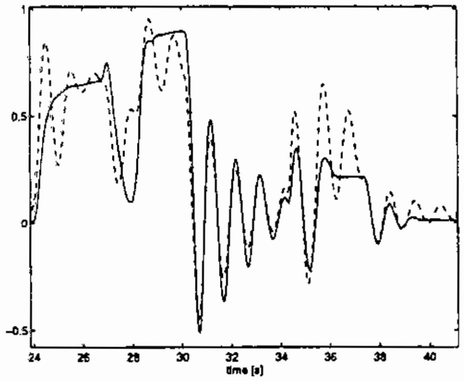


Figure 12: Detail of simulation with validation data

Conclusions

From the results above we can summarize

- The passive dynamics can be modeled very easily by a third order AR model. The performance of this model is excellent (Figure 4). Furthermore, the assumption that the passive swing back of the shank is a linear movement is valid to a large extent (Figure 3) and can be reproduced very well.
- One would expect that this linear dynamics are also valid when stimulation is switched on since physically, the "system" remains the same. However, when the muscle is generating force it acts as a damper and thus changes the dynamics.
- The impact of this damping term may be either position (output) as well as input dependent.
- With the Hammerstein approach commonly used in the literature it is impossible to correctly model both parts of the movement.

References

[1] Lucinda L. Baker, Donald R. McNeal, Laurel A. Benton, Bruce R. Bowman, and Robert L. Waters. *NeuroMuscular Electrical Stimulation: A Practical Guide*. Los Amigos Research & Education Institute, 3rd edition, 1993.

[2] L. A. Bernotas, P. E. Crago, and H. J. Chizeck. A discrete-time model of electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, 33(9):829–838, September 1986.

[3] S. Bittanti and L. Pirrodi. GMV technique for nonlinear control with neural networks. In *IEE Proc. of Control Theory Applications*, volume 141, pages 57 – 69, March 1994.

[4] L. del Re, F. Kraus, J. Schultheiss, and H. Gerber. Self-tuning PID controller for lower limb FES with nonlinear compensation. In *Proceedings of the 1994 American Control Conference*, volume 2, pages 2015–19, 1994.

[5] William K. Durfee and Karon E. MacLean. Methods for estimationg isometric recruitement curves of electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, 36(6):654–667, July 1989.

[6] M. S. Hatwell, B. J. Oderkerk, C. A. Sacher, and G. F. Inbar. The development of a model reference adaptive control for the knee joint of paraplegics. *IEEE Transactions on Automatic Control*, 36:683–691, 1991.

[7] Zoher Zain Karu. Optimization of Force and Fatigue Properties of Electrically Stimulated Human Skeletal Muscle. Master's thesis, MIT, 1992.

[8] M. Levy, J. Mizrahi, and Z. Susak. Recruitment, force and fatigue characteristics of quadriceps muscles of paraplegics isometrically activated by surface electrical stimulation. *Journal of Biomedical Engineering*, 12:150–156, March 1990.

[9] Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall Information, 1987.

[10] Li Chia Tien, Chow Po-Chuan, and Howard Jay Chizeck. Recursive parameter identification of constrained systems: An application to electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, 38(5):429–442, May 1991.

THE MECHANOMETRICS APPROACH TO MONITORING AND FAULT DETECTION OF A HYDRO-MECHANICAL SYSTEM

J. J. Milek and H. Güttinger
Sulzer Innotec Ltd., CH-8041 Winterthur, P.O. Box, Switzerland, e_mail: Janusz.Milek@Sulzer.ch

Abstract: Hydro-mechanical systems like water turbines or pumps become very well instrumented. As a result, the development of monitoring and fault detection systems for such processes attracts increasing attention. Unfortunately, physical modeling of hydro-mechanical systems may become a quite involved task, not feasible for industrial applications. This paper describes the mechanometrics approach to modeling and fault detection of hydro-mechanical systems. In this approach measurements made on a mechanical system are related to the state of the system via application of mathematical or statistical methods. It is discussed how the classical principal component analysis can be used to model basic process variables of a large Francis turbine in a water power plant. The model is constructed exclusively from the experimental data. It is shown how the data redundancy can be exploited for fault detection and isolation. The corresponding fault detection algorithm is tested in a simulation study.

Introduction

In the recent years industrial processes including hydro-mechanical systems become very well instrumented. At the same time the process data are measured more frequently [3,7]. Hence there is an increasing need to monitor such systems and detect faults. However, huge amounts of measured data have first to be processed in an appropriate way in order to extract only the useful information concerning the system behavior [8].

Mathematical models gained directly from the experimental data are useful means of such an information extraction. In particular the *Principal Component Analysis* (PCA) methods, popular in econometrics or in chemical engineering enable easy experimental modeling [6,9]. The paper aims at demonstrating that experimental modeling can be also applied with respect to the mechanical or hydro-mechanical systems. Following [8] we propose here the following definition: *Mechanometrics is the science of relating measurements made on a mechanical system to the state of the system via application of mathematical or statistical methods.* Using PCA a redundancy in the data can be analyzed and utilized for *Fault Detection and Isolation* (FDI).

Linear multivariable modeling via PCA

Linear models are perhaps the simplest but also the most efficient means of modeling various types of real processes [4].

Let the process data be stored in the data matrix Φ such that its columns φ_i, i=1..n, correspond to measured variables while rows to time samples. We assume here that the measured variables are zero mean and have identical variance, otherwise the data can be centered and normalized.

The data modeled by a linear multivariable model are supposed to satisfy a number of linear relations, i.e., to lie on a hyperplane in the data space. Hence the model can be expressed as

$$\Phi\Theta = 0, \qquad (1)$$

where Θ is a rank M matrix, spanning the null space of the data matrix Φ. Equation (1) can be thought of as M linear constraints of the form

$$\theta_1^m\varphi_1 + \theta_2^m\varphi_2 + \ldots + \theta_n^m\varphi_n = 0, \quad m = 1..M.$$

Equation (1) cannot be exactly satisfied due to unmodeled disturbances and other modeling deficiencies. Instead, one may require that

$$\|\Phi\Theta\|_2 \to \min, \quad \|\Theta\|_2 = 1. \qquad (2)$$

The solution of (2) can be computed via the so-called *Singular Value Decomposition* (SVD) [2] of Φ,

$$\Phi = U\Sigma V^T; \qquad (3)$$

it constitutes the last M columns of V. Using MATLAB notation [5] we can write the solution of (2) as

$$\Theta = V(:,[n\text{-}M+1:n]). \qquad (4)$$

The quadratic matrices U and V appearing in (3)-(4) are orthonormal. Σ is a rectangular matrix containing zeros outside the main diagonal and decreasingly ordered *singular values* σ_1, σ_2, ... σ_n on the main diagonal. The first n-M vectors of V matrix, collected in the model matrix Θ^*,

$$\Theta^* = V(:,[1:n\text{-}M]),$$

span the model hyperplane.

A common way to formulate the PCA problem is as follows: decompose the data matrix Φ into two matrices

$$\Phi = \Phi^* + E \qquad (5)$$

such that

$$||E||_F \text{->} \min, \quad \text{rank}\ \Phi^* = n-M,$$

where $||.||_F$ denotes the Frobenius norm [2]. The just formulated problem is equivalent to the one solved above. The matrices appearing in (5) can be expressed in terms of SVD as

$$\Phi^* = U\Sigma^* V^T$$

and

$$E = U(\Sigma - \Sigma^*)V^T,$$

where Σ^* denotes the matrix obtained from Σ by replacing the last M singular values by zeros.

The multivariable model of the system can be utilized for data smoothing, data reconstruction, and fault detection and isolation.

Data smoothing

Let matrix Ψ be organized like data matrix Φ but possibly do not contain data used to construct the model. An additive noise corrupting the data can be suppressed by orthogonally projecting Ψ on the model hyperspace Θ^*

$$\Psi^* = \Psi\Theta^*(\Theta^{*T}\Theta^*)^{-1}\Theta^{*T}.$$

Data reconstruction

The multivariable model can also be used to reconstruct a variable or a group of variables from the remaining data. The algorithm enabling reconstruction of i-th variable (denoted ψ_i and represented by i-th column of the data matrix Ψ) is composed of the following steps:

(i) elimination of i-th coordinate from the model hyperspace Θ^*

$$\Theta^*_{-i} = \Theta^*([1:i-1\ i+1:n],:),$$

(ii) elimination of i-th variable from the data matrix Ψ

$$\Psi_{-i} = \Psi(:,[1:i-1\ i+1:n]),$$

(iii) projection of the reduced data Ψ_{-i} on the reduced model hyperspace Θ^*_{-i} (smoothing of Ψ_{-i})

$$\Psi_{-i}^* = \Psi_{-i}\Theta^*_{-i}(\Theta^{*T}_{-i}\Theta^*_{-i})^{-1}\Theta^{*T}_{-i},$$

(iv) reconstruction of i-th variable

$$\psi^*_{i,-i} = -\Psi^*_{-i}\,\theta([1:i-1\ i+1:n])/\theta(i),$$

where $\theta(:)$ is any vector from the column space of Θ with a non-zero i-th entry $\theta(i)$.

Fault detection and isolation

A fault in one variable is to be acknowledged if, for any i, i=1..n, the norm of the reconstruction error of i-th variable scaled with respect to the corresponding threshold exceeds one

$$||\psi_i - \psi^*_{i,-i}||_2 \,/\, E_{i,\max} > 1.$$

(Thresholds $E_{i,\max}$ can be established from experimental data.) Under assumption that only one variable contains faulty data, its index 'i' can be isolated as the one that corresponds to the *smallest* value of the projection error $||\Psi_{-i} - \Psi_{-i}^*||_F$. The necessary condition for feasibility of fault detection is data redundancy (M>1).

The presented FDI algorithm can be extended for detection and isolation of faults in groups of variables and is equivalent to the algorithm analyzed in [1]. However, the algorithm from [1] has different sequence of operations (reconstruction precedes smoothing).

Example: modeling and variable reconstruction for a large Francis turbine

In the sequel we consider data collected from a 440 MW Francis turbine in water power plant. Daily mean values of the following most important process variables are recorded: (a) active power, (b) wicket gate, (c) pressure in the spiral case, (d) pressure head, (e) temperature of the cooling medium, (f) temperature of the generator at the non-driving side, (g) temperature of the generator at the driving side, (h) temperature of the axial bearing, and (i) temperature of the driving water.

The variables (a)-(i) are highly correlated. Hence, they can be well modeled using one multivariable model. The data are centered (mean value is subtracted), normalized with respect to the variance and stored column-wise in the data matrix Φ.

From (3) one obtains the following singular values (diagonal entries of Σ):

σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8	σ_9
30.5193	18.0601	10.8594	10.2031	3.2010	1.6375	1.0399	0.5223	0.3065

Table 1: Singular values of the data matrix Φ

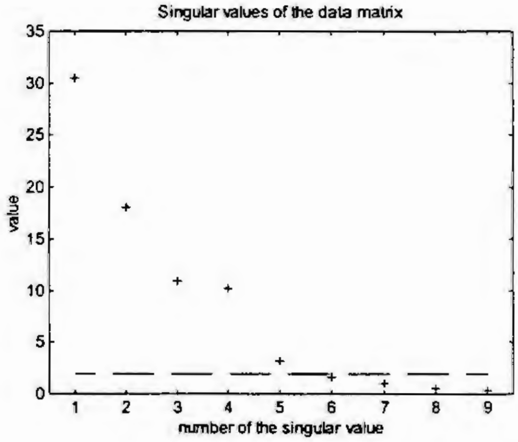


Figure 1: Singular values of the data matrix Φ and the threshold

Setting a threshold for the singular values at 2.0 as in Fig. 1 one gets a model with four degrees of freedom (the data are likely to lie on a 5 dimensional hyperplane in the 9 dimensional data space; hence four dimensions are canceled). The redundancy of the data will be further exploited for the fault detection and isolation.

The matrix Θ spanning the model null space is shown in Tab. 2.

-0.0131	0.1737	-0.8148	0.0402
0.0520	0.0506	0.1454	0.0073
0.0786	0.2347	0.3576	-0.7191
0.0577	0.2449	0.4195	0.6930
-0.6858	0.1152	0.0656	-0.0255
0.2776	-0.0539	-0.0427	0.0153
0.5384	0.3247	-0.0423	-0.0078
0.1699	-0.8502	0.0570	0.0020
-0.3486	-0.0880	0.0020	0.0033

Table 2: The matrix Θ which spans the model null space

Note that in this static model there is no distinction between system inputs and outputs. The original and reconstructed variables are shown in Figs. 2-6 (solid line - measurements and residuals, dotted line - reconstruction).

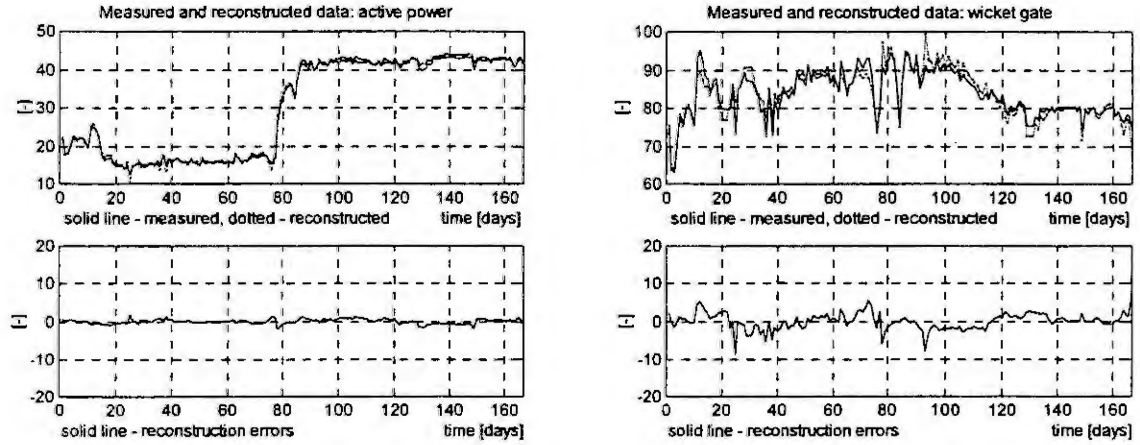


Figure 2: Modeling results for active power (left) and wicket gate (right)

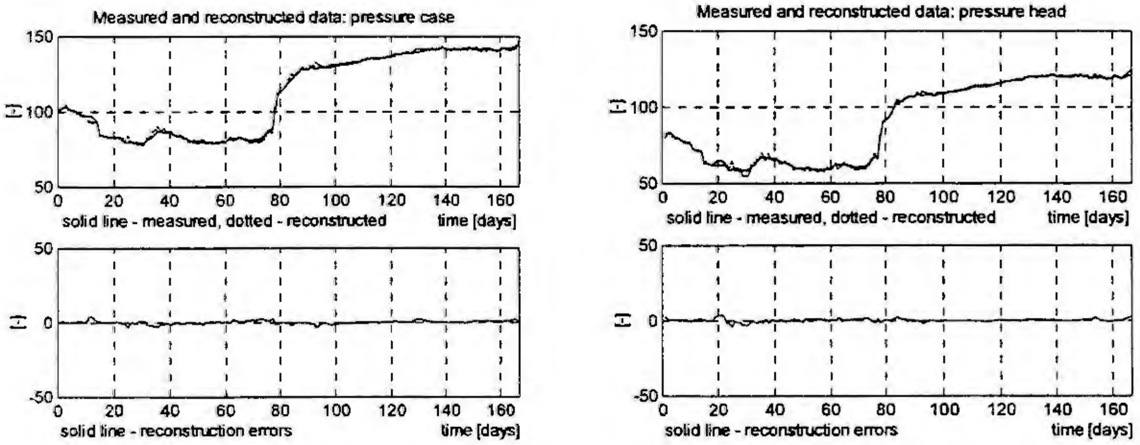


Figure 3: Modeling results for pressure in the case (left) and pressure head (right)

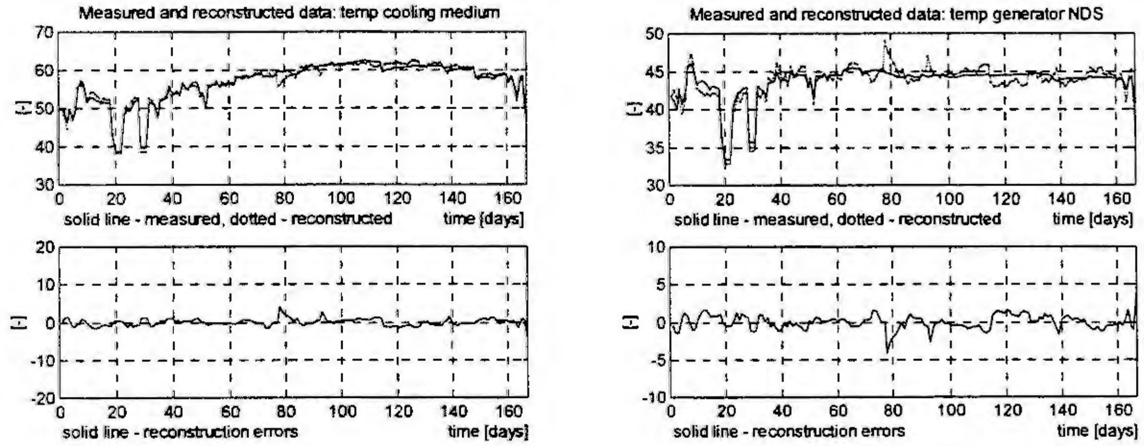


Figure 4: Modeling results for temperature of the cooling medium (left) and temperature of the generator at the non-driving side (right)

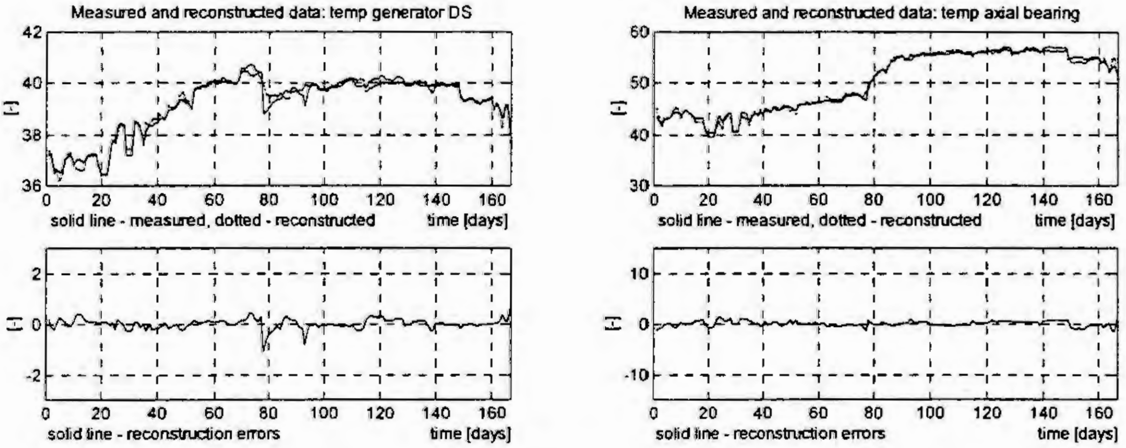


Figure 5: Modeling results for the generator temperature at the driving side (left) and the axial bearing temperature (right)

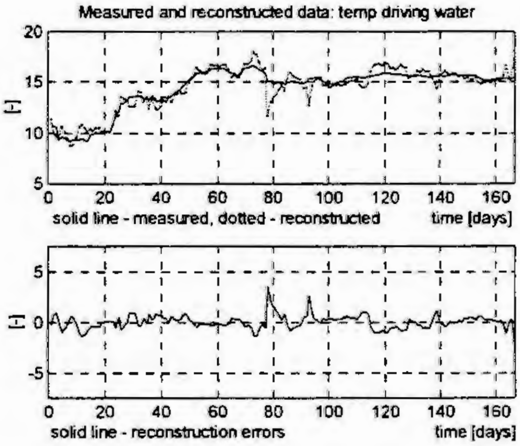


Figure 6: Modeling results for temperature of the driving water

The presented results demonstrate that most of the variables can be reconstructed from the other data with quite a good accuracy. The remaining worse modeled variables are also correlated with the reconstructed values.

Analyzing the singular values, the corresponding columns of the matrix Θ, and the modeling results we note that

(i) σ_9 and $\Theta(:,4)$: the strongest redundancy in the system is visible as the dependence between the pressures (c) and (d), since the coefficients $\Theta(3,4)$ and $\Theta(4,4)$ are much larger than the remaining ones in $\Theta(:,4)$. The redundancy is clearly visible in Fig. 3.

(ii) σ_8 and $\Theta(:,3)$: the second strongest dependence in the system is visible amongst the variables (a)-(d) (power, wicket gate, pressures). This dependence has clear physical reasons. The largest coefficient $\Theta(1,3)$ corresponds to the third best modeled variable, active power, shown in Fig. 2.

(iii) σ_7 and $\Theta(:,2)$: the third strongest dependence in the system concerns the variables (a), (c)-(e), (g) and (h) (wicket gate, temperature of the generator at the non-driving side and temperature of the driving water. The largest coefficient $\Theta(8,2)$ is related to the fourth best modeled variable, temperature of the axial bearing, shown in Fig. 5.

(iv) σ_6 and $\Theta(:,1)$: the fourth strongest dependence in the system exists for all the temperatures. This dependence clearly follows from the physics. The largest coefficients $\Theta(5,1)$ and $\Theta(7,1)$ are related to the fifth and sixth best modeled variables, the cooling medium temperature and temperature of the generator at the driving side, shown in Figs. 4-5.

Example continued: fault detection and isolation

A number of faults concerning one column of the data matrix are simulated. The column may contain (a) true data in a random order, (b) true data in the reversed time order, (c) white noise instead of the data, (d) highly correlated noise instead of the data, (e) a constant value instead of the data, (f) true data scaled by a constant. Each group of simulations contains several hundreds experiments. In each experiment there are ten possibilities (0-9): fault of one variable (denoted 1-9) or no fault (denoted 0). Probability of each event is 1/10. The FDI algorithm has to detect if a fault took place and, if so, to determine the number of the column of the data matrix that contains the wrong data. An exemplary simulation results are shown in Fig. 12. The 'true' index of the faulty variable is shown as a circle 'o'. The estimated index is shown as a cross '+'.

The simulation results are as follows: in all experiments the faults are properly detected and for only one experiment (from the total 1300 experiments) the fault is not correctly isolated.

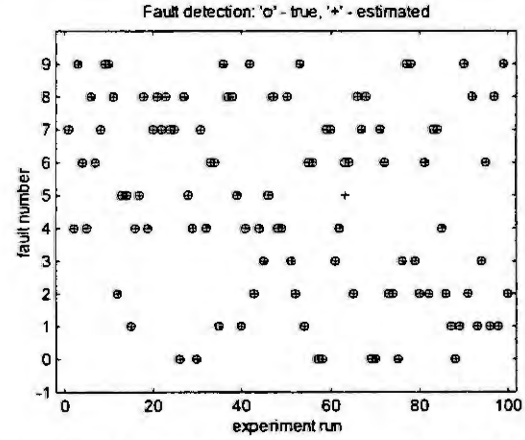


Figure 12: Results of the FDI algorithm if one column of the data matrix is replaced by a highly correlated noise

Conclusions

This paper addresses the problem of modeling and fault detection of a hydro-mechanical system, namely a large Francis turbine in a water power plant. The models are obtained directly from the experimental data via application of principal component analysis. It is shown how the data redundancy can be exploited for fault detection and isolation. The proposed fault detection algorithm is tested in a simulation study.

References

1. Dunia, R., Qin, S. J., Edgar, T. F., and McAvoy, T. J., Sensor Fault Identification and Reconstruction Using Principal Component Analysis. In: *Proceedings of the IFAC 13th Triennial World Congress*, San Francisco, 1996, Vol. N, 259-264.
2. Golub, G. H. and van Loan, C. F., *Matrix Computations*. North Oxford Academic, Oxford, 1983.
3. Güttinger, H. and Birchler, B., SUDIS - a Diagnostic System for Large Centrifugal Pumps. In: *Proceedings of COMADEM '96*, University of Sheffield, 1996, 431-440.
4. Ljung, L. and Glad, T., *Modeling of Dynamic Systems*. Prentice Hall, Englewood Cliffs, 1994.
5. *MATLAB User's Guide*. The MathWorks, Inc.. Natick, Massachusetts, 1993.
6. Raich, A. and Cinar, A., Process Disturbance Diagnosis by Statistical Distance and Angle Measures. In: *Proceedings of the IFAC 13th Triennial World Congress*, San Francisco, 1996, Vol. N, 283-288.
7. Sutter, R., Condition of Turbines at a Glance, *Sulzer Technical Review*, 3 (1995), 32-34.
8. Wise, B. M., Gallagher, N. B., and MacGregor, J. F., The Process Chemometrics Approach to Process Monitoring and Fault Detection. In: *Preprints of the IFAC Workshop on On-line Fault Detection and Supervision in the Chemical Process Industries*, Newcastle, 1995, 1-20.
9. Wise, B. M., Ricker, N. L., Veltkamp, D. J., and Kowalski, B. R., A Theoretical Basis for the Use of Principal Components Models for Monitoring Multivariate Processes. *Process Control and Quality*, 1 (1990), 41-51.

MODELING FOR CONTROL OF MEAN ARTERIAL BLOOD PRESSURE (MAP) DURING ANESTHESIA

C. W. Frei[1] **M. Derighetti**[1] **and A. M. Zbinden**[2]
[1] Automatic Control Laboratory
Swiss Federal Institute of Technology (ETH)
8092 Zürich
[2] Institute of Anesthesiology and Intensive Care
Research Department
Inselspital
3010 Bern

Abstract: In this paper we are studying models to be used for the design of automatic controllers for mean arterial blood pressure (MAP) during anesthesia. We demonstrate that for controller design a high order nonlinear model derived from physiological arguments leads to weakly observable and controllable linearized models that are best reduced to lower order. Order suggestions are given based on the linearization of a physiological compartment model.

1 Introduction

One important variable to be kept within acceptable limits during anesthesia is mean arterial blood pressure (MAP). A number of researchers have attempted to automate control of blood pressure during anesthesia [1, 2, 3]. Different approaches of controller design have been made. Some of the researchers applied non-model based control strategies such as fuzzy logic [2] or neural networks, others used linear black box models derived from experimental data [3], and some used models derived form physiological considerations.

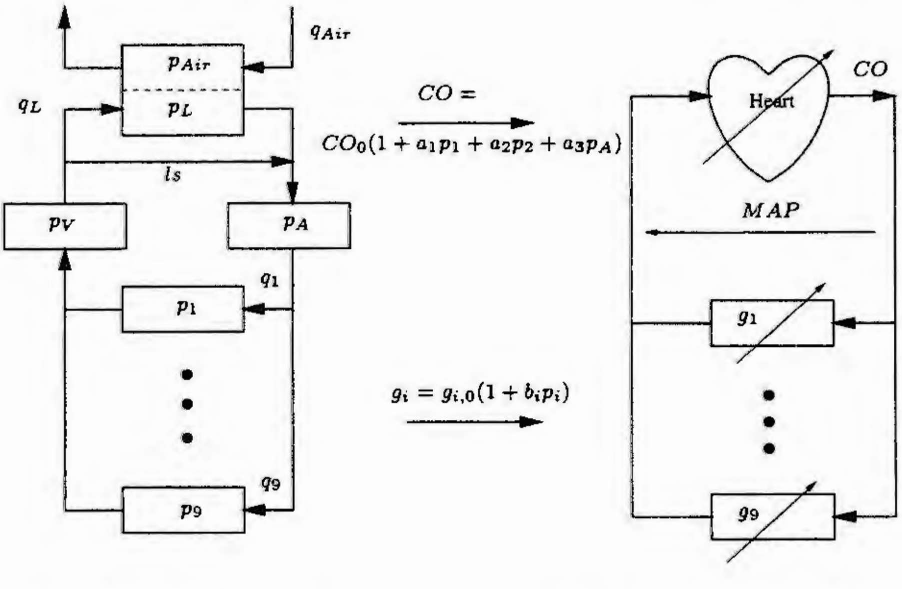


Figure 1: The physiological model consist of a part (left) that describes uptake and distribution of volatile anesthetics, and another a part (right) that describes blood circulation and the pharmacodynamic interconnection as it was proposed in [4]. The variables p_i denote the partial pressure of the anesthetic agent, the q_i denote blood flows, CO denotes cardiac output, the g_i denote conductivities, the a_i and b_i are parameters (to be defined below) and ls denotes the lung shunt (i.e. blood that passes the lung unaffected).

A fairly accurate, yet not too complex model of how MAP depends on the concentration of halothane in a body was given in [4]. With appropriate modifications of parameters the model structure can also be used for other volatile anesthetics such as isoflurane [5]. The model has been validated by several groups (e.g. [5]) and it was found to describe the process accurately. It has successfully been applied to different purposes [6, 2]. In [6] an anesthesia simulator for the education of anesthetists was developed and in [2] it was used to tune a fuzzy controller. In this paper we answer the question whether this model derived from physiological considerations can be used to design automatic controllers.

2 The Compartment Model

In this section we first reintroduce the model of [4]. It consists of two parts - one for the uptake and distribution of drugs and one for the circulation of the blood flows. To describe the drug distribution in the body a compartment model is established (left part of figure 1). The circulation model (right part of figure 1) is utilized to describe the hemodynamics. Pharmacodynamic laws are postulated to link the two models. Several simplifying assumptions were made for the derivation. These are also listed in [4].

2.1 The model for distribution of anesthetics

The uptake and distribution model (left part of figure 1) describes the pharmakokinetics of the anesthetic agent. It consists of 12 different compartments (see table 1). Compartments consist of organs or groups of organs that are assumed to have similar properties concerning uptake and distribution of an anesthetic agent, as well as conductivity and pharmacodynamics. The compartments 1 to 9 are referred to as normal compartments. Two compartments model the arterial and venous blood pool, respectively. And one compartment models the lung.

1	myocard (heart muscle)
2	brain gray matter
3	brain white matter
4	well perfused organs
5	poorly perfused organs
6	splanchnicus (stomach,intestine)
7	skeletal muscle
8	fat
9	skin shunt
L	lung
A	arterial system
V	venous system

Table 1: List of the different compartments

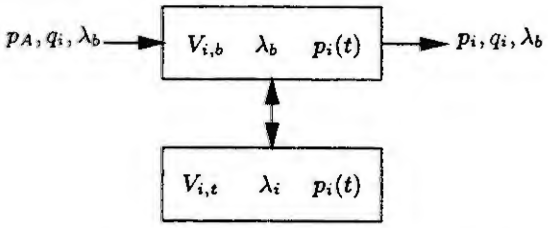

p_A, q_i, λ_b
$V_{i,b}$ λ_b $p_i(t)$
p_i, q_i, λ_b
$V_{i,t}$ λ_i $p_i(t)$

Figure 2: The model for a normal tissue compartment consists of a blood part and a tissue part with volumes $V_{i,b}$ and $V_{i,t}$ respectively. Each part has different solubility λ_b and λ_i but (by assumption) the same partial pressure of the anesthetic.

All of the normal compartments have the same structure (figure 2). They are modeled as consisting of a tissue and a blood part. Both, tissue and blood part are assumed to have the same partial pressure p_i but different solubilities. Anesthetics enter the compartment with blood and partial pressure p_A and they leave the compartment with the venous blood and partial pressure p_i. Within the compartment blood and tissue are assumed to have the same partial pressure. This description leads to the following equation for the evolution of partial pressure of the anesthetic in the compartment:

$$p_i(t) = p_i(0) + \frac{1}{\lambda_b V_{i,b} + \lambda_i V_{i,t}} \int_0^t \lambda_b q_i(\tau) \left[p_A(\tau) - p_i(\tau)\right] d\tau \tag{1}$$

The three extra compartments (L, A, V) have a slightly different structure. The lung compartment has in addition to the blood and tissue volumes, functional residual capacity that has to be taken into account, and there is a second possibility (besides transportation with the blood) for anesthetics to enter the compartment. This leads to the equation:

$$\begin{aligned} p_L(t) &= p_L(0) + \frac{1}{\lambda_b V_{i,b} + \lambda_L V_{L,t} + V_a} \int_0^t \{\lambda_b q_L(\tau) \left[p_V(\tau) - p_L(\tau)\right] \\ &\quad + q_{Air}(\tau) \left[p_{Air}(\tau) - p_V(\tau)\right]\} d\tau \end{aligned} \tag{2}$$

where q_{Air} denotes the minute volume and p_{Air} denotes the anesthetic gas concentration (partial pressure) of the inspired air.
Arterial and venous compartments differ from the normal compartments in the balance equations in so far as their flows entering and leaving the compartments are different. The equations are:

$$p_A(t) = p_A(0) + \frac{1}{\lambda_b V_{i,b} + \lambda_A V_{A,t}} \int_0^t \lambda_b CO(\tau) \left[p_V ls + p_L(1 - ls) - p_A(\tau)\right] d\tau \tag{3}$$

$$p_V(t) = p_V(0) + \frac{1}{\lambda_b V_{i,b} + \lambda_V V_{V,t}} \int_0^t \lambda_b \left[\sum_{i=1}^{9} q_i(\tau) p_i(\tau) - CO(\tau) p_V(\tau)\right] d\tau \tag{4}$$

2.2 The circulation model

The circulation model describes the blood flow. The heart produces a certain amount of average blood outflow (cardiac output denoted by CO). The total CO is distributed to the various normal compartments. Each of these compartments exhibits a certain conductance g_i. Given CO and g_i, the mean arterial blood pressure MAP is given analogously to Ohm's law by

$$MAP = \frac{CO}{\sum_{i=1}^{9} g_i} \tag{5}$$

2.3 The pharmacodynamics

Pharmacodynamics describe the effects of drugs. Effects of the anesthetic agent on the compartment conductivity and the cardiac output are known and are modeled as affine functions of partial pressure of the anesthetic in the compartment. That is:

$$g_i = g_{i,0}(1 + b_i p_i) \tag{6}$$

$$CO = CO_0(1 + a_1 p_1 + a_2 p_2 + a_3 p_A) \tag{7}$$

2.4 The combined nonlinear model

The formulation of the model of [4] given by equations (1) to (7) was probably chosen because it was meant to be implemented on a analog computer. For our purpose we prefer a (nonlinear) model of the form

$$\begin{aligned} \dot{\mathbf{p}}(t) &= \mathbf{f}(\mathbf{p}(t), u(t)) \\ MAP(t) &= h(\mathbf{p}(t), u(t)) \end{aligned} \tag{8}$$

where the state vector $\mathbf{p}(t)$ describes the partial pressure of the anesthetic in every compartment, the input is the concentration of anesthetic gas in the inspired air, and mean arterial pressure MAP is the output of the system. Utilizing equations (1) to (7) we get the different components of the function f as:

$$\dot{p_i} = k_i g_{i,0} CO_0 \frac{(1+b_i p_i)(1+a_1 p_1 + a_2 p_2 + a_3 p_A)(p_A - p_i)}{\sum\limits_{j=1}^{9} g_{j,0}(1+b_j p_j)} \tag{9}$$

$$\dot{p_L} = k_L \left\{\lambda_b (1-ls) CO_0 (1+a_1 p_1 + a_2 p_2 + a_3 p_A)(p_V - p_L) + q_{Air}(p_{Air} - p_L)\right\} \tag{10}$$

$$\dot{p_A} = k_A CO_0 (1+a_1 p_1 + a_2 p_2 + a_3 p_A)\left[p_V ls + p_L(1-ls) - p_A)\right] \tag{11}$$

$$\dot{p_V} = k_V CO_0 (1+a_1 p_1 + a_2 p_2 + a_3 p_A) \left[\frac{\sum\limits_{i=1}^{9} g_{i,0}(1+b_i p_i) p_i}{\sum\limits_{j=1}^{9} g_{j,0}(1+b_j p_j)} - p_V \right]. \tag{12}$$

$$MAP = CO_0 \frac{(1+a_1 p_1 + a_2 p_2 + a_3 p_A)}{\sum\limits_{j=1}^{9} g_{j,0}(1+b_j p_j)} \tag{13}$$

$$\tag{14}$$

This model obviously requires the determination of a lot of parameters. However, there is no need to identify all of them from experimental input/output data alone. A lot of the parameters (e.g. blood volumes) are know form anatomy, some parameters (e.g. solubility) can be found in anesthesia literature, and only a few have to be determined individually.

3 The linearized model

The system model (8) is nonlinear. For the purpose of controller design, however, one might prefer a linear model since numerous straight forward techniques exist to derive controllers for linear plants. Thus, for controller design it is preferable to use a linear model obtained by linearization around a nominal operating point.

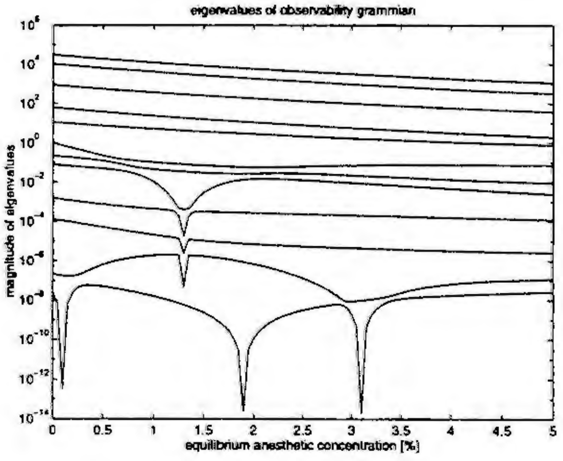


Figure 3: Eigenvalues of the observability grammian as a function of the equilibrium point. Large values correspond to well observable modes and small values correspond to weakly observable modes.

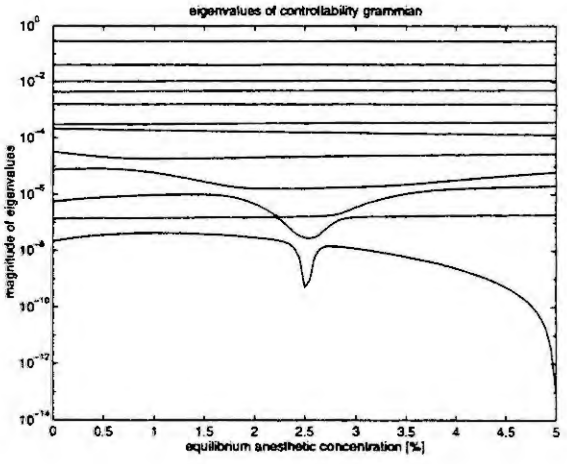


Figure 4: Eigenvalues of the controllability grammian as a function of the equilibrium point. Large values correspond to well controllable modes and small values correspond to weakly controllable modes.

It is easy to show that the system 8 has - although nonlinear - only one equilibrium point which is:

$$p_i = p_A = p_V = p_L = p_{Air} \tag{15}$$

This means that the system is at rest if the partial pressure of every single compartment is equal to the inspired partial pressure (which is intuitively clear). This suggests that it is best to linearize the system around the average inspired anesthetic concentration $\overline{p}_{Air}$.

Now there are still two problems associated with this linearized model: It is of high order and it is weakly observable and weakly controllable.

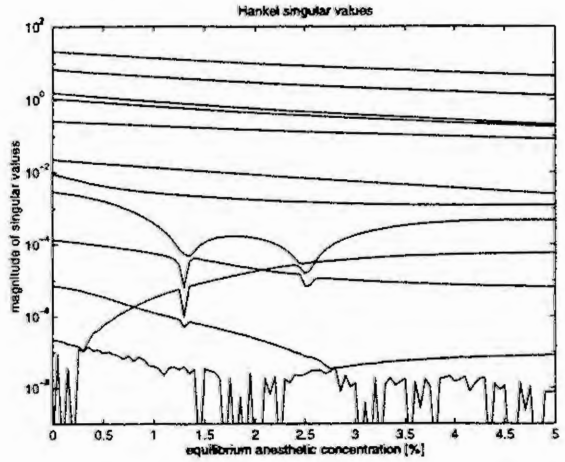


Figure 5: Hankel singular values varying with operating point $\overline{p}_{Air}$.

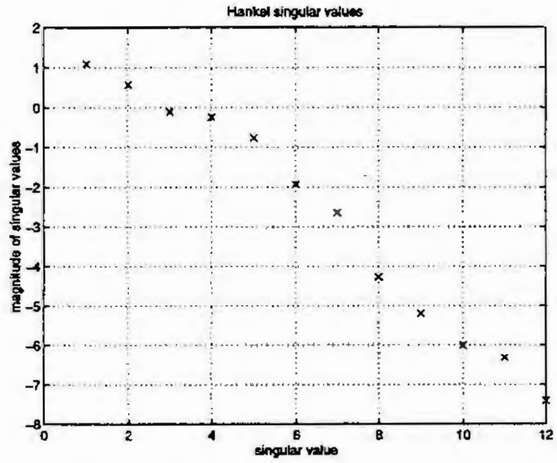


Figure 6: Hankel singular values for $\overline{p}_{Air} = 1$.

High model order is mainly a problem from a computational point of view. If, for example, a model predictive control (MPC) scheme is used where an online optimization has to be made for every control move high model order may render it impossible to find a solution to this optimization problem within the given time. Low model orders are therefore desirable.

Considering observability and controllability of the model a reduction of the order seems reasonable. Figures 3 and 4 show how the eigenvalues of observability and controllability grammians change as the operating point (point for linearization) is varied. Large eigenvalues grant well observable and well controllable modes, respectively. Small values correspond to weakly observable and controllable modes. Since the largest eigenvalues are magnitudes larger than most of the other eigenvalues (notice the log scale), and since the state equations are already properly scaled a model reduction should be considered.

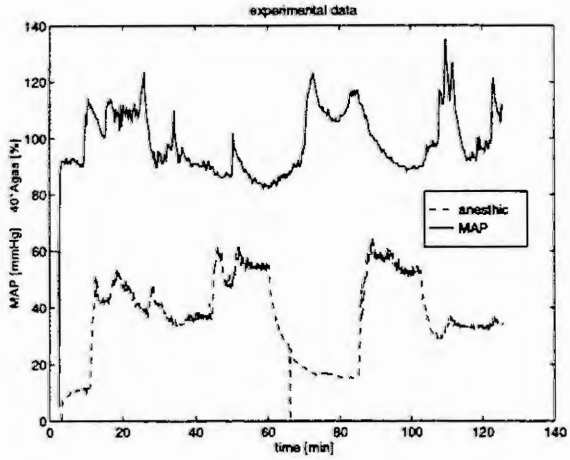


Figure 7: Experimental data used for validation. Notice that the concentrations are scaled by a factor of 40.

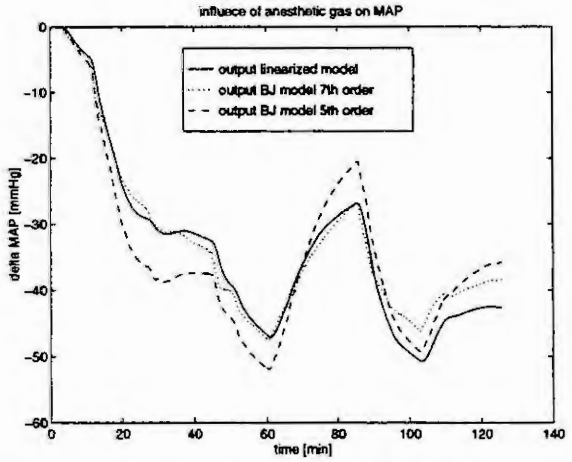


Figure 8: Response of different models to the actual input of the experiment.

How to reduce the model for control purpose cannot be decided from observability and controllability grammian alone since directions that are well observable may not necessarily be well controllable. The reduction has to be balanced and what model order has to be chosen in this context can be deduced from the Hankel singular values. Figure 5 shows how the Hankel singular values vary as the operating point is varied. It is evident that the dominant singular values do not change much with the operating point. What model orders might be appropriate can best be seen from figure 6. Here, the Hankel singular values are plotted for $\overline{p}_{Air} = 1$ which corresponds to $\overline{p}_{Air}$ of the experimental data to follow. In regard to the staircase-like structure it becomes obvious what orders might be appropriate. A very crude model could be of second order according two very significant singular values. A more detailed model of 5th order is obtained by taking the next group of singular values into account. Choosing model orders for control greater than 7 does not make sense according to the remaining singular values.

4 Experimental Validation

In this section we show that the order suggestions derived from the physiological model indeed yield good identification results. Figure 7 shows blood pressure and inspired gas concentration data recorded during an operation. The study for which the data were recorded was designed to evaluate inspiratory controllers which had to follow step changes in reference concentration. This means that the system to be identified was operated in open loop.

Numerous responses to disturbances (painful stimuli) are easily identified in the blood pressure signal just by inspection. It can be seen that the disturbance dynamics are different from the plant dynamics and that for identification a Box Jenkins model should be considered. Figure 8 shows the simulation results (input was taken from the experimental data) for the two models identified from the experimental data and the linearized model.

5 Conclusions

We have seen that the order of a linearized model derived from a physiological compartment model is too high. Model orders of 5 or 7 seem to be sufficient to model patient responses to volatile anesthetics for controller design purposes.

It should be noticed that the states of the reduced order model do not have a physiological interpretation anymore and that the reduction does not correspond to removing a compartment in the distribution model figure 1.

References

[1] S. Chaudhri, J.R. Colvin, J.G. Todd, and G.N. Kenny, "Evaluation of closed loop control of arterial pressure during hypotensive anaesthesia for local resection of intraocular melanoma", *British Journal of Anaesthesia*, vol. 69, pp. 607–610, 1992.

[2] Marco Derighetti, "Multivariable Fuzzy-Regelung in der Anästhesie", Post Graduate Thesis, Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), 1993.

[3] D.A. Linkens, M. Mahfouf, and M. Abbod, "Self-adaptive and self-organizing control applied to nonlinear multivariable anaesthesia: a comparative model-based study", *IEE Proceedings-D*, vol. 139, no. 4, pp. 381–394, July 1992.

[4] A. Zwart, N.T. Smith, and E.W. Beneken, "Multiple model approach to uptake and distribution of halothane: The use of an analog computer", *Computers and Biomedical Research*, vol. 5, pp. 228–238, 1972.

[5] Nicolet Alexandre, *Programme de simulation de la pharmacocinétique et de la pharmacodynamique des anesthésiques par inhalation*, PhD thesis, University of Berne, 1995.

[6] A. Murbach, "Computermodell der Aufnahme volatile Anästhetika", Informatikprojekt, Institut für Informatik und angewandte Mathematik der Universität Bern, May 1991.

Continuous and discrete time representations in modelling and identification

Franta Kraus and Walter Schaufelberger
Institut für Automatik
ETHZ - CH-8 092 Zürich - Switzerland
kraus@aut.ee.ethz.ch

Abstract

Models of processes are used for different purposes: For the representation of systems, simulation, controller design, fault detection etc. Depending on the application type different models are needed. Several representations are available to serve these purposes. Available modeling techniques are discussed in the paper and proposals are made for a proper use of models in given circumstancies.

I. Models and their use

The following are major uses of models:

o Simulation, improved understanding of systems

Models are often built to improve understanding of a given process by simulation. The internal structure of the model, the relationship and the interaction between different subsystems and internal signals are important. These models are usually in continuous time and may be very complex. Controllability and observability of such models are usually not checked.

o Controller design

Design techniques available ask for rather simple and low order models. The range of high model precision is prespecified by the controller design goal. Experience shows that these simple models are most often sufficient for controller design. They differ significantly from long time range simulation models for the same process.

o Discrete events systems

Discrete events systems are an extra class of models for processes, which include generalised decisions (often externally driven). Thereby a new kind of signals - events - occurs. The appearance of an event changes the behaviour of the overall system. In their nature these models are hybrid including discrete automaton (ev. time dependent) and continuous submodels.

o Fault detection and isolation

Models or filters that show reactions to faulty conditions are increasingly applied in the new field of fault detection and isolation. The faults can be viewed as a special kind of discrete events.

The use of the models dictates many of their properties. Accuracy, range of validity, purpose of use - they all influence the way in which models are set up. A good model for controller design may not be a good model for simulation and vice versa. Even for simulation the time horizon of interest usually leeds to completely different models. In the following we will focus our discussion on classical models for continuous processes.

II. Continuous versus discrete time representations

Models that are obtained from first principles (as for example energy preservation) are usually formulated in continuous time. Design or implementation often asks for a discrete time representation. During the development of an overall system including possibly a controller, both continuous and discrete time represention are used. It is therefore interesting to compare the two approaches.

- Advantages of continuous time representations
  - linear and nonlinear models are locally compatible.
  - Physical interpretation is possible.
  - Tunable accuracy in local regions (of state space, frequency range etc.).
  - Structure information may be used (building systems from parts).
  - May be converted to discrete time for various sampling times.
  - Refinement or simplification of a model can be physically justified.
- Disadvantages of continuous time representations
  - The basic equations are not suitable for identification (derivatives).
  - Direct simulation is not possible (only in analog computers for relatively simple systems).
  - Advantages are lost in special forms such as the normal forms.
- Advantages of discrete time representations
  - Easy for identification but numerically difficult.
  - Simple integration (recursion).
  - Adapted to a simple form of the input signals (as staircase).
- Disadvantages of discrete time representations
  - Only for linear systems there exists a simple connection to the continuous time representation.
  - Difficulty of obtaining nonlinear difference equations.
  - Physical interpretation of parameters lost.
  - A simple physical parameter change may be spread over the entire system.
  - Numerical difficulties for $T \rightarrow 0$ (can be improved by δ parametrisation).
  - No adaptation of step length.
  - Difficulty to use structure information, separation into subsystems.
  - Sampling time fixed at the modelling stage.

III. Simulation of systems

There are major differences between the simulation of models in continuous and in discrete time.

Let us assume, the system description is first given as the standard, nonlinear differential equation

$$\frac{d}{dt}x = F(x(t), u(t), t) \quad \text{with} \quad x_o = x(t_o)$$

where $F(\cdot)$ is a smooth, differentiable function with a Taylor expansion with respect to t. We will now compute the trajectory $x(t)$ using for example the Runge-Kutta integration method RK7/8, which is one of the integration methods implemented in MATLAB/SIMULINK. To obtain the next trajectory point

$$x(t_o + h^*) = x(t_o) + h^* \cdot \Delta(x(t_o), u(t_o), h^*)$$

we have to

1. calculate the difference $\delta(\cdot)$ for a stepsize h^* using RK7
2. check the integration accuracy with the more precise RK8 and adapt the stepsize h^* if necessary

Thereby the first step of the algorithms is done in such a way, that the Taylor expansion of $x(t)$ in the point t_o w.r.t. h for the first 7 terms is identical to the expansion of

$$x(t_o) + h \cdot \Delta(x(t_o), u(t_o), h)$$

From this we can conclude some properties of the simulation method

- the trajectory is obtained locally by extrapolation. Comparing to a general, nonlinear diference model (with a fixed stepsize)
$$x_{k+1} = H(x_k, u_k)$$
RK7/8 generates nothing else as a local approximation of $H(\cdot)$ without an explicit parametrisation.
- the stepsize is locally optimised.
- for linear systems the solution corresponds to low accuracy approximation (only 7 terms of the expansion are considered).
- if $u(t)$ is not a known function of time, the time derivatives of $u(t)$ can not be calculated. We have to approximate $u(t)$ within the stepsize with a given function (for example a constant).
- we have to simulate the overall system at once. It is not possible to simulate the subsystems separately.
- the method is not suitable for non-smooth $u(t)$ inside the stepsize interval.
- advantage of adapted stepsize can be lost, if a digital controller with a short sampling time is incorporated in the overall system.
- although the sampling time is not fixed during the modelling stage its range is prespecified by modelling assumptions.

The corresponding simulation for discrete time system descriptions is just the standard recursion. Beside the problem to obtain a (parametrised) nonlinear, difference description even for the linear systems and simple forms of the input $u(t)$ it is not a priori clear, which way of simulation is numerically more tractable - more accurate local solution for the transition $x_k \rightarrow x_{k+1}$ for a fixed, not adjusted stepsize by discrete time representation or adapted stepsize and less accurate local approximation for time continuous representation.

IV. Identification of system parameters

There are also major differences between the two model categories when using identification techniques. We focus our discussion now just on linear models.

A. Identification of discrete time models

In general PEM-based identification techniques are well fitted for time discrete identification. Input / output values u_k, y_k appearing in the difference equation

$$y(k) + a_1 y(k-1) + \cdots + a_{n_a} y(k-n_a) = b_1 u(k-1) + \cdots + b_{n_b} u(k-n_b) + e(t)$$

are directly measurable. Although this description looks very appealing, it hides some inherent difficulties

- the I/O description is numerically poorly conditioned.
- for short sampling times the a_i-coefficients approach the binomial coefficients independently of the system dynamics and the nominator coefficients approach zero. The relative accuracy of the b_i coefficients is therefore poor.
- internal signals can not be easily used to improve the identification task.
- because of the numerical aspects, it is very difficult to obtain models valid over a large dynamical range.

Some new subspace identification techniques, essentially performing indirectly in the state space, resolve some of the numerical problems.

B. Identification of continuous time models

Although the I/O description of a linear, time continuous system looks very similar to a time discrete one
$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y(t) = b_1 u^{(n-1)}(t) + \cdots + b_n u(t) + e(t)$$

where $y^{(i)}, u^{(i)}$ denote the i-th time derivative of the input and output signals, there is a fundamental difference: the signals $y^{(i)}, u^{(i)}$ are not measurable. Therefore an indirect schema such as Modulating function method, Identification in frequency domain or PEM combined with a numerically optimisation (based on time discrete models) as implemented in the Identification Toolbox of Matlab must be used.

Even for periodical signals all of these methods applied to time discrete measurements are basically discrete time methods. Because of the staircase like form of the input signal the Shanon sampling theorem is not fulfilled. The aliasing /frequency folding produces errors for all calculations on output signals. Nevertheless using the time continuous parametrisation we have a more natural scaling of the unknown parameters, of gradients during the optimisation as well as a coupling between the allowed changes of the time discrete parameters (model family bounding).

V. Examples

A. *Organizational modelling (Modelling for understanding and simulation)*

The general diagrams of systems behavior and the archetypes introduced by Senge 1990 for management systems are very similar to the ones used in general simulation and system dynamics. Typical limits to growth structure is modelled and simulated by a control engineering minitool (Jia and Schaufelberger 1995):

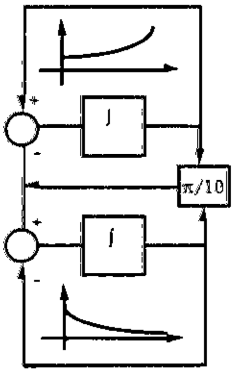


Fig. 1. Structure of a Lotka-Volterra model (limits to growth system) with a positive and a negative feedback loop, notation following Senge.

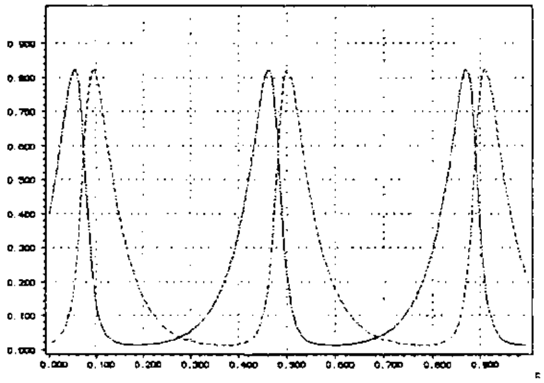


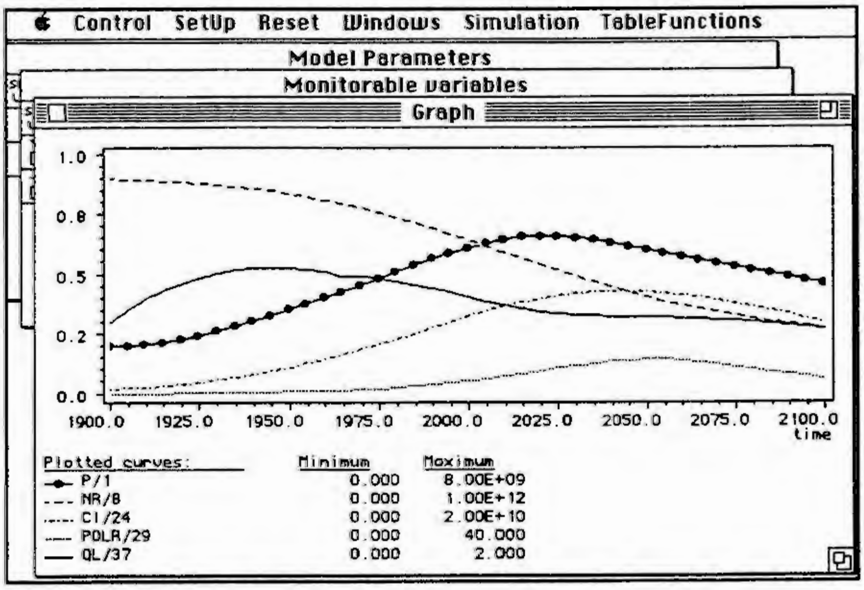
Fig. 2. Behavior of Lotka Volterra model, calculated by a minitool

B. *World models*

World models of Forrester and Meadows (Modelling for understanding and simulation).

Fischlin et. al. 1990 describe a simulation environment for students which allows investigations on the well known world model by Forrester and Meadows. Quotation from www, from where the software is downloadable:

"Worldmodel 2" tries to describe the behavior of the human population on earth considering certain processes as industrial growth, consumption of fossil resources, agriculture and environmental pollution. Starting with the year 1900, it rebuilds first the conditions in the past, and then predicts the future behaviour of population, food and energy balance until the end of the 21st century. "Worldmodel 3", a slightly refined successor, served as a basis for the publicated report of the "Club of Rome", concerning the situation of the world population: "Limits of Growth" (Meadows, 1972).



Plotted curves:	Minimum	Maximum
—●— P/1	0.000	8.00E+09
- - - NR/8	0.000	1.00E+12
.-.-. CI/24	0.000	2.00E+10
..... POLR/29	0.000	40.000
—— QL/37	0.000	2.000

Fig. 3. World model of Forrester: Simulation environment for students.
P: Population, NR: Natural Resources, CI: Capital Investment, POLR: Pollution Ratio, QL: Quality of Life.

C. *Servo (Modelling for controller design)*

A classical servo process is often used for demonstrations and teaching purposes to illustrate identification, control design and implementation as well as recursive estimation and adaptive control.

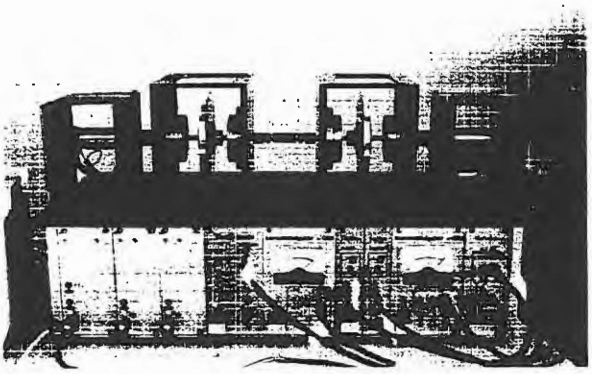


Fig. 4. Laboratory experiment setup: Servo System, scanned photo

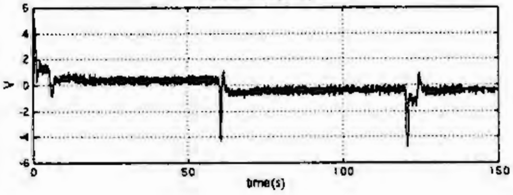


Fig. 5. Discrete time control of servo system

Results from system identification:

For this simple dynamical process it was possible to identify a time discrete model as well as a time continuous one. The main difference was, that for the discrete time identification much more effort was needed to find a suitable input sequence to obtain acceptable, accurate results. Thereby the identification success was not visible by simulation nor clearly by validation with a similar input sequence but immediately after the implementation of the controller.

VI. Conclusions

Models of a process can not be developed without a knowledge of the intended purpose of the model and of the overall system design goals. Thereby for time continuous processes it seems to be favourable to keep the continuous time description as long as possible.

VII. References:

Fischlin A., et al.: Modelworks, An Interactive Simulation Environment for Personal Computers and Workstations. ETH-Zürich, 1990.

Fischlin A., et al.: Unterrichtsprogramm Weltmodell 2, Report No. 1 of Institute of Terrestrial Ecology, ETH Zürich, 1990.

Jia L., Schaufelberger W.: Software for Control Engineering Education. vdf ETH-Zürich, 1995.

Meadows D. H., et al.: The Limits to Growth. A report for the Club of Rome's project on the predicament of mankind (Potomac Associates), 1972.

Senge P. M.: The Fifth Discipline, The Art and Practice of The Learning Organization. Doubleday, 1990.

Kraus F. and Schaufelberger W.: Identification with Modulating Functions, Differential Operators and Block Pulse Functios. Int. J. Control, Vol.51, No.4,pp. 931-942, 1990.

van Overschee P. and De Moor B.: Subspace Identification for Linear Systems. Kluwer Academic Publishers, 1996.

Ljung L.: Issues in System Identification, IEEE Control Systems, Vol. 11, No. 1, pp. 25-29, 1991.

Ljung L.: System Identification Toolbox for use with Matlab. The Mathworks Inc., Mass. USA, 1991.

Ljung L.: System Identification - Theory for the User. Prentice Hall, 1987.

Gevers M.: Towards a Joint Design of Identification and Control?; In: Trentelman and Willems (Eds.), Essay on Control: Perspectives in the Theory and its Applications. Birkhäuser, pp. 111-151, 1993.

Schrama R.J.P.: Accurate Identification for Control: The Necessity of an Iterative Scheme. IEEE Tr. AC., Vol. 37, pp. 991-994, 1992.

Kraus F., Qiu X. and Schaufelberger W.: Identification and Control of a Servosystem. Submitted for SYSID, 1997.

Derighetti M. et al.: Laboratory Experiments for an Integrated Basic Control Course. 13th IFAC World Congress, Vol. G, pp. 35-40, 1996.

Milek J.: A Lecture Demonstration for a Course in Digital Control. 13th IFAC World Congress, Vol. G, pp. 13-18, 1996.

THIN - STRIP CASTING - MODELING OF THE THICKNESS PROFILE WITH NEURAL NETWORKS

U. Albrecht-Früh*, R. Kopp
Institute for Metal Forming, RWTH Aachen , Intzestr.10, 52072 Aachen, Germany
Tel: +49/(0)241/80-3547, FAX: +49/(0)241/8888-234, alf@ibf.rwth-aachen.de

Abstract

The paper discribes the complexity of the combined casting/metal-forming process „Twin-Roll Strip Casting" for steel products, which is a focus of intensive research activities of steel producers. The influence on and the interaction between the strip temperature and thickness profile are the main aspects to produce the required quality. Considering a special strip attribute, the modeling of the process becomes more and more important. To analyse the strip forming process the modeling had been devided into two subsystems. One model contents the roll in order to investigate the thermal expansion with a Finite Element Model. The other systems include the melt pool in order to calculate the flow- and temperature field which was done by a Finite Volume Model. The final criteria for a successful calculation of each subsystem is directly related to the integration of the local interactions between melt and roll surface. Concerning this complexity in using the convenient boundary conditions both of the separated models could just fit special static operating points.

To improve the prediction of the strip thickness profile concerning industrial requirements and changing operating points, modeling with Artifical Neural Networks has been investigated. The results of the prognosis and their dependence on the process parameters will be discussed.

Introduction

In addition to the high pressure of cost and market, an enormous process shortening development has been carried out to produce thin-strips of steel. A special potential was seen in the „Near Net Shape Casting" - strategy, which is today worldwide undertaken from numerous companies and in different process variations. Some are already realized in industral scale [1].

One of those processes is the Twin-Roll Strip Casting technique (Figure 1). Since 1988, the Institute for Metal Forming (IBF) together with the Institute for Automatic Control at the RWTH Aachen and the Thyssen Stahl AG is operating a pilot-scale Twin-Roll strip caster. The steel melt is cast via a tundish between two horizontal internally cooled and in opposite direction rotating rolls. Two ceramic sealing plates are positioned on the roll faces to prevent the melt flowing out. Together with the copper-rolls, they form a mould for the melt pool. On both roll surfaces, the melt solidifies into shell strips, which are then joined in the narrowest working gap (kissing point). The strip is guided by a driver into a cooling section and is then coiled. Although Sir H. Bessemer already patented this process in 1856, the industrial realization could only be obtained as far as the requirements to operate the high non-linear and time-variable process were solved by a complex control system [2][3].

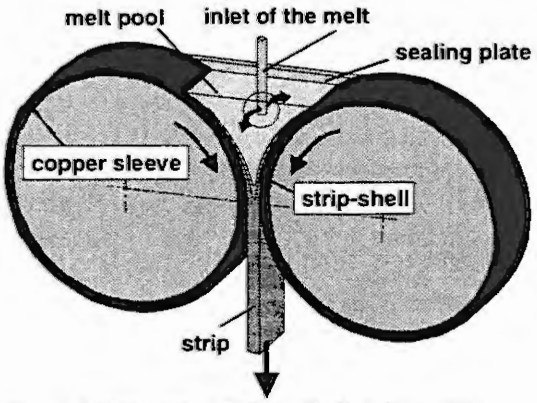


Figure 1: Principle of Twin-Roll strip casting

The considerable savings in energy and costs, the reduced amount of investment (90% of Continous Casting process) as well as numerous possibilities of adjusting technological properties, caused the interest in Twin-Roll strip casting technology [4]. Using those advantages, the strip quality(for example: the geometry) has to be established within only one process step. A reduced quality - the thickness profile being out of tolerance - can not be corrected afterwards. Additional to this requirements, the complexity of the casting process is increased, because the two process steps „casting and forming" are now linked together.

The investigation conducted at the Institute for Metal Forming is focused on testing the castability of different steel grades and improves the required quality, which is expressed in homogenous strip temperature and thickness profile over the width and the length. To fit the small tolerances, models are needed, which will also be able to perform a wide range of operating points. This problem of modeling will be discussed at an example of one specific product attribute, that appeared during few experiments.

Strip attribute - analysis of the reasons

The videoprint in Figure 2, taken just beyond the roll-gap, shows the temperature field of the produced strip over the width (150 mm) and characterizes the mentioned strip attribute. Remarkably, the strip temperature at the edges (each about one third of the width) is about 200 K higher than the center temperature.

The reason for this phenomena is tightly connected with the strip forming conditions. Low temperature stands for a higher forming grade because the longer the forming zone, the longer the time of ideal heat flow conditions and the more the strip temperature can decrease. Those inhomogeneous forming conditions could have their reasons in:

1. an inhomogeneous roll profile
   → the gap is smaller in the center of the strip,
2. an inhomogeneous thickness of the strip-shells
   → the „kissing point" is higher in the center of the strip
3. a combination of both reasons.

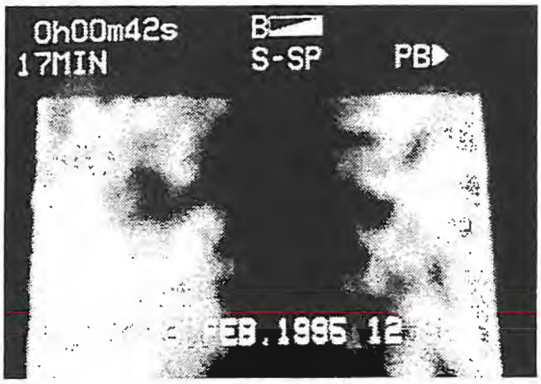


Figure 2:Videoprint of the strip - showing an inhomogeneous temperature profile over the width

The third explanation is the best representation of the real conditions but includes several difficulties considering the modeling of such an interactive process. That is the reason why the existing models reduce their systems on (1) the roll or on (2) the melt pool. Both models and their limits will be explained.

Calculation of the thermal roll crown with FEM

Leaving the radiation aside, the internally cooled rolls have to conduct the whole heatransport - including the difference between the melt and strip enthalpy as well as the solidification enthalpy, as the largest part. Considering the short time of contact, a high heat flow density is necessary. The copper sleeves of the rolls are characterized by a high temperature gradient in radial direction between the surface and the cooling plain. The resulting tension and displacement lead to a convex crown formation, so that the roll gap is reduced in the center of the width. In case of small rolls (width < 200mm) the crown can be expressed by only one value, the difference between the gap distances at the edges and the center. A negative gap-crown is characterized by an increased strip thickness in the centerline. In order to compensate the thermal crown in the steady state condition and produce flat strips, the rolls are machined with a corresponding negative profile.

The modeling of the thermal roll crown was carried out by using Finite Element Method (ABAQUS) [4]. The theoretical results of the model could only be transfered with three dimensional calculation (Figure 3). In this way it became possible to describe special static process conditions and define the qualitative dependence between thermal crown and process parameters as the thickness in Figure 4.

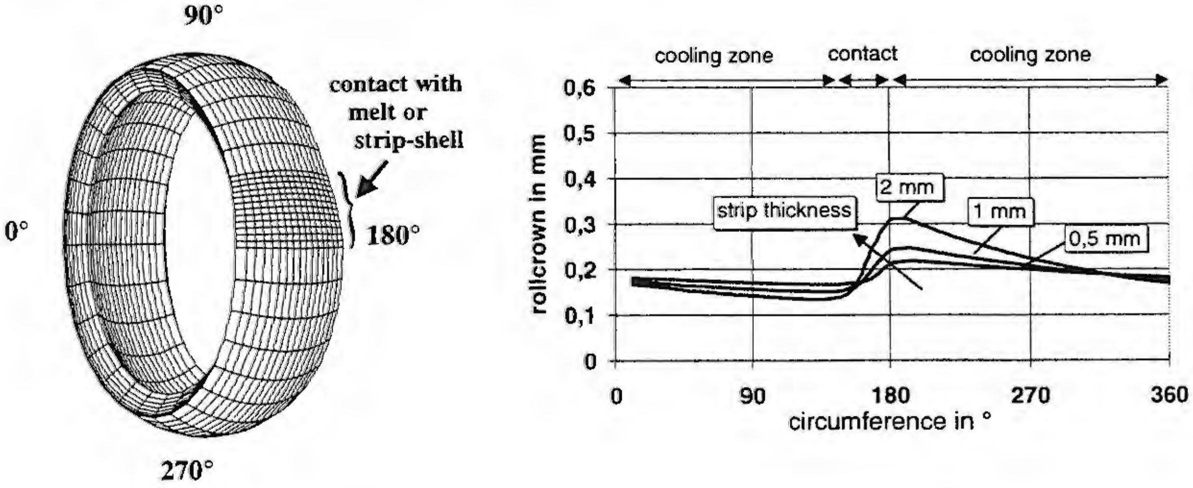


Figure 3: Three dimensional FEM calculation of the thermal roll crown (superelevated 200 times) and the dependence between roll crown, circumference and different strip thicknesses [4]

Considering a quantitative prediction of the roll profile, the suitability of the FEM-model is restrained. The reasons are:

1. Several approaches considering the boundary conditions have to be made (for example: the heat flow between the roll surface and the strip-shell was described by an average heat transfer coefficient).
2. The temperature profile itself influences the thickness profile, because it causes local roll deformations.
3. The real construction of the roll is not fully transformable into the FEM-model.
4. All those influences, which are linked together with the strip forming process itself as well as to the process control, can not be separated or can only be taken into consideration with high amount of calculation time.

Because the thermal crown is related to several of process parameters, it is difficult to predict the exact negative profile to be machined into the roll. In the mentioned attribute (Figure 2) the negative profile could not compensate the thermal expansion.

Consequence A: The temperature profile in Figure 2 can be explained only by the roll gap conditions.

Calculation of the strip-shell thickness in the melt pool

Depending on the type and configuration of the prescribed inflow parameters, the inlet of melt to the pool induces characteristic melt flows, which influence the temperature field and hence solidification and cooling parameters through the mechanism of convective and conductive heat transport. For modeling the Finite Volume Method (PHOENICS) had been used. During the development of this model it turned out, that the usage of an average heat flow coefficient as boundary condition as to be done in the roll crown FEM model, is not able to represent the temperature profile in the strip. A large number of influencing factors with a wide range of different effects take place during the contact between metallic surfaces. It was therefore necessary to couple the advance of the strip forming process with the local events at the roll/strip-shell interface [5]. Taking this into account, Figure 4 shows the supposed heat transfer coefficent in dependence upon the local contact conditions implemented in the FEM-model. The calculation of the specific inlet conditions in the experiment of Figure 2 took these boundary conditions into account. The results in Figure 5 showing the edges of the pool being good supplied with hot melt while the center remained colder.

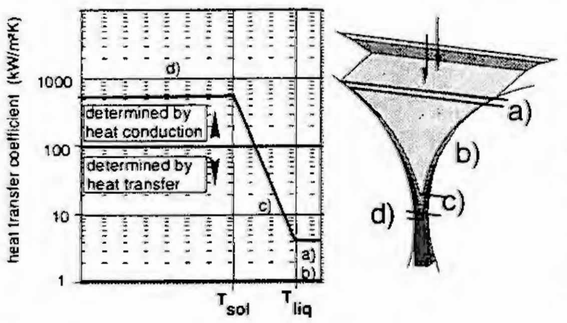
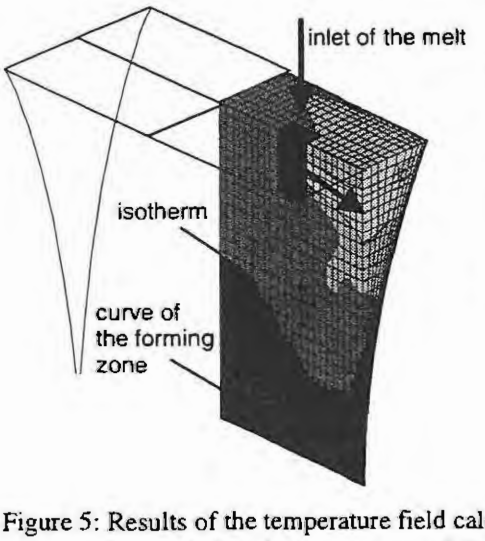


Figure 4: Heat transfer coefficient curve

This is directly related to an enlarged forming zone in the roll gap and corresponds to the strip profile in Figure 2.

Consequence B: In opposite to the FEM roll crown model this result underlines the strip attribute explained by the strip-shell formation profile.



Of course this is again a qualitative explaination, because several approximations have been made:

1. No determination of the local conditions over the contactlength, especially if the material or the process conditions will change.
2. Only approximated description of the flow conditions in the semi-solid state.
3. Only representing the static condition (no pool filling or change of the pool level).
4. No calculation of the surface radiance and turbulences.
5. No calculation of the heat flow through the ceramic sealing plates.

Figure 5: Results of the temperature field calculation in the pool [5] (presentation shows a quarter of the pool)

Reasons for using Neural Networks to model the strip thickness profile

The different estimations turned out, that a quantitative prediction of the strip thickness profile has to take both models into consideration. Such a „Mega numerical Model" is not only limited by calculation performance, but also by the determination of the boundary conditions:

1. The solidification morphology of different materials and operating points, causes different heat transfer conditions.
2. The determination of the forming zone remains difficult.
3. The mechanical deformation of the rolls is directly linked with the extension of the forming zone and not defined yet.
4. The knowlede of the high temperature yield stress has to be defined.

The following arguments caused the motivation to investigate in modeling the strip thickness profile based on the measured process data:

- If experiments with a high amount of costs are necessary to adapt a numerical model to the experimental results, the process data itself could be used for modeling.
- Simulations with a model, which is developed out of process data, should have a better correspondence with the real process than an adapted one.
- Any further experiments will improve the models status and the quantified prognosis.
- The interaction between the roll gap and strip-shell profile will include in the model automatically.
- The expence in developing a new model for a changed geometry is rather small.

Preparing the data for the process data based modeling with Neural Networks

The primary task by using process data based modeling (PBM) successfully is not the calculation - for example with Neural Networks - but the data preparation. The main question in this context is to find all influencing parameters, find a way to measure them and bring them together in the causal relation. As methodical approach the general off-line phase modeling [6] in Figure 6 was used. After dividing the hole production chain into seperat phases (Pi, Pii) with their processes and outgoing properties, the inner process „strip-forming" was analysed in detail. The real process (middle branch) is going to be modeled by conventional estimations (right branch) to find the relevant parameters with a high range relevance (input) to the thickness profile and supplied by the process data based model (left branch) to improve the quantitative prognosis.

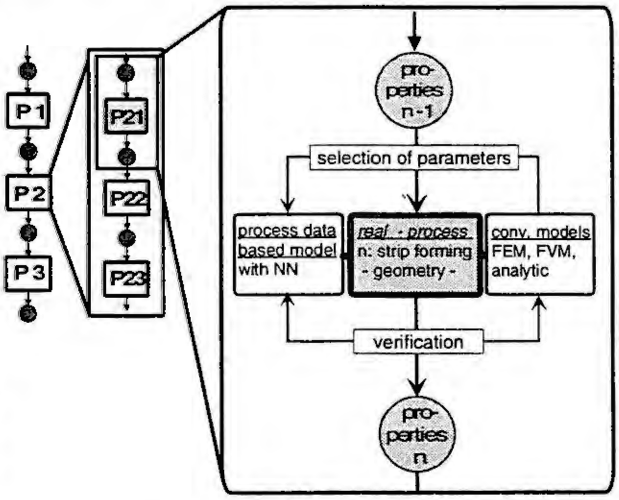

P1
P2
P3
P21
P22
P23
pro-perties n-1
selection of parameters
process data based model with NN
real - process n: strip forming - geometry -
conv. models FEM, FVM, analytic
verification
pro-perties n

Figure 6: Phase modeling of the Thin-Strip production

The NN-model predicts the strip thickness profile (output) and can be verified by the other models, because the simulations can be compared with the conventional ones. By this way the influencing parameters for Neural Network training were found and prepared (Figure 7). All those process data were selected out of the measured static data during two experiments. Parameters which describe the flow condition remained constant, to reduce the dimension of the data matrix. All together 60 datasets had been put into relation with each other and the strip crown results.

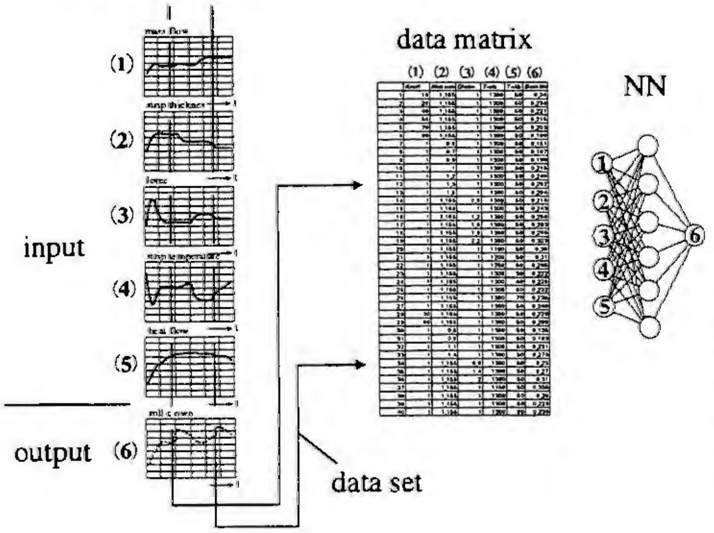


Figure 7: Method to create the NN-model

About 30 % of the data matrix were isolated for testing the net. As net architecture a Multilayer-Perceptron (MLP) with three layers was chosen. The number of hidden neurons had been optimized concerning the average error of the test data and the correlation between measured and predicted data. Instead of the common Backpropagation-Algorithm the efficient Feedforward-Algorithm of [7] was used. To provide the training from the effect of overlearning, the number of outer iterations had been limited.

Simulations of the strip thickness profile with Neural Network

The optimized net, trained with measured data, could not provide an average error of 7 % of the scale. Even though the correlation was high, a successful training of the connected relations could be mentioned.

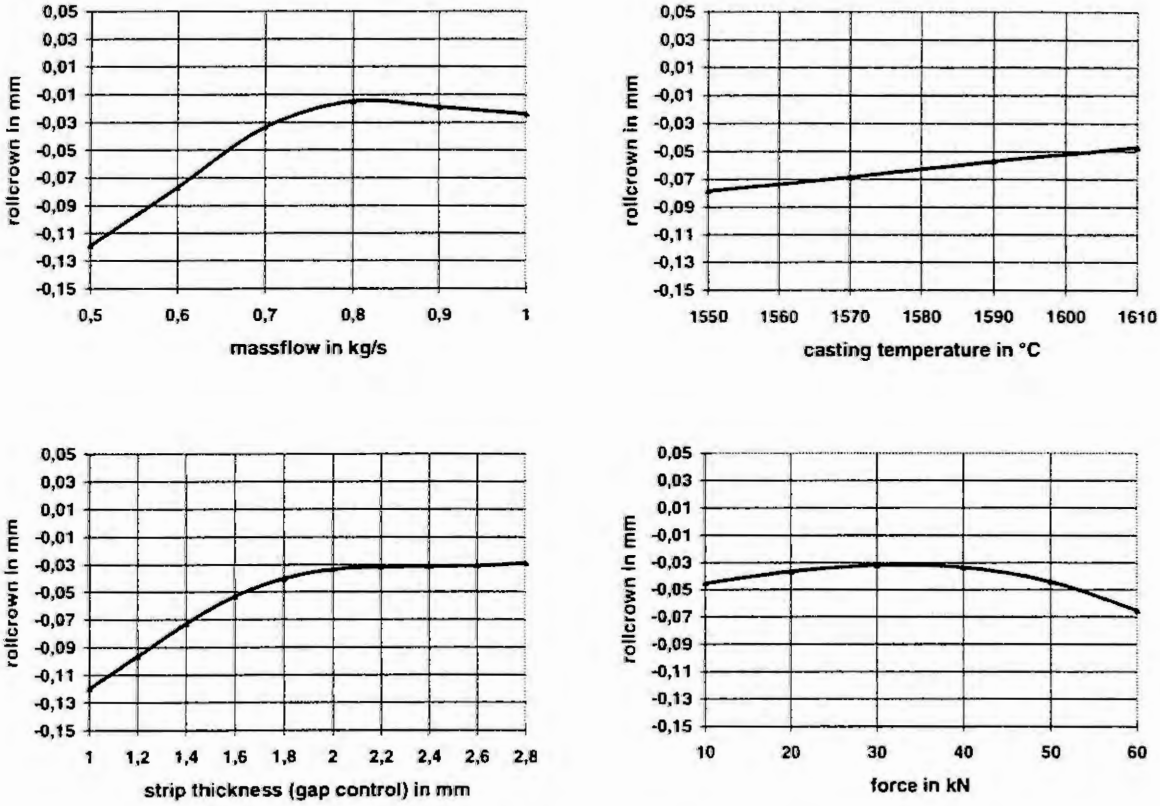


Figure 8: Results of the NN-prognosis of the roll crown by variation of different influencing parameters (operating point: massflow: 0.7 kg/s; thickness: 2 mm; casting temp.: 1600 °C; force: 40 kN; dT/dF: -4K/kN; heat flow: 180 kW)

Simulations have been carried out in a selected operating point and with the variation of one input parameter. In Figure 8 the results of four prognosis are shown. To interpret these results physically and compare them with the results of the FEM-model, it must be noticed, that the roll was machined with a concave profile, which ideally equalizes the thermal roll crown. Both experiments had a positive strip crown (negative roll crown). The FEM-model calculates linear increasing between roll crown and different parameters. Only the proportional factor differs with the parameter because of the range of influence.

In the NN-prognosis only the overheat follows a monotone increasing curve. In case of low values, the parameters mass flow, thickness and force underline this linear tendency as well as the absolute range. But in a higher range of those parameters the predicted rollcrown remain an unexpected constant or decreasing tendency. The analysis of both experiments with the video of the pool and the outcoming strip delivered the explanation for these effects. In both experiments comparable steel grades had been cast. The thickness range differs (exper. A: 2-2,8 mm; exper. B: 1-2 mm) while the inlet of the melt (as in Figure 1) is principally the same (in exper. B a higher intensity in the flow to the sealing plates).

Those experiments do not show the attribute in Figure 3 for the whole time. In the beginning of the process the concave profile of the roll compensates the strip-shell profile in Figure 6, so that the temperature profile remains homogeneous, as Figure 9 proves. After some seconds the thermal crown is growing steadily until a point of change is reached. This is when the roll profile is going to be flat and not able to equalize the inhomogeneous strip-shell profile. By the change of the strip temperature profile - becoming a colder centerline (Figure 2) - the specific forming force increases. That means in the case of an increasing thermal roll crown (higher mass flow, higher thickness) the local elastic deformation is intensified as a further consequence. The higher the force the more the deformation predominates in this context. This effect is also connected with lower integral heat flow into the rolls and a reduced thermal roll crown. Both explanations underline the NN-prediction.

In the beginning of this paper the interaction between the process parameters was figured out. The neglection of this complexity leads to low ability of transformation by conventional modeling. The process data trained Neural Network found these interactions automatically.

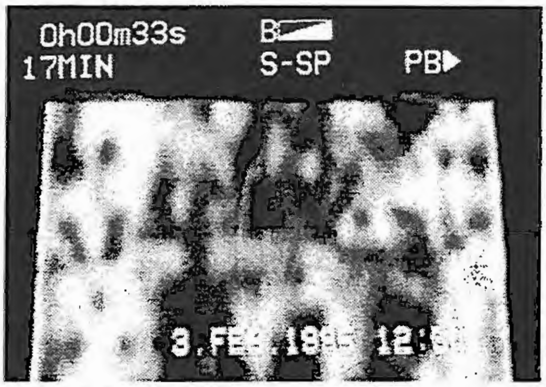


Figure 9: Videoprint of a strip showing a more homogeneous temperature profile over the width - before the point of change

Summary

The quality of Twin-Roll strip casting products is mainly influenced by the interaction between the temperature and the thickness profile. Conventional numerical models (FEM and FVM) explain either reasons for the temperature field or the thickness profile. A quantitative prediction of each quality aspect is limited, because the boundary conditions between the two subsystems (roll and pool) are not able to determine.

Taking the link of different process parameters into consideration, the modeling of the strip thickness profile with Neural Networks has been investigated. The procedure in selecting the influencing parameters, the formation of the data sets and the results of some simulations have been discussed. The prognosis of the NN led to sufficient quantitative results and was able to realise the special characteristics of the investigated experiments, which are based on the interaction between temperature and thickness profile.

In selecting the influencing parameters and in explaining the NN simulations, the numerical models were such a good supplement, that with regard to other complex processes a big potential can be presumed.

Literature

[1] Nippon Steel baut erste Twin-Roll-Gießanlage, Stahlindustrie in der Welt, Stahl und Eisen 116, Nr.3, Verlag Stahleisen, Düsseldorf 1996

[2] Hendricks, C.: Bandgießtechnik - eine Revolution in der Stahlindustrie? Vortrag auf dem 9. Aachener Stahlkolloquium 1994, Stahl und Eisen 115 Nr.3, Düsseldorf 1995

[3] Bernhard, S.: Zur Regelung und Steuerung von Bandgießanlagen nach dem Zwei-Rollen-Verfahren; Forschungsberichte VDI, Reihe 8, Nr. 365, VDI-Verlag, Düsseldorf 1994

[4] Szekely, J.; Trapada, G.: Zukunftsperspektiven für neue Technologien in der Stahlindustrie, Vortrag auf dem Metec-Kongreß in Düsseldorf 1994, Stahl und Eisen 114 Nr.9, Düsseldorf 1994

[5] Beyer-Steinhauer H.: Thermische Walzenbombierung beim Zwei-Rollen-Gießwalzverfahren, Dissertation an der RWTH, Umformtechnische Schriften Band 39, Verlag Stahleisen, Düsseldorf 1993

[6] Jestrabek, J.: Stahlbandherstellung nach dem Zweirollenverfahren - Modellierung des Strömungs- und Temperaturfeldes, Dissertation an der RWTH Aachen, Umformtechnische Schriften Band 59, Verlag Stahleisen, Düsseldorf 1995

[7] Bärmann, F.; Biegler-König, F.: On a Class of Efficient Learning Algorithms for Neural Networks, Neural Networks, Vol.5, pp.139-144, Pergamon Press, 1992

ON-LINE PROCESS MODELING AND OPTIMISATION FOR A 20HIGH COLD ROLLING MILL

A. Schneider [1] and R. Werners [2]
[2] Mannesmann Demag Hüttentechnik MDS Walzwerktechnik
Daniel-Goldbach-Straße 17-19, D-40880 Ratingen
[1] Lehrstuhl für Automatisierungstechnik, BUGH Wuppertal
Fuhlrottstraße 10, D-42097 Wuppertal

Abstract : The paper describes the basic principles of a model formation used for the optimisation of mill utilisation and presetting of actuators in a 20high cold rolling mill. Besides the special requirements placed to the force and torque calculation model in a Sendzimir mill arrangement [1], the paper in particular describes the model effort for the description of the mill stand behaviour and the interaction with the strip. Finally, the chosen concept for the on-line implementation in a process control computer is shown.

Introduction

Product quality such as thickness, flatness and surface as well as mill utilisation and operation flexibility are ever increasing demands placed on cold rolling mill operators. However in case of complex mill arrangements, the presetting of the available mill actuators to achieve the desired product quality represents an extremely difficult task. This results from the great number of variable process parameters and actuators to be considered. Figure 1 gives an overview about the mill arrangement and main technological actuators.

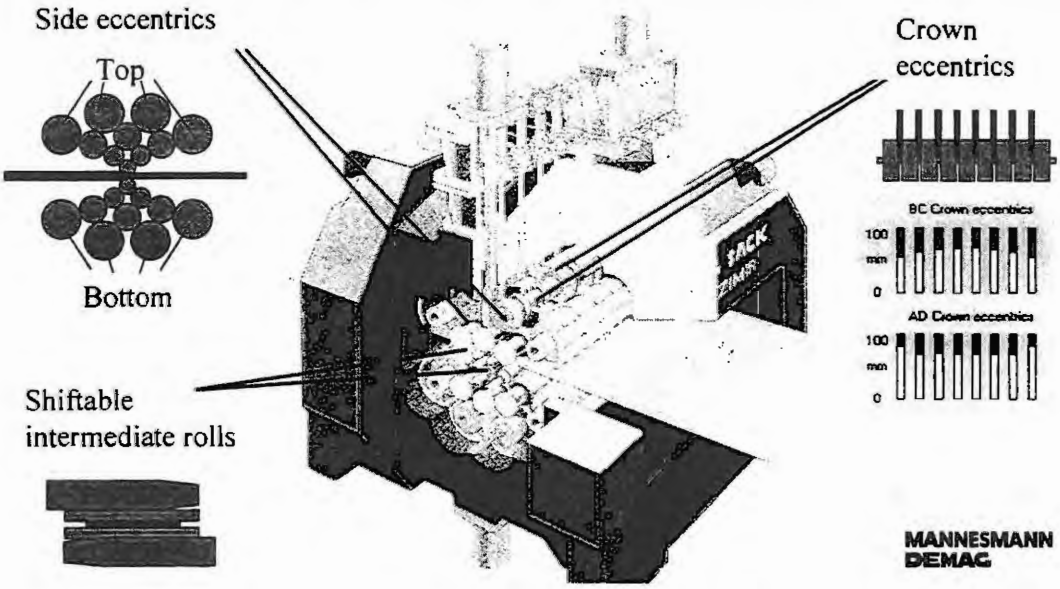


Figure 1 : Main technological actuators in a Sendzimir mill

So-called side eccentrics at each of the backup rolls are used to adjust the position of the corresponding roll axis in a wide range and by this indirectly serve for adjusting the roll gap. The side eccentrics at the lower backup rolls mainly serve for maintaining the mill pass line whereas those at the upper backup rolls are used to achieve the desired strip thickness. Side eccentrics appear to be mechanically or electrically coupled.

Crown eccentrics, available at several locations over the barrel length on all upper backup rolls, serve for specific roll axis contours which affects the roll bite geometry respectively. This is of particular importance for matching the roll gap contour to the profile of the strip entering the roll gap and for achievement of a certain desired strip flatness. The shiftable first intermediate rolls represent further contour actuators, which however mainly serve the modification of the roll gap contour in strip edge area.

Measurements of the geometrical relations in the roll stack are available at the backup rolls and first intermediate rolls. The shifting positions of the first intermediate rolls are measured directly. Information about the backup roll axis contour is based on rotation angle measurements at each side and crown eccentric. Together with the mill spring, for which knowledge about the roll separating force is needed, this then leads to the position of each backup roll axis over the barrel length. The roll separating force however is in most applications only measured indirectly through the adjustment pressure needed for certain side eccentrics. Apart from hysteresis also the relation to the geometry in the mill stand causes this indirect measurement to be usually insufficient.

Due to the complex mill arrangement, the reproducibility of the final product quality and the optimum usage of available mill resources to increase productivity represents an extremely difficult task, which can only be fullfilled with a comprehensive model approach, which takes all relevant mill and process parameter into consideration. Due to the fact, that neither direct geometrical information nor accurate roll separating force measurements exist, a variety of individual mathematical models is needed to describe the complex elastic mill stand behaviour and the elastic plastic characteristic of the material to be rolled.

Roll gap analysis for force, power and slip determination

The roll separating force, torque and power brought about by forming of the material in the roll gap are some of the most important process information. Only if exact knowledge about these parameter exists, accurate presetting of the mill actuators and optimum mill utilisation can be obtained.

Because reliable roll separating force measurements are missing in most Sendzimir mills, special analysis effort is required to describe the different stress components in the roll gap. Those can be broken down into vertical and tangential stresses acting on the work rolls. The sum of the vertical stress components leads to the roll separating force, whereas the sum of the tangential stress components leads to the roll torque and thus to the main drive power. Figure 2 (left part) shows exemplary the conditions in the roll bite including those tangential and vertical stress components.

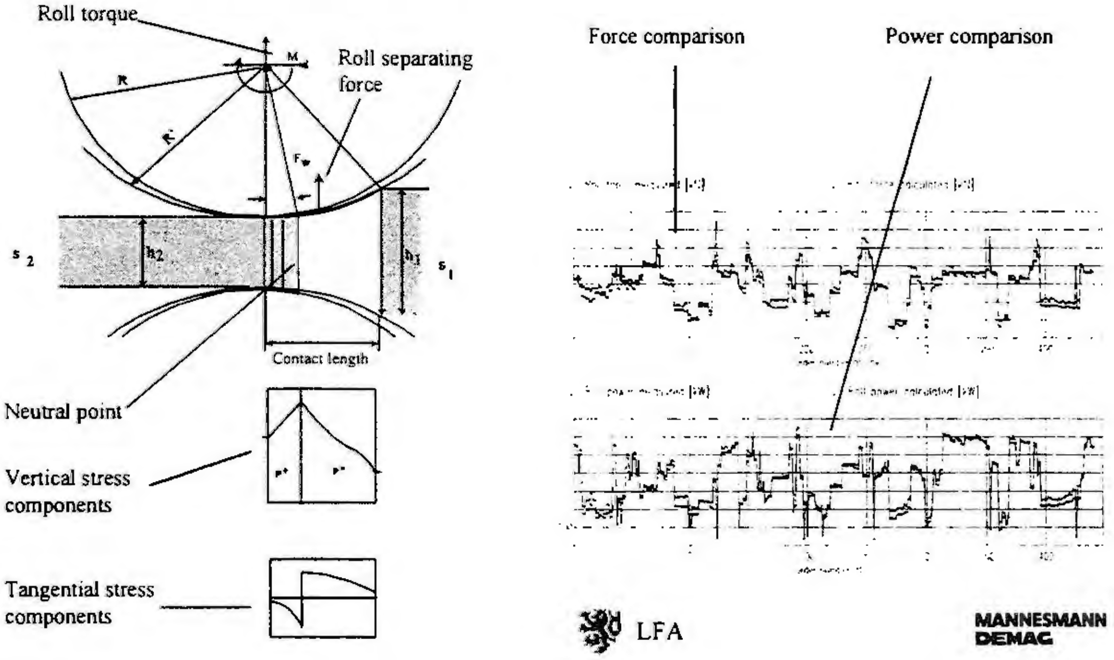


Figure 2 : Stress components (left) and model evaluation in a 4high mill (right)

A model approach, which simultaneously provides accurate information about the vertical and tangential stress components opens up capabilities of detecting forces based on the main drive power, providing that friction and drive efficiency are known. This in particular enables material yield stress adaption even in case of missing or unreliable roll separating force measurements. Material yield stress adaption is required in any case, where high flexibility in terms of the alloy spectrum to be rolled is needed.

In particular the demand for optimisation of the strip surface requires an indication about the slip present in the roll gap. By minimising the forward slip, an improvement of the strip surface can be obtained. The slippage present in the roll bite is correlated to the position of the neutral point (figure 2 - left part), which is affected to a certain extend by the entry and exit tension applied. Entry and exit tension thus in small ranges serve as an actuator for the optimisation of the strip surface. To provide accurate presetting, the gap analysis model has also to deliver accurate information about the neutral point position.

The mathematical approach chosen here is a strip fibre model (Lippmann, Mahrenholtz [2]), the basics theory of which has been described by v. Karman [3] and Siebel [4,5]. The strip in the roll gap is divided into several fibres for which the individual forming parameters are determined. The yield stress of the material is herein described with the aid of an equation introduced by Spittel [6]. The coefficients necessary for this have been determined by torsion bar trials and were collected to an extensive material data base [7]. This then leads to detailed information about the vertical and tangential stress components and in particular also about the slip present.

A comparison of measured and calculated roll separating forces and drive powers taken in a 4high mill, where direct measurements are available, is shown in figure 2 (right part). The comparison performed for a great number of different coils with different geometry and chemistry clearly shows the simultaneously high accuracy of force and power calculation.

Model for description of elastic mill stand behaviour

It is of importance to describe all the several effects present in the mill stand in their entirety in order to allow propagation from the measured eccentric adjustments down to the roll bite contour, which is the target for further optimisation steps.

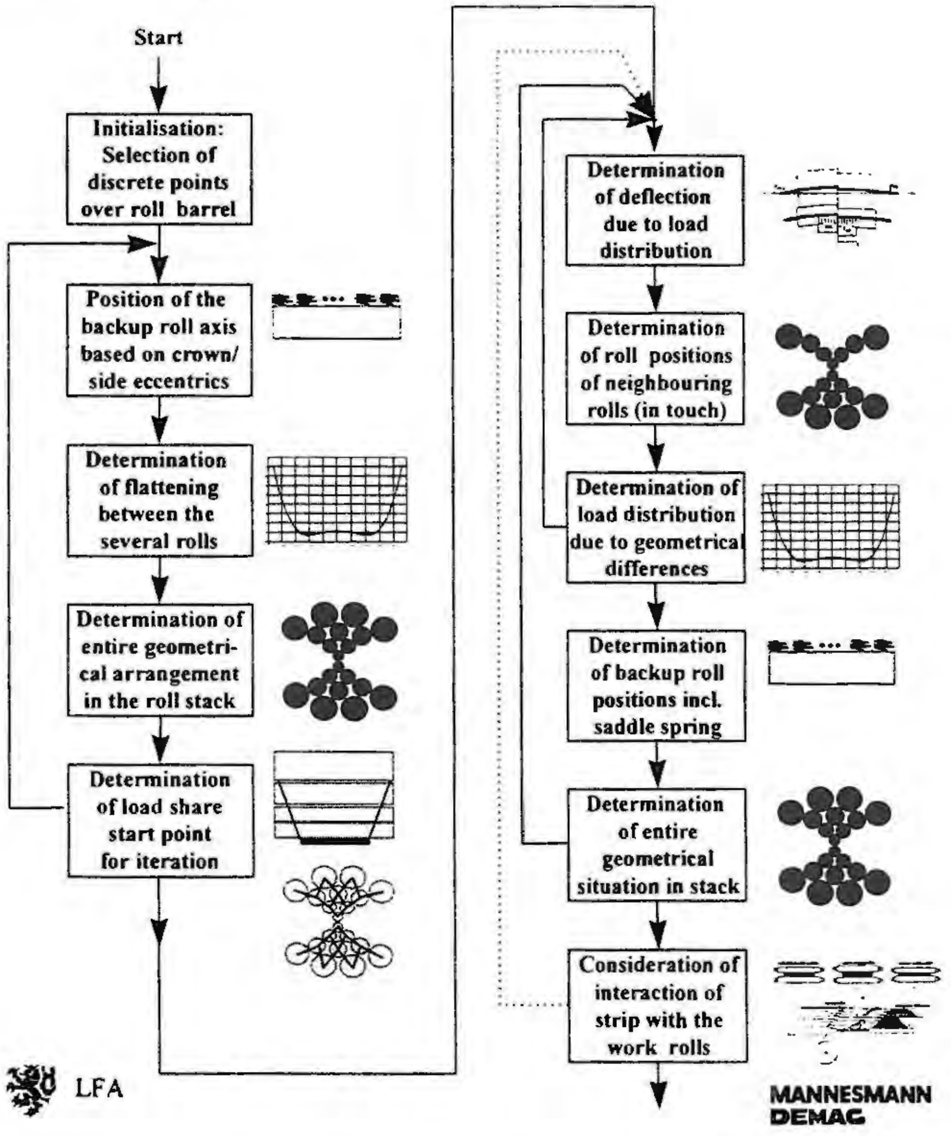

Start
Initialisation: Selection of discrete points over roll barrel
Position of the backup roll axis based on crown/ side eccentrics
Determination of flattening between the several rolls
Determination of entire geometrical arrangement in the roll stack
Determination of load share start point for iteration
Determination of deflection due to load distribution
Determination of roll positions of neighbouring rolls (in touch)
Determination of load distribution due to geometrical differences
Determination of backup roll positions incl. saddle spring
Determination of entire geometrical situation in stack
Consideration of interaction of strip with the work rolls
LFA
MANNESMANN DEMAG

Figure 3 : General concept of model for description of the elastic mill stand behaviour

One effect considered here is the mill spring, which appears to be a position change of the saddle segments. Furthermore the flattening between the several rolls and likewise between the strip and the work rolls as well as the deflection of the several rolls is considered.

Another relevant effect is the thermal expansion of the work rolls brought about by the forming energy. A model has been implemented, which is a combination of an analytical approach in radial roll direction and a finite difference approach over the barrel of the roll [8]. Due to the complexity of this calculation itself, it is not discussed within this paper any further. The results of the thermal crown model are taken into consideration in the elastic mill stand model as offsets to the contour of the work rolls.

One requirement placed onto the elastic mill stand model was the ability to cover a variety of different mill configurations, different roll grindings (including the dynamically changing thermal crowns of the work rolls) and different roll material characteristics even broken down to the individual rolls in the stack to be able to cover also process situations, where unusual roll combinations are chosen. Therefore the description of the elastic mill stand behaviour is based on a numerical solution approach, where the different parameter can be considered easily. The different effects, such as the flattening between the rolls and likewise between the strip and the work rolls as well as the deflection of the several rolls are derived from multiple iterations. A general overview about the model concept is given in figure 3.

The elastic mill stand model can generally be divided into two parts. An initialisation part (refer to figure 3 - left part) cares on the fast determination of an image about the load share in the roll stack considering flattening and mill spring however neglecting the roll deflection. The deflection is then considered in a more time consuming final part (refer to figure 3 - right part).

Based on the roll separating force, the model first provides a rough image about the load share in the entire roll stack (refer to figure 3 - left part). This initial load share determination already takes mill housing spring effects, which are represented at the locations of the saddle segments as well as flattening effects between the rolls into account. The spring modulus of the mill housing valid at the locations of the saddle segments have initially been determined by FEM calculations and are adapted on-line.

The initial load share determined is taken in the second step as a basis for the iterative determination of the interaction between load distribution, flattening and deflection. The deflection of each roll is derived from the currently determined load distribution. The geometrical differences between two neighbouring rolls are then interpreted as flattening and thus lead to a new load distribution along the contact curve, which finally leads to a new deflection. This iteration is performed until a solution has been reached, where the entire load, the deflection and the flattening of the several rolls match with each other.

Interaction of the strip with the rolls

After the roll gap contour has been determined for a certain load situation in the roll bite, the interaction of the strip with the work rolls is investigated. This is done with the aid of an analytical approach [9,10], which is the solution of the differential equation in the current setpoint, the basic principles of which are explained in the following.

As long as the roll gap contour matches the incoming strip thickness profile, the thickness reduction appears to be the same throughout the width. Therefore no elongation differences, which are also referred to as flatness errors, are produced (refer to figure 4 - item 1).

Roll gap contours, which differ from the strip thickness profile entering the roll bite, cause local differences in the thickness reduction over the width of the strip (refer to figure 4 - item 2 and 3). The different thickness reductions for their part cause differences in the elongation over the width of the strip, so that certain flatness errors are produced.

Differences in the elongation over the width also cause changes in the local tension applied, which then cause changes in the local pressure. These circumstances apply likewise to flatness errors already approaching the roll bite, which produce local variance in the entry side strip tension. The reduction differences brought about by unmatched roll gap contours also produce local variance in the roll pressure.

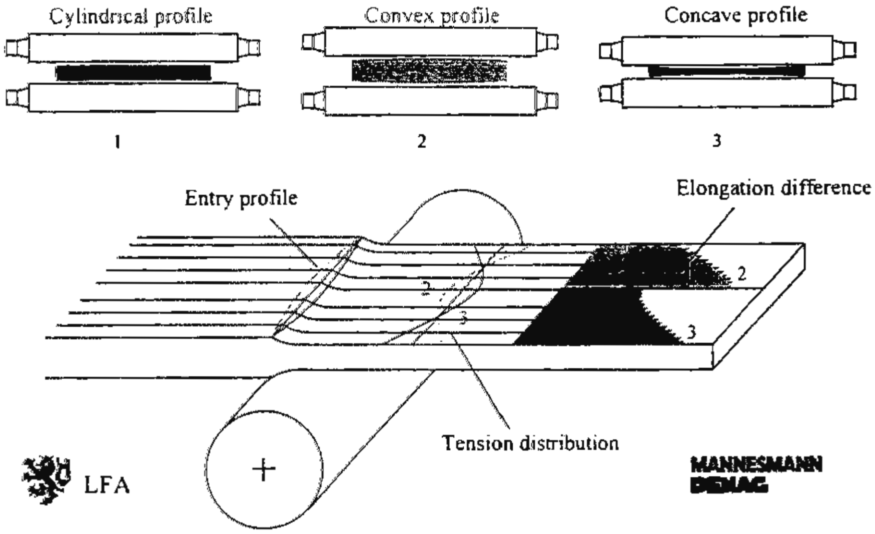


Figure 4 : Thickness profile changes in the roll gap

Variance in the local roll pressure causes changes in the local flattening of the strip into the work rolls and furthermore also in the entire load share and deflection within the roll stack. Due to these circumstances, the load share in the roll stack is affected and by this also the spring, deflection and flattening effects, so that a new investigation of the entire elastic mill stand behaviour is required (refer to figure 3).

Whereas the described model concept comprising roll separating force calculation, investigation of elastic mill stand behaviour and interaction with the strip provides the required flexibility in terms of process parameter to be considered, it is as well time consuming. Apart from the individual models shown here, also optimisation algorithm form part of the entire system, which require additional computation time. Within the process control computer however, the time frame available for set-up calculations is usually to be kept as short as even possible in order to avoid production delays. Therefore the model concept described requires either adequate hardware or special implementation methods as follows.

On-line implementation concept

The tasks fulfilled by the mathematical models and their dedicated optimisation algorithm running on-line on the process control computer can generally be broken down into the following main items:

- optimisation of pass reduction schedule for the whole coil in advance to rolling
- optimisation of pass specific adjustments (eccentrics, shifting, speed, tension ...)
- continuous observation of thermal roll crown and of main and coiler drive status and capabilities
- short and long term adaption

Whereas the determination of the pass reduction schedule is required prior to start of the first pass of a coil, the optimisation of pass specific parameter can only be performed accurately if the actual mill situation is known. This is the case before start of the concerned pass. Thus the several optimisation tasks can be broken down into those needed before start of manufacturing of a coil and those needed prior to each individual pass. In particular the optimisation of the pass reduction schedule is a time consuming functionality due to the fact, that optimisation calculations have to be performed for all planned passes.

These circumstances enable a split of the optimisation tasks, where the time consuming pass schedule determination is performed far in advance to start rolling in a so-called preliminary set-up module (refer to figure 5 - SETPRE) and the less time extensive pass specific optimisations are performed shortly before start of each individual pass in a separate dedicated pass specific module (see figure 5 - SETUP).

Main and coiler drive status and in particular the thermal expansion of the work rolls change dynamically throughout rolling of a pass and even during mill delay times and thus need also to be continuously investigated. This is performed using a cyclically started module (refer to figure 5 - CYCLIC), which is continuously provided with actual process information rather than with snapshots or summary data only.

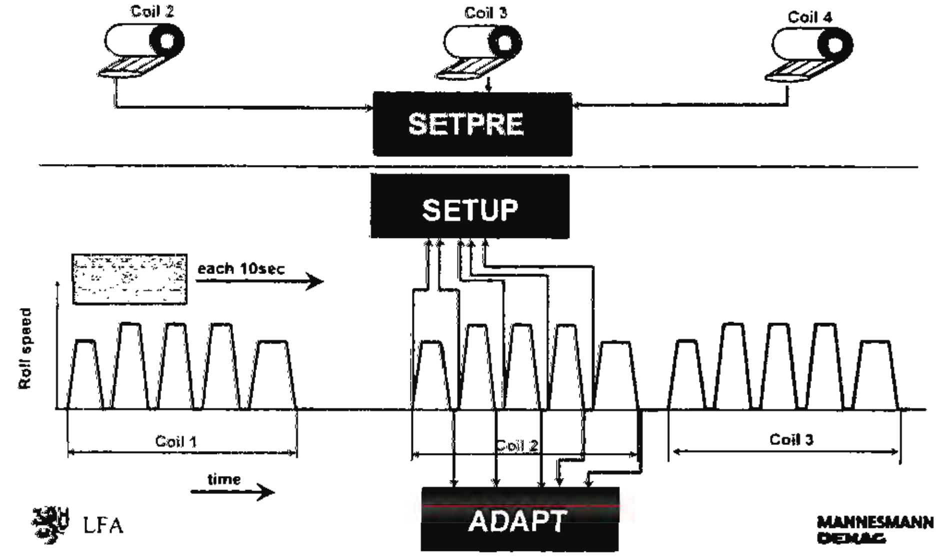


Figure 5 : Principle of task sharing in on-line application

Due to the fact, that the amount of reliable measured process information is usually not adequate at a single snapshot, long term adaption calculations can only be performed taking a significant amount of historical data into account. The amount of information to be considered however causes these investigations to be time consuming too, so that another independent process has been foreseen, which operates after completion of a pass or coil (refer to figure 5 - ADAPT) independent from all other tasks.

Conclusion

The complexity of the mathematical models used to entirely describe and optimise the rolling process provides the capability of taking a great amount of variable process parameter into account. This ensures the process control computer system to be flexible and open for various applications and situations however causes the model calculations to be time consuming.

Due to a special on-line implementation concept, in which the several tasks are broken down to the point in time where they are needed, even standard hardware and operating systems can be used.

References

1. Sendzimir, M. G.: The Sendzimir Cold Strip Mill. Journal of Metals 8 (1956) 9, S. 1154-1158
2. Lippmann, H. and Mahrenholtz, O.: Plastomechanik der Umformung metallischer Werkstoffe. Springer-Verlag, Berlin 1967
3. Karman, T. v.: Beitrag zur Theorie des Walzvorganges. Zeitschrift für angewandte Mathematik und Mechanik 5 (1925), S. 139-141
4. Siebel, E. : Die Formgebung im Bildsamen Zustande. Verlag Stahleisen, Düsseldorf 1932
5. Siebel, E. : Kräfte und Materialfluß bei der bildsamen Formgebung. Stahl u. Eisen, 45 (1925) 37, S. 1563-1566
6. Hensel, A. and Spittel, T. : Kraft- und Arbeitsbedarf bildsamer Formgebungsverfahren. VEB Deutscher Verlag für Grundstoffindustrie, Leipzig 1978
7. Spittel, M., Spittel, T., Teichert, H. and Skoda-Dopp, U. : Material Data Base for Steel and Non-Ferrous Metals, Int. Report of Mannesmann Demag MDS Walzwerktechnik, June 1994
8. Sauer, W. : Thermal Modelling and Control Strategies in "Advances in Aluminium Rolling" Int. Conference of Mannesmann Demag MDS Walzwerktechnik, May 1993, S. 89-95
9. Mielke, R. : Regelung des Bandlaufs in einer Warmbreitbandstraße. Dissertation an der BUGH Wuppertal, Lehrstuhl f. Automatisierungstechnik, 1992
10. Schneider, A., Kern P. and Steffens, M. : Model supported Profile and Flatness Control Systems. 49° Congresso Internacional de Tecnologia Metalúrgica e de Materiais - International Conference 09.-14.10.1994 in São Paulo (Brasil), Vol. 6 S. 49/60

INTEGRATION OF MODEL-BASED METHODS IN PROCESS CONTROL SYSTEMS

Dirk Meyer
University of Technology Aachen
Chair of Process Control Engineering
D-52056 Aachen

Abstract. The new generation of distributed control systems (DCS) is characterised not only by increasing performance but also by a tendency towards an open system structure. This allows the use of commercial hardware components at the operator level, such as PCs and workstations. These platforms provide an ideal environment for the integration of model-based methods. While this is true in principle, there are no standards concerning the structure of a system in which model-based methods are performed. In this paper an adequate architecture of such a system is proposed.

Introduction.

According to Polke [1], p. 422, "process control takes in all measures that effect the desired course of a process in the sense of established objectives." One part of the functions required for the performance of this task is executed "automatically" by the process monitoring and control hardware (the process control system) and the other one "manually" by a human operator. The degree of automation [1], p. 218, is the fraction of all functions that can be represented by automatic functions, see Fig. 1.

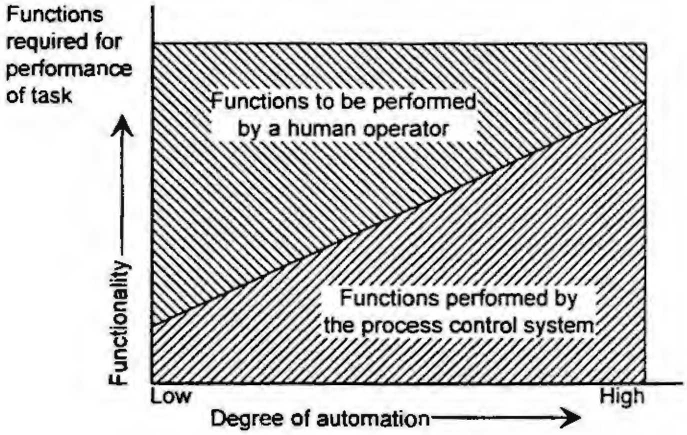

Functions required for performance of task
Functionality
Functions to be performed by a human operator
Functions performed by the process control system
Low
High
Degree of automation

Figure 1: Degree of Automation.

The degree of automation can be increased if additional knowledge about the process is made available [2]. A form of knowledge suitable for automation are mathematical models describing the behaviour of a dynamic process. Methods using such models are called "model-based methods" and offer possibilities for the improvement of the automation of a process [3]. Examples for the most commonly used model-based methods are:

- dynamic simulation,
- parameter estimation,
- state observer,
- predictive control,
- model-based optimisation,
- model-based diagnosis,
- model-based validation,
- forecasting methods.

In order to integrate these model-based methods in a process control system, some essential system elements are required (chapter 1 through 4). The object-oriented structures and the communication system proposed in these chapters are certainly of great help when integrating model-based methods in process control systems.

While there are still open questions concerning model-based methods, e.g. the numerical integration scheme, chapter 5 focuses on a problem which has been neglected so far when applying model-based methods in a process control system: A common scheme how to handle model-based methods and their results is of great importance for the operator who is involved in the process control (compare Fig. 1).

1 System Structure

A distributed control system (DCS) consists of individual components connected to a system bus, see Fig. 2. Typically, there are two different kinds of system components, the "local components (LC)" (components close to the process) and the "remote components (RC)" (central components). The "local components" contain the I/O devices. These are interfaces to the sensors which gather information from the process and to the actuators which act on the process. In the local components control and monitoring tasks are performed independently from each other. Based on a consistent information model throughout the system, the entire process can be controlled and monitored by the central "remote components". Operator functions, engineering functions, logging functions and reporting functions are realised in these components.

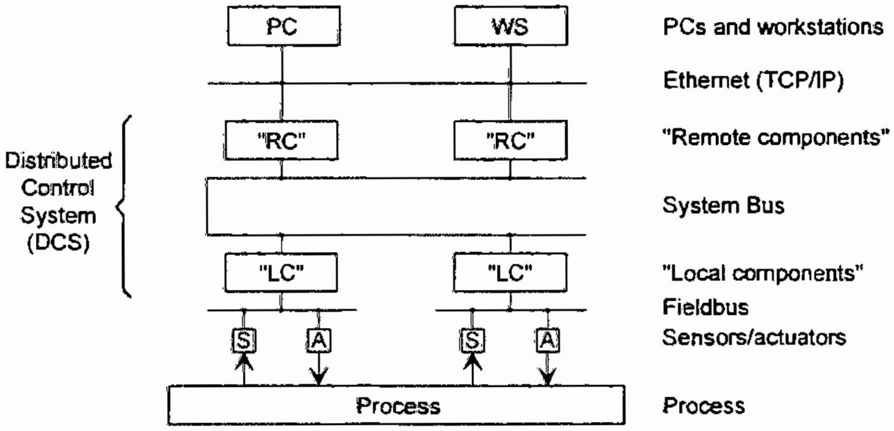


Figure 2: Open system structure of a Distributed Control System (DCS).

The modern generation of process control systems is characterised by an open system structure [4], which means that it is possible to connect commercial hardware components such as PCs and workstations to this system by an Ethernet (TCP/IP protocol). This external computer network provides an ideal environment for the integration of functions which are not covered by the functionality of the process control system itself, e.g. high level functions required for the application of model-based methods.

Unfortunately there are no standards for the system architecture of the external PC/workstation network. Many applications of model-based methods have successfully been implemented in this network environment so far, but usually these solutions have been function-oriented. Thus, these individual solutions do neither have a common mechanism of communication in the network nor a common data management. A standard system is desirable, because in such a system one does not have to solve problems with the handling of the model-based method and the data it produces but can concentrate on the problems with the model or the method itself.

In order to find the essential elements for the application of model-based methods in a process control system, an object-oriented view is very helpful: input data is processed to output data by the model-based method, which can be regarded as a "black box". A communication system is necessary in order to connect this "black box" to the "data sources" and "data sinks". There are different types of data, e.g. on-line data from the process control system, off-line data or parameters. Furthermore, an archive for all kinds of data involved in the application of model-based methods is necessary. The operator manages this system by an user interface which allows to access the data (viewing and editing), to link it to a model-based method and to execute the method itself. Thus, the essential system elements are:

- a communication system,
- a subsystem performing the model-based method,
- a data base (archive) and
- the user interface.

The structure of a system comprising these elements is shown in Fig. 3. It has to be pointed out, that this scheme shows virtual function modules which are not necessarily identical with hardware components.

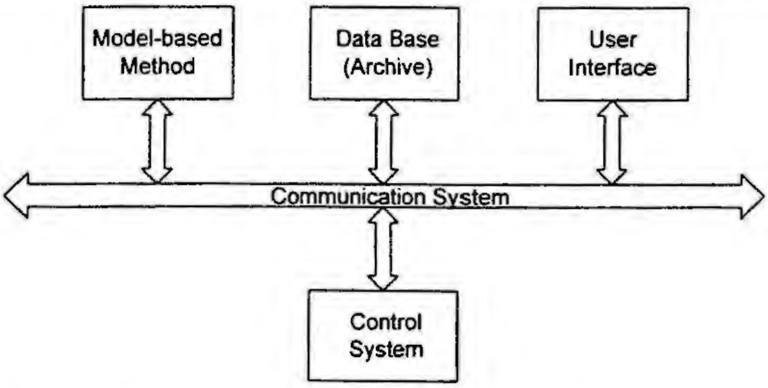


Figure 3: System structure for the integration of model-based methods.

2 Communication System

A communication system suitable for connecting functional modules as shown in Fig. 3 in a distributed system is ACPLT/KS, which is an open standard of the Chair of Process Control Engineering, University of Technology, Aachen/Germany [5]. If an application needs to communicate with another application in the network, its ACPLT/KS client contacts the ACPLT/KS server of the other application and then uses the communication services provided by ACPLT/KS. In order to carry out the communication over the network, a TCP/IP-based protocol is available for Unix-, VMS- and Windows-based PCs and workstations.

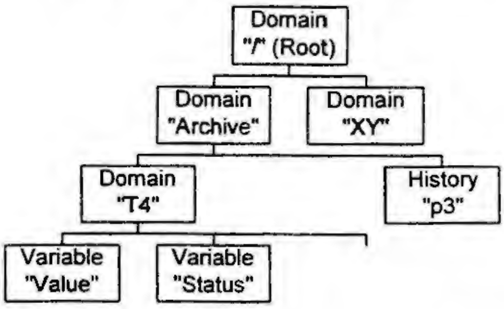


Figure 4: Hierarchical structure of the communication objects in ACPLT/KS.

The data to be exchanged is mapped to three kinds of communication objects, which are realised in an hierarchical structure (tree structure, Fig. 4):

- domains, which are containers for their children objects,
- variables, e.g. process values or parameters and
- histories, e.g. signals (time series).

The services provided by ACPLT/KS are

- object services, which allow the handling of the hierarchical communication object structure and
- variable and history services (read, write, synchronised data exchange).

The history services are currently defined.

3 Function Block System

When programming the algorithms of model-based methods, one can allow the input and output data to be manipulated by different procedures. In this procedure-oriented technology, the different procedures modules have to communicate through shared data modules and there is no protection of access to the data areas, see [1], pp. 241-242. In contrast to this method, an object-oriented modular system, where data can only be accessed by communication connectors, the procedure (the "method") and its data (the "internal data") form a unit, see

Fig. 5. These units are called "function blocks" and can only be handled as a whole. Data which has to be processed by this function block is sent to or in return received from the function block via connectors and is referred to by the term "communication objects". This principle does not only result in data protection but also in a system with low complexity compared to the procedure-oriented method.

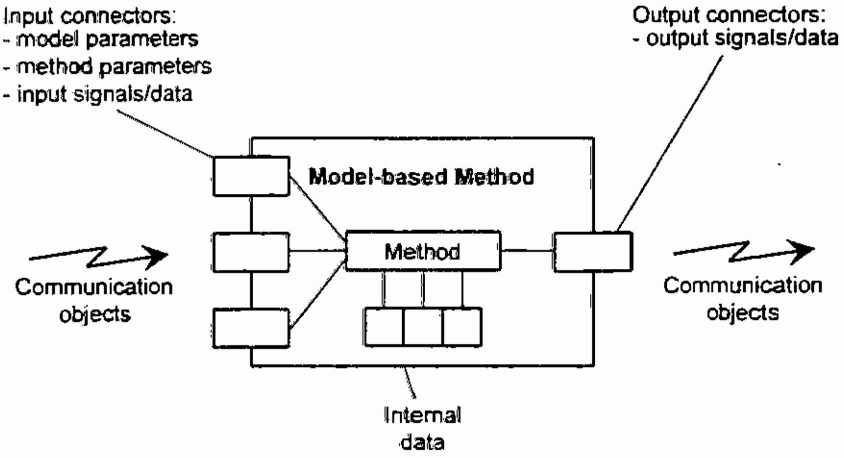


Figure 5: Model-based method in a Function Block System.

In our particular case, the function block contains the algorithm of our model-based method and some kind of internal data, such as model or method parameters or the simulation state. If allowed by the internal method of the function block, these data can be set by sending it to the referring input connectors. The actual input signal or input data (e.g. time series of measurements in case of an state observer) is sent to another input connector. It is then processed by the method of the function block which finally creates the output signals or data (e.g. the process state estimated by the observer) and writes it into the output connector where it can be accessed.

In the area of process control it is absolutely necessary that the communication objects are able to handle simple mechanisms as reading data from output connectors as well as complex message exchange procedures.

4 Object Model of the Data

Obviously it makes sense to archive the data received from the use of model-based method in a data base, because this data provides additional knowledge about the process which might be of great interest later.

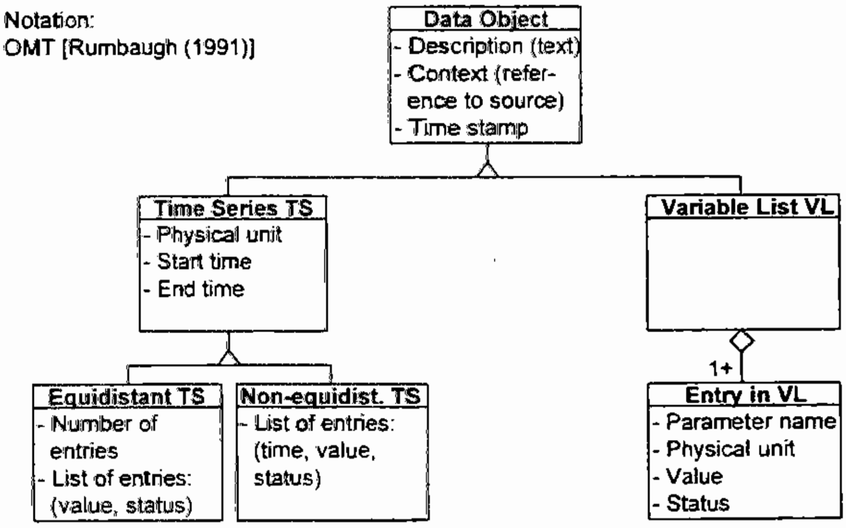


Figure 6: Object model of the data (simplified).

When data has to be archived in a data base, a model of the data objects needed is necessary. Such an object model can be graphically described by the object modeling technique (OMT) by Rumbaugh [6] and is shown in Fig. 6.

When using model-based methods, one has to deal with signals (equidistant or non-equidistant time series), different types of parameters (model parameters, method-specific parameters), and process states. Both parameters and states can be mapped to lists of variables, so that one has to distinguish between time series (histories) and variables (compare to the communication objects used by ACPLT/KS). In addition, it is often useful to describe the "quality" of the data values by a status.

A point which has often been neglected is which information needs to be stored in addition to the data generated by the model-based method itself (e.g. signals created by a simulation of the process). If an operator takes a look at old data from an archive (e.g. a signal) he needs to know the data source (e.g. if it is a measurement or simulation data) and the creation time of the data (e.g. when the simulation has been run). Thus every data object should be described by a text, a reference to the data source and a time stamp.

5 Scenarios

In the past, most applications of model-based methods have been restricted to the usage in one fixed "context", meaning that the data paths (or connections) involved have been fixed. One example is a predictive on-line simulation, which is fed by on-line measurements and regularly predicts the process trajectory for a couple of future time steps (usually for an unchanged operating mode of the plant), which is then shown on an operator monitor screen.

The usage of model-based methods in such a fixed context is of course helpful for the operator; but even more knowledge about the process can be made available, if he can use model-based methods as a tool without fixed data paths (connections), but where he has the free choice of data sources (and data sinks).

Fig. 7 shows an example, where the operator can dispose of a planning unit and a operation unit: The planning unit comprises modules which contain model-based methods and does not interact directly with the process control, while the operation unit executes the control strategy, either automatically or manually. If the operator wants to change the control strategy, he can define a planned control strategy, some kind of constraints and an operation mode of the plant. This data is the input for the planning unit. In this example the model-based method is a simulation of the process and the automatic functions and yields the process trajectory which may be analysed and evaluated by another tool. Based on this analysis and evaluation, a new control strategy may be specified either by a recursive optimisation algorithm or manually by the operator. If the operator is satisfied by the predicted process trajectory, he can either feed the planned control strategy into the operation unit as the actual control strategy or he can influence the process by manual control inputs which are based on the planned control strategy.

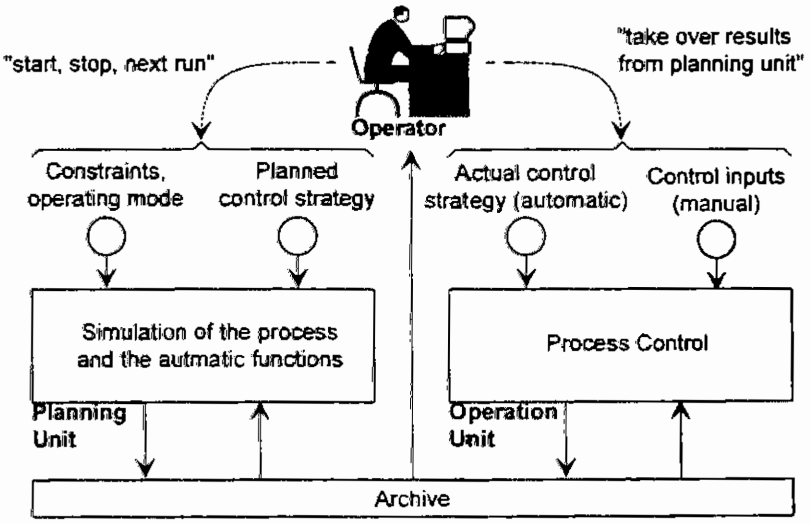

"start, stop, next run"
"take over results from planning unit"
Operator
Constraints, operating mode
Planned control strategy
Actual control strategy (automatic)
Control inputs (manual)
Simulation of the process and the autmatic functions
Process Control
Planning Unit
Operation Unit
Archive

Figure 7: Planning Unit and Operation Unit.

In such a flexible environment a common scheme how to handle model-based methods and their results becomes necessary. Because of the possibility of choosing different data sources, e.g. manual inputs or measurements (process history), it is necessary to keep the data context (i.e. the relation between input and output data) in order to be able to interpret it later. For example, when time series are loaded from the archive, one needs to know how these time series have been generated. Input data, the model-based methods involved and the output data form an information unit. In this paper, this unit is called a "scenario". The simplified object model of such a scenario is shown in Fig. 8 (object modeling technique [6]).

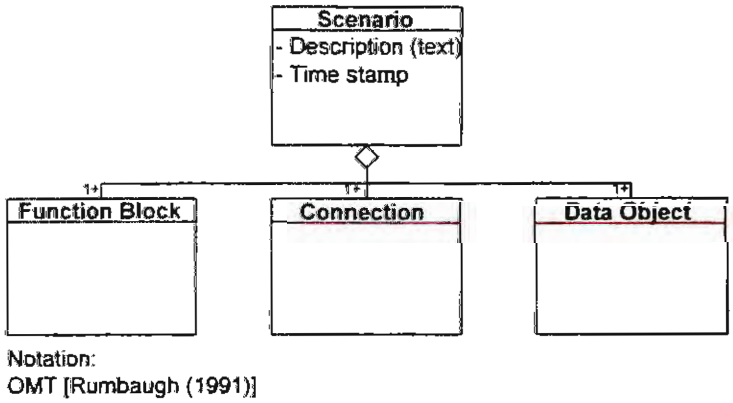


Figure 8: Object model of a Scenario (simplified).

A scenario consists of function blocks, connections between them and the data objects described before. The data is related to the connectors of the function blocks by linking it to the connector, e.g. an input signal of a simulation function block is linked to the referring input connector.

Summary

Due to an open system structure modern process control system offer the possibility to realise model-based methods in a distributed environment. This environment consists of commercial hardware (PCs and workstations) connected to the system by an Ethernet with the TCP/IP protocol. For the application of model-based methods, the following functional units are necessary: a communication system which allows to transfer variables and histories between the different platforms, a function block system which performs the algorithms of the model-based methods, a data base (archive) and an user interface. The data objects exchanged by these components are modelled using object-oriented methods. The basics of a scheme allowing to handle model-based methods, their input data and their results as one information unit is proposed. These information units, called "scenarios", are of great importance for the operator who is involved in the control of the process.

References

1. M. Polke (Editor): Process Control Engineering. VCH, Weinheim (Germany), 1994.
2. H. Schuler: Aufwand/Nutzen-Analyse von gehobenen Prozeßführungsstrategien in der Verfahrenstechnik. Automatisierungstechnische Praxis 36 (1994) 6, pp. 28-40.
3. H. Schuler: Methoden der Prozeßführung mit Simulationsmodellen. Automatisierungstechnische Praxis 31 (1989) 10, pp. 475-481 (part 1) and 31 (1989) 11, pp. 519-523 (part 2).
4. A. Brucker, J. Eul and T. Tauchnitz: INTERKAMA 95: Prozeßleitsysteme. Automatisierungstechnische Praxis 38 (1996) 4, pp. 12-38.
5. M. Arnold, U. Epple, M. Polke: Unternehmensweiter Zugriff auf Prozeßinformationen mit dem "PLT-Internet". Automatisierungstechnische Praxis 39 (1997) 1.
6. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen: Object-Oriented Modeling and Design. Prentice Hall, Inc. 1991.

SIMULATION OF THE REGENERATION OF DUST FILTERS

Ch. Stöcklmayer and W. Höflinger
Email: cstoeckl@fbch.tuwien.ac.at, whoeflin@fbch.tuwien.ac.at
Institute for Chemical Engineering, Fuel and Environmental Technology
Technical University Vienna, Getreidemarkt 9, 1060 Vienna, Austria

Abstract. Periodically regenerateable cake-forming filters have assumed a commercially dominant role in air purification because of their excellent dust collection capabilities. However intensive experimental investigations are usually necessary to optimally design such filters due to the absence of calculation models. In this work an extended computer simulation model is presented, which is capable of simulating the whole filtration (a alternate succession of filtration and regeneration of the filter medium). With this model the long-term behaviour of such a filter was examined and it was found that the compressibility of the dust has an major influence on the operation costs: The more compressible a dust is the sooner the filter has to be replaced due to particles inside the filter medium which cannot be removed anymore.

Introduction

During recent years the standards for the emission of pollutants into the air were steadily improved resulting in lower and lower limits for the concentration of gaseous and solid components in emitted gas-streams. A special problem hereby is meeting the limits for solid components (dust) especially in submicron size. In this area **periodically regenerateable cake-forming filters** have assumed a commercially dominant role due to their excellent dust collection capabilities. These filters are used to separate particles from gases with high dust concentrations, whereby the separation arises as the dust-gas mixture passes through the filter medium and the particles are retained. At the beginning of the filtration process the deposition of particles takes place inside the filter medium and later a dust cake is formed on the surface of the filter-medium. During the build-up of the filter cake the pressure drop over the filter increases making a regeneration of the filter medium indispensable. The regeneration typically starts at a defined maximum pressure drop of the filter and is either performed by a reverse pulse jet, or by mechanically shaking of the filter medium or by reverse flow through the filter medium. After regeneration the next filtration-cycle starts and a new dust filter cake is built-up on the cleaned filter medium. More details can be found in [1].

In order to optimally design such filters, there are three important parameters for the economical evaluation of the filter. The **pressure drop** increase of the filter, the dust separation efficiency of the filter especially directly after regeneration and the **residual pressure** drop after the regeneration of the filter medium. Once the filter medium and dust type are given very time-consuming experimental investigations have to be performed to characterise the filtration behaviour in respect to these parameters and to optimally design the filter. This procedure implies a kind of trial and error strategy combined with a lot of empirical experience essential to keep the effort within reasonable limits. Due to this it would be very **useful to have a model for the whole filtration process**, which enables the prediction of pressure drop, residual pressure drop after regeneration and dust separation efficiency for a certain combination of dust and filter medium therebye **saving much experimental effort**. Additionally many effects found in practical investigations are not yet fully understood and therefore **theoretical explanations of the regeneration of dust filters** would be very useful too. Due to the complexity of the problem, a computer simulation model would have to be employed, but till now there was no such model available.

Improving regenerateable dust filters

Figure 1 schematically depicts the pressure drop-curve of a typical filtration. In the course of each filtration cycle the pressure drop increases until it reaches a defined maximum pressure drop. Then regeneration takes place. After regeneration the pressure drop (=residual pressure drop) is quite low again, but unfortunately the residual pressure drop does not stay constant, but increases from one filtration cycle to next. The two most important parameters are now discussed in more detail:

The **pressure drop** during filtration can increase linearly or exponentially depending on the compressibility of the dust filter cake. In the example in Figure 1 an linear increase is schematically shown indicating an incompressible dust filter cake. This behaviour has an major influence on the filtration process because in the compressible case the maximum pressure drop for regeneration is reached much faster, which leads to an more frequent cleaning of the filter decreasing the lifetime of the filter medium.

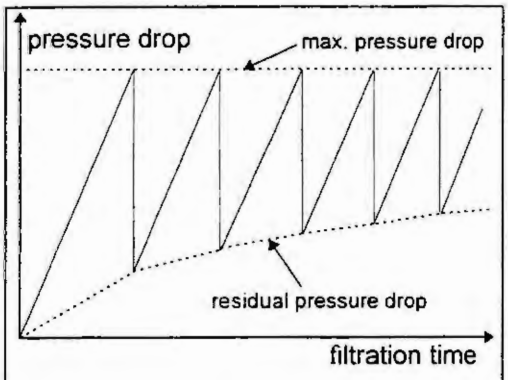


Figure 1: Schematical pressure drop curve after during some filtration cycles

The **residual pressure drop** is a measure for the dust remaining on and especially in the depth of the filter medium. Unfortunately it happens quite often in practice that the residual pressure drop does not reach a constant value after several filtration cycles, but increases steadily due to particles inside the filter medium which cannot be removed by cleaning. As soon as the residual pressure drop reaches a certain limit, the filtration process becomes uneconomic because of high pressure losses and short cycle times and the filter medium has to be replaced. Therefore investigations concerning the influence of various material and operation parameters on the residual pressure drop are of major interest to increase the lifetime of a filter and save operation costs of the plant. An important group of these parameters are the interparticle forces between the dust particles, which define the compressibility of particle masses. They have an major influence on the long-term behaviour of such filters.

The **regeneration** of the filter medium is performed in one of three ways: by a reverse pulse jet, by mechanically shaking of the filter medium or by reverse flow through the filter medium. When the filter medium is cleaned in one of these ways, different physical mechanism are acting. If the filter medium is mechanically cleaned the acceleration of large parts of the dust filter cake will be the major physical effect for cleaning. If the cleaning takes place with reverse flow the drag force on parts of the dust filter cake and especially the drag forces, when the gas passes through holes in the filter cakes will be the major physical effect responsible for dust removal. Pulse-jet-cleaning combines both methods by first accelerating the dust cake and then blowing gas through the filter medium. Though there are theories about the physical processes taking place when a dust filter cake is cleaned there is still a considerable lack of theoretical knowledge for all cleaning-methods.

In practice a specific dust is present which has to be removed from the gas and the filter medium and the operation conditions and the cleaning method have to be found with respect to a minimisation of costs. This means that the requirements are low pressure drop rises, a sufficiently high separation efficiency and a low residual pressure drop, which does not increase significantly after several cycles of filtration and regeneration (stable conditions). Typically the optimal operation and design-parameters are found by a kind of trial and error strategy, where the important parameters are varied under the guidance of empirical knowledge. With a very **high experimental effort** the optimal configuration can then be found.

This experimental effort can be drastically reduced by using a simulation model for the whole filtration process. First the parameters of the model have to be determined from a few experiments with a given type of dust and then it should be possible to do a lot of the testing with the model instead of the filtration apparatus. The model necessary for such a simplification should ideally have the following features: It should be capable of simulating the compressible build-up of different dust filter cakes on various filter media including the calculation of the correlating pressure drop and the dust separation efficiency during filtration, it should be able to simulate the regeneration of the filter medium and to calculate the residual pressure drop after regeneration and it should also be possible to simulate the filtration behaviour for several cycles of filtration and regeneration to see the long-term-behaviour of the filter.

Due to the complexity of this problem, a **computer simulation model** has to be employed. The major idea is again that the parameters of this simulation model can be calculated from a few experiments with a given filter medium and dust and that afterwards the filter can be optimised by primarily using the computer simulation. Till now there are some calculation schemes for the pressure drop of a dust filter but these models lack a physical connection between the parameters of the model and e.g. the dust properties. Especially no uniform model, which includes the regeneration of the filter medium is available.

The authors have already developed a computer simulation model, which is capable of simulating the 2-dimensional build-up of a compressible dust filter cake on a flat plane based on defined interparticle forces, which are the most important parameters to describe the behaviour of dust particles in the dust filter cake [2]. In

[3] it was shown that these parameters can be determined from a single pressure drop curve of one filtration cycle. The model was then extended to the simulation of the dust cake build-up on needle-felts. Here the resulting pressure drop of the filter and the number of dust particles in the clean gas (dust separation efficiency) were in good qualitative agreement with experimental data [4]. The influence of the interparticle forces between the particles on different compression phenomena and therefore on the resulting pressure drop of a dust filter was in detail investigated in [5]. In this work the regeneration of the filter medium is included into the existing computer simulation model. As a first step, this study concentrates on the first basic mechanism of regeneration, the acceleration of the dust filter cake. This extension enables the simulation of a number of filtration cycles and therefore the long-term-behaviour of a filter. We especially concentrate on the mechanism behind mechanical cleaning of the filter medium and investigate the influence of the dust properties (interparticle forces) on the residual pressure drop after several cycles of filtration and regeneration, because this effect is of major practical importance.

Simulation model

This computer simulation model enables the simulation of the two-dimensional build-up and the compression of a dust filter cake on a needle-felt at a constant fluid flow. The dust particles are modelled as being spherical and of uniform diameter. A friction angle φ and a maximum adhesion force Z_{max} are used for the description of the interparticle forces and therefore the definition of the limits for compression movements of particles inside the filter cake and in the filter medium. The friction angle and the maximum adhesion force are assumed to be the same for particle-particle contacts and for particle-fibre contacts. The filter medium itself is 2-dimensionally modelled as a number of layers of randomly arranged spheres, each of them idealising a fibre of the needle-felt. According to this model, the filter medium is defined by a overall porosity ε_F, a fibre diameter d_F and the height h_F of the medium. The porosity in the needle-felt is assumed to be constant over the height.

When modelling the 2-phase-flow in a filter medium the flow field has to be calculated first considering the fibres of the filter medium. Then the particle-paths in the flow can be determined as a function of the drag forces on the particles. Due to the complexity of these calculations with a random fibre arrangement, simplifications have to be made on the one hand for the flow-field and on the other hand for the fibre arrangement. The result of the calculation of the particle-paths is usually given as the collection efficiency η of one fibre [1], which defines the percentage of particles that hit a fibre when they approach it. Additionally it has to be taken into account that a lot of the particles hitting the fibre are repelled and therefore do not permanently adhere on the fibre. Therefore a effective separation efficiency of a fibre is defined. It is the product of the collection efficiency η and the adhesion probability, which is the percentage of particles sticking on the fibre when they hit the fibre [1]. In the literature there are a lot of formula for the collection efficiencies for different fibre arrangements and flow fields. The problem is that there is no applicable model for the calculation of the adhesion efficiency, which was found to be of major importance for the total separation efficiency for the whole filter in practice [1].

Therefore we made the collection efficiency of the fibres a parameter of our simulation, which should later be defined by experiments. Of course it has to be kept in mind that this value is only defined for a single particle and fibre diameter, a filter medium porosity and a filtration velocity. Though the filtration velocity changes inside the filter medium due to deposited particles, we assume that this can be neglected, because the collection efficiency has only an influence on the filtration when almost no particles are yet deposited and then the velocity is not changed very much. The adhesion probability was set to 1. Then we simplified the calculation of the particle-paths inside the filter medium by the following assumptions: As long as there is no fibre in their way the particles move along straight lines rectangular to the surface of the filter medium (they are not deflected by particles already deposited). As soon as a particle approaches a fibre (the distance is lower than one fibre diameter d_F) the given collection efficiency of the fibre defines, if the particle hits the fibre or it is deflected and passes the fibre. Each particle hitting a fibre is assumed to adhere to the fibre.

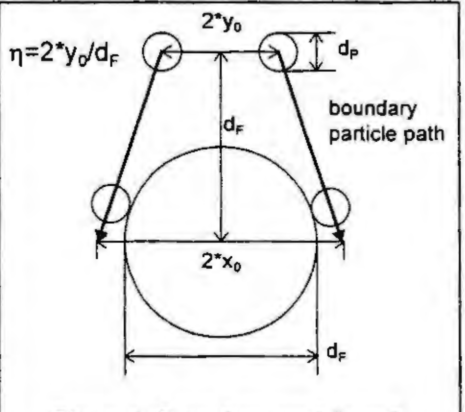


Figure 2: Boundary particle path

Figure 2 shows a fibre of the filter medium with the boundary-path for particles being captured by the fibre defined by the collection efficiency η. With a given fibre diameter d_F and collection efficiency η y_0 is calculated with the formula given in figure 2. Then x_0 is calculated by using the criteria that a particle starting at a horizontal distance of y_0 from the centre of the fibre should just pass it. All particles approaching the fibre within a horizontal distance of y_0 to the centre of the fibre are captured, all others pass the fibre. The collection efficiency controls now how many particles are captures by a fibre. It can have values between 0 (all particles pass) and $1+d_P/d_F$ (all particles are captured).

For the calculation of the pressure drop the whole filter medium and filter cake is divided into layers. The pressure drop of a single layer in the dust cake is calculated from the porosity of the layer and the specific surface of the particles and the Kozeny-constant $K_{0,P}$ using the Carman-Kozeny-equation [2]. The same approach is used to calculate the pressure drop of the filter medium, but a new Kozeny-constant $K_{0,F}$ for the fibres is defined and the specific surface of the fibres is also calculated. In all layers of the filter medium where particles are already deposited the pressure drop is calculated with the overall porosity and an average Kozeny-constant and an average specific surface depending on the fraction of the filled volume of a layer filled with particles and fibres. The compression pressure acting at the particles of a certain layer can be easily defined in the dust cake by just summing up all pressure losses of all layers above. Inside the filter medium the compression pressure on particles has to be reduced, because the fibres support the dust filter cake. This reduction of the compression pressure is done in dependence on the fraction of particle and fibre volume in each single layer.

Concerning the regeneration a model for the removal of the dust cake from the filter medium as a function of the applied acceleration was implemented. First it is determined which dust particles belong to the connected mass of dust particles called dust filter cake. This is done by taking all particles, which have a direct connection to any particles above the filter medium, which are unequivocally part of the dust cake. First the force acting on the filter cake is calculated from the applied acceleration the mass of the dust cake and then the stress on the particle-fibre contacts is determined, assuming that the dust cake does break off only at particle-fibre contacts and not at particle-particle-contacts. The comparison between the strength and the stress on these contacts shows, if the dust cake is removed or not. In case the acceleration was sufficient high, all particles, which are not mechanically obstructed by fibres are removed. The rest stays inside the filter medium.

Beside these new parts of the model briefly discussed in this section more detailed information about the compression model can be found in [2].

Simulation results

The aim of this work was to investigate the influence of the interparticle forces of the dust particles on the residual pressure drop of the filter and especially the long-term behaviour of the residual pressure drop. Therefore the friction angle φ was varied and all other parameters of the simulation were kept constant (Table 1). The maximum pressure drop, where regeneration is performed was set to 1000 Pa and the acceleration of the filter medium was set high enough to remove the dust filter cake in any case.

particle diameter d_P [µm]	15	filter medium height h_F [-]	15 layers
fibre diameter d_F [µm]	50	max. adhesion force [N]	10^{-7}
filter medium porosity ε_F [-]	0,9	Kozeny-constant (particles) $K_{0,P}$ [-]	100
collection efficiency η [-]	0,1	Kozeny-constant (fibres) $K_{0,F}$ [-]	10

Table 1: Simulation parameters kept constant

Figure 3 shows the pressure drop curves as a function of filtration time for 4 different friction angles (15°, 20°, 30° and 40°), but here the long-term behaviour of the residual pressure drop is of major interest. A low friction angle means that the interparticle forces are quite low and the dust filter cake is very compressible. Looking at the upper left graph in figure 3 shows that the residual pressure drop steadily increases and reaches very high values already after a few filtration cycles for a very compressible dust. The next graph on the upper right side shows a much slower increase of the residual pressure drop. This tendency continues which higher friction angles. In the last graph on the lower right side the friction angle is already high enough to make the dust filter cake almost incompressible. Here the residual pressure drop stays very low and almost becomes constant. This case is of course most wanted in practice.

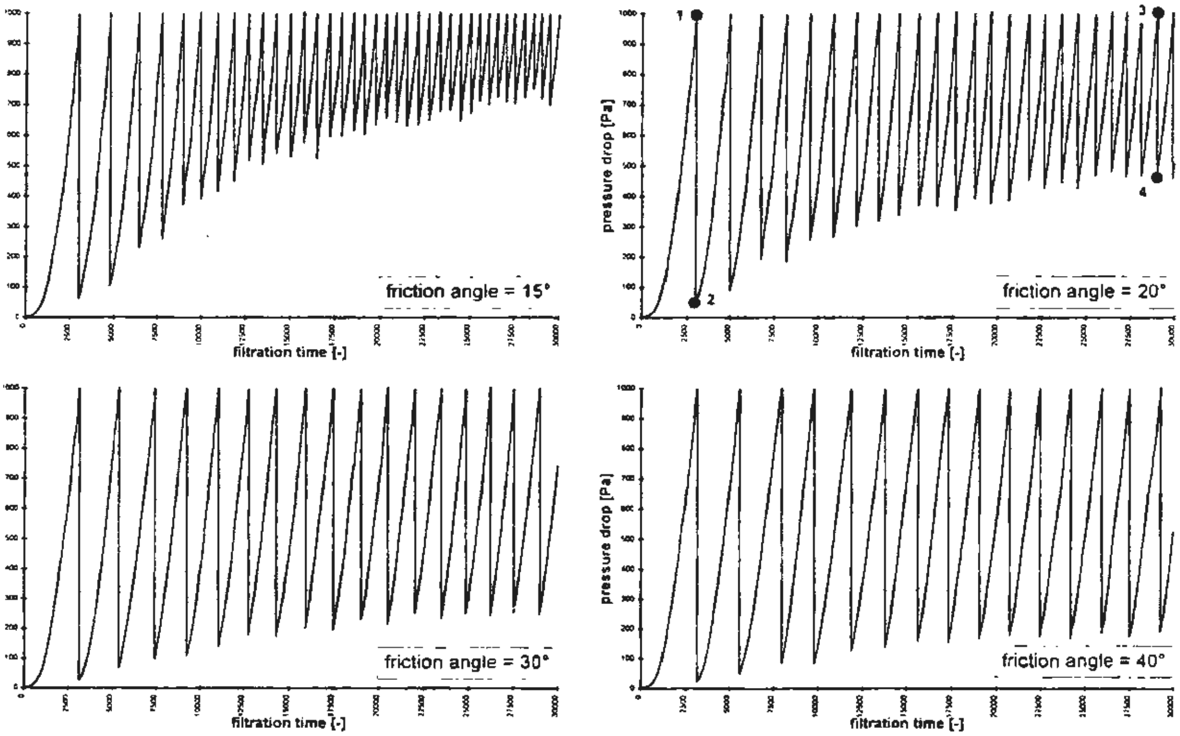
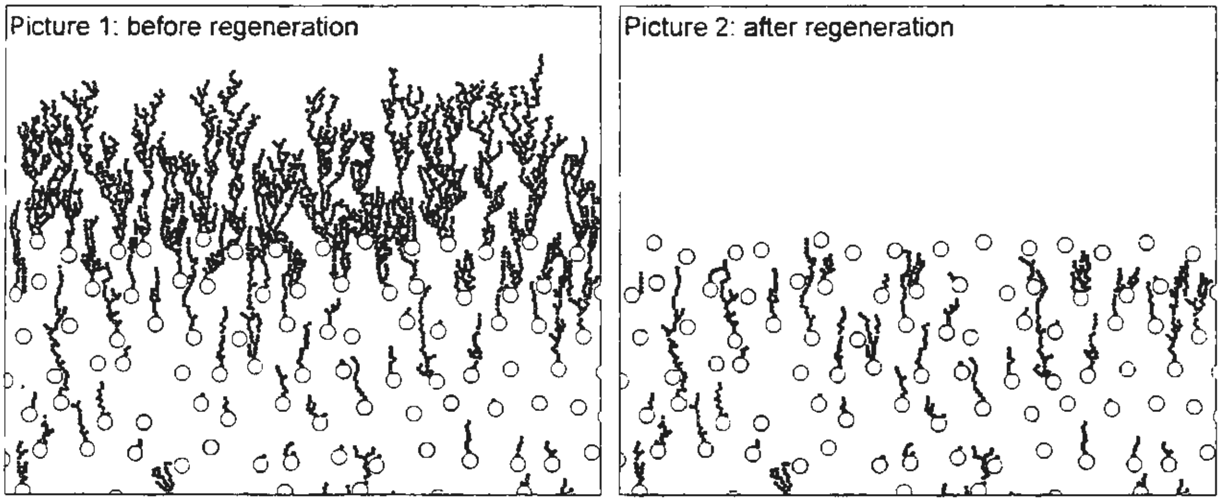


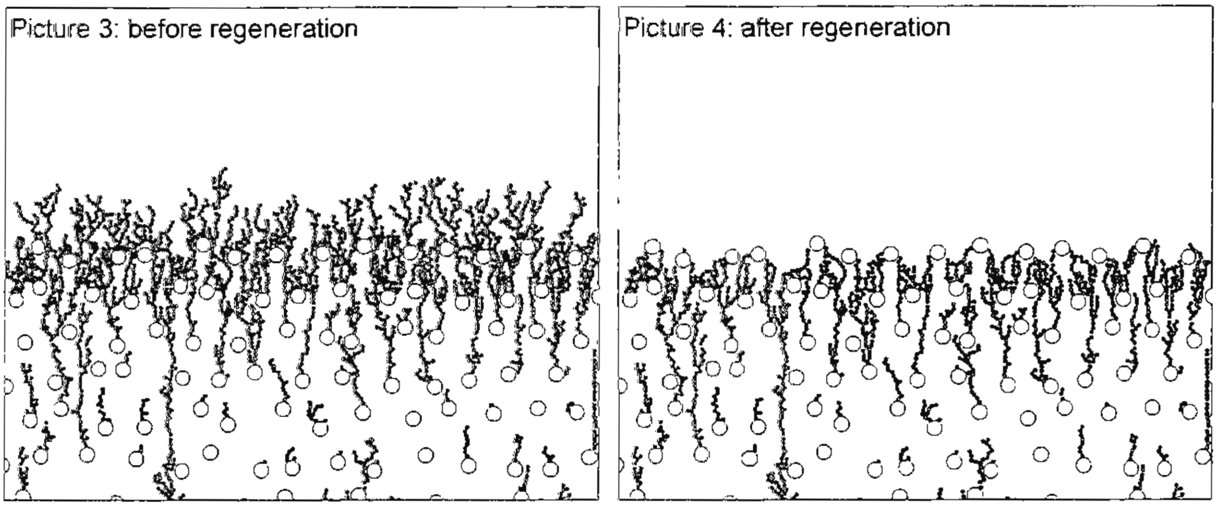
Figure 3: Residual pressure drop as a function of the friction angle


Picture 1: before regeneration

Picture 2: after regeneration

Figure 4: Structure of the dust cake before and after the 1st regeneration for φ=20°

To find out, how the interparticle forces between the dust particles influence the long-term behaviour of the dust filter so much, one of the four simulation examples (with a friction angle of 20°) is taken out and examined in more detail. Therefore the structure of the dust filter cake before and after regeneration is shown on the one hand for the first regeneration (figure 4) and on the other hand for the last but one (figure 5). This is also indicated in figure 3 with 4 dots. Figure 4 shows that the dust cake at the beginning of the filtration is primarily built on the surface of the filter medium and can therefore be almost completely removed. After some filtration cycles (figure 5) the dust filter cake is built more and more in the depth of the filter medium. This entails that quite a large part of the dust cake cannot be removed, because it is inside the filter medium, which can be seen very good in the right picture of figure 5.


Picture 3: before regeneration

Picture 4: after regeneration

Figure 5: Structure of the dust cake before and after the 24th regeneration for φ=20°

The explanation of this behaviour is connected with, what we call the **build-up of a second compressible dust cake inside the filter medium**. This means that at the beginning of the filtration a dust cake is build on top of the filter medium, but later after several cycles of filtration and regeneration the cake builds more and more in the depth of the filter medium depending on the compressibility of the dust. The reason, why the residual pressure drop is so much increasing with a compressible dust is that the dust cake built in the filter medium itself is compressed and therefor more and more particles are deposited in even deeper layers of the filter medium where they cannot be removed any more.

Conclusions

In this work we show that the extended computer simulation model is capable of simulating the dust cake build-up and the mechanical regeneration of a dust filter. Filtrations consisting of several cycles of filtration and regeneration were performed with this model and the simulation results show that the compressibility of the dust is a major factor for the increase of the residual pressure drop of a filtration. Summarizing it can be said that ideally **a incompressible dust gives low and stable residual pressure drops**. The model also explains why the residual pressure drop of a filtration can increase steadily: This is due to a compressible dust cake built in the depth of the filter medium.

Acknowledgement

This project is supported by the Fond zur Förderung der Wissenschaftlichen Forschung (grant P10192-ÖCH).

References

1. Löffler, F., Staubabscheiden. Georg Thieme Verlag, Stuttgart, 1988.
2. Höflinger, W., Stöcklmayer, Ch. and Hackl, A., Model Calculations of the Compression Behaviour of Dust Filter Cakes. Filtration & Separation, 1994, 31(8), 807-811.
3. Stöcklmayer, Ch., Krammer, A. and Höflinger, W., Using a Computer Simulation Method for Calculating Compression Parameters from Experimentally Obtained Pressure Drop Curves in Dust Cake Filtration. In: Proc. Int. Symposium Filtration and Separation of Fine Dust, Vienna, Austria, 1996, 126-139.
4. Stöcklmayer, Ch. and Höflinger, W., Simulation of the Filtration Behaviour of Dust Filters. Simulation Practice and Theory, to be published 1997.
5. Stöcklmayer, Ch. and Höflinger, W., Using a Computer Simulation Method for Investigating and Clarifying different Compression Phenomena in Dust Cake Filtration. In: Proc. EUROSIM'95, Vienna, Austria, 1995, 831-836.

DIRECT AND ADJOINT OIL SPILL ESTIMATES

Yuri N. Skiba
Centro de Ciencias de la Atmósfera, UNAM
Circuito Exterior, CU, México, D.F., 04510, MEXICO
E-mail: skiba@servidor.unam.mx

Abstract. Propagation of the oil spilling from a damaged oil tanker is considered in a limited sea area. The accident consequences are evaluated by means of direct and adjoint oil concentration estimates in ecologically sensitive zones. While the direct estimates are preferable to get a comprehensive idea of the oil spill impact on the whole area, the adjoint ones are useful and economical in studying the sensitivity of the oil concentration in some zones to variations in the accident site and in the oil spill rate from the tanker. Thanks to special boundary conditions set at the inflow and outflow parts of the open boundary, the main and adjoint oil transport problems are both well-posed according to Hadamard [5]. The estimates obtained in [14] are generalized to the three dimensions. Balanced, absolutely stable 2nd-order finite-difference schemes based on the splitting method are constructed for the 2-D and 3-D cases, both. The main and adjoint schemes are compatible in the sense that at every fractional step of the splitting algorithm, the 1-D split operators of both the schemes satisfy a discrete form of the Lagrange identity [8]. In the special unforced and non-dissipative case, the schemes have two conservation laws each. Numerical algorithms are realized by the factorization method.

Introduction.

Marine oil transportation has increased enormously in recent years. A heavy oil spill caused by an accident involving tanker often has dramatic impact on the environment. The problem is of great practical importance [12],[17], and many studies have been devoted lately to the oil slick movement and spreading [2],[3],[11]. The mathematical formulation of the model is complicated by the fact that the oil being released into marine environments is subjected to various weathering processes such as spreading and drift, advection and dissolution, evaporation and sinking, and others. Besides, the oil velocity can be correctly determined only if both the wind and the current factors are taken into account [1]. Nevertheless, simulation of the oil movement can give additional insights into the dynamical processes that influence such a motion.

Obviously, the knowledge of the oil transport equation solution allows us to get a comprehensive idea of the oil spill impact on the whole area, and in particular, to evaluate average oil concentrations in ecologically sensitive zones of the area (fishery regions, tourist coastal zones, etc.). Any average value obtained in this way is hereinafter called the direct estimate. One more oil concentration estimate equivalent to the direct one, is derived in [14] with the adjoint transport equation and called the adjoint estimate. It is a simple integral formula that explicitly relates the average oil concentration in the zone to the oil spill rate through the value of the adjoint oil transport problem solution at the accident site.

There is a fundamental difference between the adjoint and direct evaluations. In the direct method, the oil transport problem solution depends on the two principal parameters: the accident site and the oil spill rate (i.e., the oil amount spilling in a unit time from the damaged tanker). Contrastingly, the solution of the adjoint transport problem depends on the ecologically sensitive zone monitored and is independent of these two parameters. Hence it can be found beforehand irrespective of a concrete accident (i.e., irrespective of the accident site and oil spill rate). Due to these features, the adjoint method has certain advantages over the direct one in the model sensitivity study, especially in the 3-D case, for example, to analyze the accident site dependence, or the oil spill rate dependence, of the oil concentration in the zone. The direct method requires to solve the oil transport problem repeatedly whenever the accident site or the oil spill rate varies, and therefore is time consuming. The adjoint method is more economical, for no solution of the oil transport problem is required, and the adjoint problem solution once calculated can repeatedly be used in evaluating the oil concentrations for various possible accident sites and oil spill rates. On the whole, the direct and adjoint approaches complement each other nicely in studying the consequences of the oil spill [14].

In the present paper, a simple three-dimensional limited area oil transport model as well as its adjoint are formulated. The 2-D direct and adjoint oil concentration estimates obtained in [14] are generalized here to the three dimensions. Balanced unconditionally stable 2nd-order finite-difference schemes and numerical algorithms based on the splitting method are also given. The same methods can be applied if oil enters the marine environment from natural sources, offshore production, waste waters, etc.

The 2-D oil spill problem

We now briefly give the main results of the work [14]. Let $r_0 \equiv (x_0, y_0)$ be the site of an accident with the oil tanker in a 2-D sea domain D with boundary S, and $t=0$ be the accident initial moment. Denote by $Q(t)$ the oil spill rate (the oil amount spilling in a unit time) from the damaged tanker, and by $\phi(r,t)$, the amount of oil (associated with the oil slick thickness on the sea surface) at the point $r=(x,y)$ of D and the instant $t>0$. The oil slick propagation in D and time interval $(0,T)$ is described by the transport equation

$$\frac{\partial}{\partial t}\phi + U \cdot \nabla \phi + \sigma\phi - \nabla \cdot \mu \nabla \phi = Q(t)\delta(r - r_0) \tag{1}$$

where μ is the diffusion coefficient, Δ the 2-D gradient, $\delta(r)$ the Dirac mass at the point r, and the parameter s characterizes decreasing of $\phi(r,t)$ because of evaporation. The velocity $U(r,t)=(u(r,t),v(r,t))$ of the oil propagation is assumed to be known and satisfy the continuity equation

$$\frac{\partial}{\partial x}u + \frac{\partial}{\partial y}v = 0 \tag{2}$$

This vector can be calculated by using the climatic (seasonal or monthly) sea surface currents and winds [1], or special dynamic models. As the initial condition we take the absence of the oil on the sea surface:

$$\phi(r,0) = 0 \quad at \ \ t = 0 \tag{3}$$

Care is required in setting conditions at the limited area boundaries [7],[10],[13]. Let U_n be a projection of the velocity U on the outward normal n to the boundary S. We divide S into the "outflow" part S^+ where $U_n \geq 0$ (i.e., U is outward-directed, and oil flows out of the domain D), and the "inflow" part S^- where $U_n<0$ (U is inward-directed). As the boundary conditions for Eq.(1) we take

$$\mu\frac{\partial}{\partial n}\phi - U_n\phi = 0 \quad at \ \ S^-, \qquad \mu\frac{\partial}{\partial n}\phi = 0 \quad at \ \ S^+ \tag{4}$$

By the first condition (4), the combined diffusive plus advective oil flow is absent at the part S^- (no oil flows into D from the outside where water is free of oil). The second condition (4) means that at the boundary S^+, the diffusive oil flow is negligible as compared with the advective oil outflow $U_n\phi$ from D. In the non-diffusion limit ($\mu=0$), the first condition (4) is reduced to $\phi=0$ (there is no oil on the inflow boundary), while the other vanishes, as it must, since the pure advection problem ($\mu=\sigma=0$) does not require conditions at the outflow boundary where its solution is predetermined by the method of characteristics [4]. The second condition (4) includes the coast line where $U_n=0$. For a closed basin D everywhere bounded by the coast line, S^- is empty and $S=S^+$. Thus formulas (4) not only include the well-known coast line condition, but also approach, in the non-diffusion limit, the correct boundary conditions for the pure advection equation. The problem (1)-(4) is well-posed according to Hadamard [5], that is, any its solution is unique and stable to initial perturbations [14].

The adjoint transport problem

$$-\frac{\partial}{\partial t}g - U \cdot \nabla g + \sigma g - \nabla \cdot \mu \nabla g = P(r,t) \tag{5}$$

is constructed using the concept of the adjoint operator in the Hilbert space [6,9]. Equation (5) is completed by the boundary conditions

$$\mu\frac{\partial}{\partial n}g = 0 \quad at \ \ S^-, \qquad \mu\frac{\partial}{\partial n}g + U_n g = 0 \quad at \ \ S^+ \tag{6}$$

and the "initial" condition

$$g(r,T) = 0 \quad at \ \ t = T \tag{7}$$

It can be shown [13] that the adjoint problem (5)-(7),(2) is well-posed according to Hadamard only if it is solved in D backward from $t=T$ to $t=0$. Let the average oil concentration in some ecologically sensitive zone Ω and time interval $(T-\tau,T)$ is given by the direct estimate

$$J(\phi) = \frac{1}{\tau|\Omega|}\int_{T-\tau}^{T}\int_{\Omega}\phi(r,t)\,dr\,dt \tag{8}$$

For the adjoint estimate to be equivalent to (8), the adjoint problem must be solved with a special forcing P which is uniquely determined by the form of the direct estimate. For example, in the case of (8), P must be as follows: $P(r,t) = 1/(\tau|\Omega|)$ if the point (r,t) belongs to Ω and $[T-\tau,T]$, and $P(r,t)=0$ otherwise, where $|\Omega|$ is the area of Ω. Then the adjoint estimate yields

$$J(\phi) = \int_0^T g(r_0,t)Q(t)dt \tag{9}$$

where $g(r_0,t)$ is the value of the adjoint problem solution at the accident point r_0.

In some situations, one or the other of these estimates is preferred. Several examples given in [14] explain in detail how to decide between them. The direct estimate (8) utilizes the solution $\phi(r,t)$ of the problem (1)-(4), and hence, depends on the two principle oil spill parameters: the oil spill rate $Q(t)$ and accident site r_0. Therefore it is favoured if a comprehensive information should be obtained on the oil concentrations in the whole D or/and in various its zones Ω, or if the time available to take precautions against polluting a zone Ω is assessed ([14], example 3). By contrast, the adjoint estimate (9) employs the solution $g(r_0,t)$ of the equation (5). Being independent of both Q and r_0, the adjoint problem may be solved for each ecologically sensitive zone Ω irrespective of a concrete accident with the oil tanker. This approach is convenient and economical in the model sensitivity study when the r_0-dependence, or $Q(t)$-dependence, of the oil concentration $J(\phi)$ is analyzed ([14], examples 4,5). For instance, given ecologically sensitive zone Ω, let us search the most dangerous accident point on the tanker route in that oil spill in this point maximizes (9). The integral (9) as calculated with any r substituted for r_0 determines in D the 2-D function whose maximum on the tanker route indicates the most dangerous route point.

The adjoint method is especially efficient in the case when the oil transport is studied with climatic (seasonal or monthly mean) winds and currents ([14], examples 1,2). Then the adjoint transport equation solution can be calculated for each ecologically sensitive zone in advance, and fed into a computer. Besides, the estimate (9) uses the adjoint solution values just at the accident site, and hence, the adjoint solution values at the grid points resting on the tanker route is all that must be kept in the computer. Indeed, any of these points (and only such points) is a possible site of the oil spill. Thus if oil tanker has met an accident, and both its site and oil spill rate are approximately known then it is easy to give a preliminary estimate of the average oil concentration in any ecologically sensitive zone. All one has to do is to choose out of the set of various solutions stored in the computer the adjoint problem solution corresponding to the zone and use its values at the accident site to take the time integral (9).

The 3-D oil transport problem

Let us consider a three-dimensional limited area domain D of the upper layer of the sea with the boundary $\Omega=S+S_0+S_H$ being the union of the lateral surface S, the sea surface S_0 at $z=0$, and the lower surface S_H at $z=H$. Let $\phi(r,t)$ be an oil content (amount) in an infinitesimal volume of the water in the point $r=(x,y,z)$ of D, and let $r_0=(x_0,y_0,z_0)$ be the oil spill source location. The oil propagation in the domain D and time interval $(0,T)$ is described by the transport-diffusion equation

$$\frac{\partial}{\partial t}\phi + U\cdot\nabla_3\phi + \sigma\phi - \nabla\cdot\mu\nabla\phi - \frac{\partial}{\partial z}\nu\frac{\partial\phi}{\partial z} = Q(t)\delta(r-r_0) \tag{10}$$

where μ, ν are the diffusion coefficients, ∇_3 is the 3-D gradient, σ is the parameter that characterizes decreasing of $\phi(r,t)$ because of various chemical processes. It is assumed here that the oil propagation velocity vector $U(r,t)=\{u(r,t),v(r,t),w(r,t)\}$ is known and satisfies the continuity equation

$$\frac{\partial}{\partial x}u + \frac{\partial}{\partial y}v + \frac{\partial}{\partial z}w = 0 \tag{11}$$

Let U_n be the projection of U on the outward normal n to the boundary Ω, besides $U_n=0$, i.e. $w=0$ on S_0. In practice, known horizontal components u and v, the vertical component w is determined by integrating (11) over z from 0 to z. Hence, typically, $w(x,y,H,t)$ is nonzero on the lower surface S_H. Once again, we use (3) as the initial condition in D, and denote by the superscript "+" the parts of the boundaries S, S_H where $U_n \geq 0$ (U is outward-directed), and by the superscript "-", their complementary parts where $U_n<0$ (U is inward-directed). If the lateral boundary surface S consists only of some parts of the surfaces $x=Const$ or $y=Const$ (such will be indeed the case for the finite-difference problem) then U_n always coincides with w on S_H, and with either $\pm u$ or $\pm v$ on S. As boundary conditions for Eq.(10) we take

$$\mu\frac{\partial}{\partial n}\phi - U_n\phi = 0 \quad at\ S^-, \qquad \mu\frac{\partial}{\partial n}\phi = 0 \quad at\ S^+ \tag{12}$$

$$\nu\frac{\partial}{\partial z}\phi = \alpha\phi \quad at\ S_0 \tag{13}$$

$$\nu\frac{\partial}{\partial z}\phi - U_n\phi = 0 \quad at\ S_H^-, \qquad \nu\frac{\partial}{\partial z}\phi = 0 \quad at\ S_H^+ \tag{14}$$

Condition (13) where $\alpha \geq 0$ is a known coefficient (or function) defines the losses of oil on the sea surface through evaporation. It is easy to show that (10)-(14) is the well-posed problem whose solution ϕ is unique and stable to initial perturbations in the $L_2(D)$-norm [13]. Besides, this norm is conserved provided that $Q=0$, $\mu=\sigma=0$, $\nu = 0$, and S is a coast line ($U_n=0$ everywhere at S). By the oil balance equation

$$\frac{\partial}{\partial t}\int_D \phi dr = Q(t) - \int_D \sigma\phi dr - \int_{S_0} \alpha\phi d\Omega - \int_{R+} U_n\phi d\Omega \tag{15}$$

the average oil concentration in the domain D increases because of the oil spill ($Q>0$), and decreases by reason of various physical and chemical processes ($\sigma>0$, $\alpha>0$) and advective oil outflow across $R+=S^+ + S_H^+$.

The 3-D adjoint problem and oil concentration estimates

The adjoint equation

$$-\frac{\partial}{\partial t}g - U\cdot\nabla_3 g + \sigma g - \nabla\cdot\mu\nabla g - \frac{\partial}{\partial z}\nu\frac{\partial g}{\partial z} = P(r,t) \tag{16}$$

is solved in D backward from $t=T$ to $t=0$ with "initial " condition (7) and boundary conditions

$$\mu\frac{\partial}{\partial n}g = 0 \quad at\ S^-, \qquad \mu\frac{\partial}{\partial n}g + U_n g = 0 \quad at\ S^+ \tag{17}$$

$$\nu\frac{\partial}{\partial z}g = \alpha g \quad at\ S_0 \tag{18}$$

$$\nu\frac{\partial}{\partial z}g = 0 \quad at\ S_H^-, \qquad \nu\frac{\partial}{\partial z}g + U_n g = 0 \quad at\ S_H^+ \tag{19}$$

Note that the zero initial condition (7) is essential in deriving the adjoint estimate [15]. Also, let $P(r,t)\equiv 0$, and $Q(t)\equiv 0$. Then the substitution of $t'=T-t$ into (16) shows that the homogeneous equations (10) and (16) differ only in the sign of U. Thus S^+ being the outflow part of S for Eq.(10) is the inflow part S^- for Eq.(16). This fact explains the difference in the form of boundary conditions (12)-(14) and (17)-(19).

Note that the direct and adjoint estimates of the average oil concentration in a 3-D zone Ω have the same form as (8) and (9) except that the integral in (8) is now three-dimensional, and $|\Omega|$ is the volume of Ω. In general, the adjoint estimate (9) with g found from (16) with the forcing P is equivalent to the direct estimate

$$J_P(\phi) \equiv \int_0^T\int_D P(r,t)\phi(r,t)dr\,dt \tag{20}$$

Main and adjoint numerical schemes of the oil transport problem

Using (11) the operator A of Eq.(10) can be written as $A=A_1+A_2+A_3$ where

$$A_1\phi = \frac{1}{2}\frac{\partial}{\partial x}(u\phi) + \frac{1}{2}u\frac{\partial\phi}{\partial x} - \frac{\partial}{\partial x}\mu\frac{\partial\phi}{\partial x}, \qquad A_3\phi = \frac{1}{2}\frac{\partial}{\partial z}(w\phi) + \frac{1}{2}w\frac{\partial\phi}{\partial z} + \sigma\phi - \frac{\partial}{\partial z}\nu\frac{\partial\phi}{\partial z} \tag{21}$$

For brevity, the split (in y) operator A_2, its discrete analogue A_2^h as well as their adjoints A_2^* and $(A_2^h)^*$ are not given

(14), each 1-D split operator A_i $(i=1,2,3)$ is positive semidefinite [13]. The operator of the problem (16)-(19),(11) is the adjoint of A, and can be presented as the sum $A^*=A_1^*+A_2^*+A_3^*$ where

$$A_1^* g = -\frac{1}{2}\frac{\partial}{\partial x}(ug) - \frac{1}{2}u\frac{\partial g}{\partial x} - \frac{\partial}{\partial x}\mu\frac{\partial g}{\partial x}, \qquad A_3^* g = -\frac{1}{2}\frac{\partial}{\partial z}(wg) - \frac{1}{2}w\frac{\partial g}{\partial z} + \sigma g - \frac{\partial}{\partial z}\nu\frac{\partial g}{\partial z} \tag{22}$$

The problems (10)-(14) and (16)-(19) are solved with the splitting method [9],[16]. Suppose, without loss of generality, that $\mu=\mu(z)$, and define the net functions on different grids: $\phi_{ijk}\equiv\phi(x_i,y_j,z_k)$, $u_{ijk}\equiv u(x_{i-1/2},y_j,z_k)$, $v_{ijk}\equiv v(x_i,y_{j-1/2},z_k)$, $w_{ijk}\equiv w(x_i,y_j,z_{k-1/2})$, $\mu_k\equiv\mu(z_k)$, $\nu_{ijk}=\nu(x_i,y_j,z_{k-1/2})$. The 2-nd order discrete approximations of A_i and the continuity equation (11) have the form (the indices i,j,k that do not vary are omitted)

$$(A_1^h\phi)_{ijk} = \frac{1}{2\Delta x}[u_{i+1}\phi_{i+1} - u_i\phi_{i-1}] - \frac{\mu_k}{(\Delta x)^2}[\phi_{i+1} - 2\phi_i + \phi_{i-1}] \tag{23}$$

$$(A_3^h\phi)_{ijk} = \frac{1}{2\Delta z}[w_{k+1}\phi_{k+1} - w_k\phi_{k-1}] + \sigma\phi_k - \frac{1}{(\Delta z)^2}[\nu_{k+1}(\phi_{k+1} - \phi_k) - \nu_k(\phi_k - \phi_{k-1})] \tag{24}$$

$$[u_{i+1} - u_i]/\Delta x + [v_{j+1} - v_i]/\Delta y + [w_{k+1} - w_k]/\Delta z = 0 \tag{25}$$

The adjoint operators $(A_i^h)^*$ are obtained with the substitution of $-u,-w,g$ for u,w,ϕ in (23) and (24). As to the boundary conditions, we give only one example (see [13] for more details). Let u_{ijk} be a positive value of the u-component of the velocity at the left boundary point $M=(x_{1/2},y_j,z_k)$ of the grid domain. Then $U_n=-u_{ijk}<0$ and M belongs to S^-, and the first conditions (12) and (17) are approximated as

$$\mu_k(\phi_{0jk} - \phi_{1jk})/\Delta x + u_{1jk}(\phi_{0jk} + \phi_{1jk})/2 = 0, \qquad g_{0jk} = g_{1jk} \tag{26}$$

For any $i=1,2,3$, the discrete operators A_i^h and $(A_i^h)^*$ are positive semidefinite, and all of them are skew-symmetric if $\mu=\sigma=0$, $\nu=0$, and S is a coast line ($U_n=0$ everywhere at S). Within each double time step interval $(t_n-\tau,t_n+\tau)$ the main and adjoint numerical schemes have the form

$$\Phi[n-\tfrac{3-i}{3}] - \Phi[n-\tfrac{4-i}{3}] = -\frac{\tau}{2}A_i^h(\Phi[n-\tfrac{3-i}{3}] + \Phi[n-\tfrac{4-i}{3}]) \quad (i=1,2)$$

$$\Phi[n+\tfrac{1}{3}] - \Phi[n-\tfrac{1}{3}] = -\tau A_3^h(\Phi[n+\tfrac{1}{3}] + \Phi[n-\tfrac{1}{3}]) + 2\tau q[n]$$

$$\Phi[n+\tfrac{4-i}{3}] - \Phi[n+\tfrac{3-i}{3}] = -\frac{\tau}{2}A_i^h(\Phi[n+\tfrac{4-i}{3}] + \Phi[n+\tfrac{3-i}{3}]) \quad (i=2,1) \tag{27}$$

and

$$G[n+\tfrac{3-i}{3}] - G[n+\tfrac{4-i}{3}] = -\frac{\tau}{2}A_i^h(G[n+\tfrac{3-i}{3}] + G[n+\tfrac{4-i}{3}]) \quad (i=1,2)$$

$$G[n-\tfrac{1}{3}] - G[n+\tfrac{1}{3}] = -\tau A_3^h(G[n-\tfrac{1}{3}] + G[n+\tfrac{1}{3}]) + 2\tau p[n]$$

$$G[n-\tfrac{4-i}{3}] - G[n-\tfrac{3-i}{3}] = -\frac{\tau}{2}A_i^h(G[n-\tfrac{4-i}{3}] + G[n-\tfrac{3-i}{3}]) \quad (i=2,1) \tag{27}$$

where F, G and q, p are the vectors representing grid values of the solutions ϕ, g at fractional time steps, and grid values of the forcings Q, P at the moment t_n, respectively [13].

Summary

The movement and spreading of the oil spilling from a damaged oil tanker are considered in a limited sea area when there is an oil flow through the liquid boundaries. The 2-D and 3-D oil transport-diffusion problems are formulated, and equivalent direct and adjoint estimates of the average oil concentration in ecologically sensitive zones are derived. As the direct estimate is based on the oil transport problem solution, the adjoint transport problem solution is required to make the adjoint estimate. Because of special boundary conditions, both the main and adjoint problems are well-posed according to Hadamard, that is, any solution of either problem is unique and stable to initial perturbations. These conditions are reduced to the well-known and natural boundary conditions not only in the non-diffusion limit (pure advection problem), but also in the case of the closed sea basin having the coast line as its boundary.

The dual estimates complement each other nicely in studying the consequences of the oil spill. While the direct estimate (8) is preferable if a comprehensive oil information is required in the whole domain *D*, or in many its zones, the adjoint estimate (9) explicitly relates the average oil concentration in a zone to the oil spill rate through the adjoint solution at the accident site. Therefore it is very convenient to study oil concentration variations caused by variations in the oil spill rate or/and the accident point. Several examples given in [14] show how to decide between dual estimates in various situations. The adjoint estimate can be modified for the prediction purpose [14]. Note that the case of a few ecologically sensitive zones is analyzed in the perfect analogy to that considered here. The dual estimates can also be applied if oil enters the marine environment from natural sources, offshore production, waste waters and etc.

Balanced, absolutely stable 2nd-order finite-difference schemes based on the splitting method are given for the solution of the main and adjoint transport problems. In the absence of the model dissipation and sources, each scheme has two conservation laws. The 1-D split discrete equations obtained at every fractional step of the numerical algorithm are solved by the factorization method.

References

1. Doerffer, J.W., Oil Spill Response in the Marine Environment. Pergamon Press, Oxford, 1992.
2. Elliott, A.J., Shear diffusion and the spread of oil in the surface layers of the North Sea. Deutsche Hydrographische Zeitung, 39 (1986), 113 - 137.
3. Elliott, A.J., Dale, A.C. and Proctor, R., Modelling the movement of pollutants in the UK shelf seas. Marine Pollution Bulletin, 24 (1992), 614 - 619.
4. Godunov, S.K., Equations of Mathematical Physics. Nauka, Moscow, 1971.
5. Hadamard, J., Lectures on Cauchy's Problem in Linear Partial Differential Equations. Yale University Press, 1923.
6. Lyusternik, L. and Sobolev, V., Elements of Functional Analysis. Ungar, New York, 1964.
7. Marchuk, G.I., Mathematical Models in Environmental Problems. Elsevier, New York, 1986.
8. Marchuk, G.I., Adjoint Equations and Analysis of Complex Systems. Kluwer, Dordrecht, 1995.
9. Marchuk, G.I. and Skiba, Yu.N., Numerical calculation of the conjugate problem for a model of the thermal interaction of the atmosphere with the oceans and continents. Izvestiya, Atmospheric and Oceanic Physics, 12 (1976), 279-284.
10. Poinsot, T.J. and Lele, S.K., Boundary conditions for direct simulations of compressible viscous flows. Journal of Computational Physics, 101 (1992), 104 - 129.
11. Proctor, R., Elliott, A.J., Dale, A.C and Flather, R.A., Forecast and hindcast simulations of the Braer oil spill. Marine Pollution Bulletin, 28 (1994), 212 - 229.
12. Schneider, D., Oil spill has long-term effects. Water Environment Technology, 5 (1993), 30 - 32.
13. Skiba, Yu.N., Balanced and absolutely stable implicit schemes for the main and adjoint pollutant transport equations in limited area. Revista Internacional de Contaminación Ambiental, 9 (1993), 39 - 51.
14. Skiba, Yu.N., Dual oil concentration estimates in ecologically sensitive zones. Environmental Monitoring and Assessment, 43 (1996), 139 - 151.
15. Skiba, Yu.N., The derivation and applications of the adjoint solutions of a simple thermodynamic limited area model of the atmosphere-ocean-soil system. World Resource Reviews, 8 (1966), 98 - 113.
16. Skiba, Yu.N., Adem, J. and Morales-Acoltzi, T., Numerical algorithm for the adjoint sensitivity study of the Adem ocean thermodynamic model. Atmósfera, 9 (1996), 147 - 170.
17. Wolff, G.A., Preston, M.R., Harriman, G. and Rowland, J.S., Some preliminary observations after the wreck of the oil tanker Braer in Shetland. Marine Pollution Bulletin, 26 (1993), 567 - 571.

WAVELET MODELLING OF HIGH RESOLUTION RADAR IMAGING AND CLINICAL MAGNETIC RESONANCE TOMOGRAPHY

Walter Schempp
Lehrstuhl für Mathematik I
Universität Siegen
D-57068 Siegen, Germany

Abstract. The speed with which clinical magnetic resonance imaging (MRI) systems spread throughout the world was phenomenal. Coherent wavelets allow for a unified model of the multichannel perfect reconstruction analysis–synthesis filter bank of high resolution radar imaging and MRI. The geometric quantization construction of matched bank filters depends upon the Kepplerian spatiotemporal strategy which succeeds in the synchronous and stroboscopic summation over phase histories in local frequency encoding channels. The Kepplerian planetary clockwork of quantum holography is implemented in symplectic affine planes by Fourier analysis of the Heisenberg nilpotent Lie group G, and the associated reconstructing operational calculus on the selected energetic stratum of the unitary dual $\hat{G}$ or the symbolic calculus of the $C^{\star}$–algebra of G. The neural network performed by quantum holograms allows for localization of cortical activations of the human brain.

A radar system employs a directional antenna that radiates energy within a narrow beam in a known direction. One unique feature of the synthetic aperture radar (SAR) imaging modality is that its spatial resolution capability is independent of the platform altitude over the subplatform nadir track (Figure 1). This is a result of the fact that the SAR image is formed by simultaneously storing the phase histories and the differential time delays in local frequency encoding subbands of wideband radar, none of which is a function of the range from the radar sensor to the scene. It is this unique capability which allows the acquisition of high resolution images from satellite altitude as long as the received echo response has sufficient strength above the noise level.

The Kepplerian spatiotemporal strategy of physical astronomy succeeds in the synchronous and stroboscopic summation over phase histories in local frequency encoding channels, and suggests the implementation of a matched filter bank by orbit stratification in a symplectic affine plane. Application of this procedure leads to the landmark observation of the earliest SAR pioneer, Carl A. Wiley, that motion is the solution of the high resolution radar imagery and phased array antenna problem of holographic recording. Whereas the Kepplerian spatiotemporal strategy transferred to quantum holography is realized in SAR imaging by the range Doppler principle [1], [8], it is the solvable affine Lauterbur encoding principle which takes place in clinical MRI [3], [12]. At the background of both high resolution imaging techniques lies the construction of a multichannel coherent wavelet perfect reconstruction analysis–synthesis filter bank of matched filter type [2]. Beyond these applications to local frequency encoding subbands, the Kepplerian spatiotemporal strategy leads to the concept of Feynman path integral or summation over phase histories.

As approved by quantum electrodynamics and photonics [7], [14], geometric quantization allows for a semi–classical approach to the interference pattern of quantum holography [12]. Indeed, the unitary dual $\hat{G}$ of the Heisenberg nilpotent Lie group G consisting of the equivalence classes of irreducible unitary linear representations of G allows for a coadjoint orbit fibration by symplectic affine planes $\mathcal{O}_{\nu}$ ($\nu \neq 0$) spatially located in tomographic slices inside the vector space dual $\mathrm{Lie}(G)^{\star}$ of the real Heisenberg Lie algebra $\mathrm{Lie}(G)$ [9]. This is a consequence of the Kirillov homeomorphism

$$\hat{G} \longrightarrow \mathrm{Lie}(G)^{\star}/\mathrm{CoAd}_{G}(G)$$

which is at the basis of the geometric quantization. In terms of standard coordinates, the Heisenberg group G consists of the set of unipotent matrices

$$\left\{ \begin{pmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} \,\middle|\, x, y, z \in \mathbf{R} \right\}$$

under the matrix multiplication law. If the unipotent matrices $\{P, Q, I\}$ denote the canonical basis of the three-dimensional real vector space Lie(G), where

$$\exp_G P = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \exp_G Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \exp_G I = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

holds under the matrix exponential diffeomorphism $\exp_G : \mathrm{Lie}(G) \longrightarrow G$, the coadjoint action of G on Lie$(G)^*$ is given by

$$\mathrm{CoAd}_G \begin{pmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & -y \\ 0 & 1 & x \\ 0 & 0 & 1 \end{pmatrix}.$$

Therefore the action CoAd$_G$ reads in terms of the coordinates $\{\alpha, \beta, \nu\}$ with respect to the dual basis $\{P^*, Q^*, I^*\}$ of the real vector space dual Lie$(G)^*$ as follows:

$$\mathrm{CoAd}_G \begin{pmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} (\alpha P^* + \beta Q^* + \nu I^*) = (\alpha - \nu y)P^* + (\beta + \nu x)Q^* + \nu I^*.$$

The linear varieties

$$\mathcal{O}_\nu = \mathrm{CoAd}_G(G)(\nu I^*) = \mathbf{R}P^* + \mathbf{R}Q^* + \nu I^* \quad (\nu \neq 0)$$

actually are symplectic affine planes in the sense that they are in the natural way compatibly endowed with both the structure of an affine plane and a symplectic structure. Their rotational curvature forms are exactly the standard symplectic form of $\mathbf{R}\oplus\mathbf{R}$, dilated by the frequency ν. The corresponding equivalence classes of irreducible unitary linear representations U^ν of G acting on the standard complex Hilbert space $L^2_{\mathbf{C}}(\mathbf{R})$ of square integrable wave functions on the bi-infinite time scale $\mathbf{R}$ are infinite dimensional and can be realized as Hilbert-Schmidt integral operators with kernels K^ν in $L^2_{\mathbf{C}}(\mathbf{R} \oplus \mathbf{R})$ [9]. Therefore the symplectic affine planes $\mathcal{O}_\nu$ $(\nu \neq 0)$ in Lie$(G)^*$ are contiguous, adjacently decoupled energetic strata, predestinate to implement the Kepplerian spatiotemporal strategy, and to carry quantum holograms [10].

- In radar imaging, $\nu \neq 0$ denotes the center frequency of the transmitted pulse train, whereas in clinical MRI ν is the frequency of the rotating coordinate frame defined by tomographic slice selection of the energetic stratum.

The confocal plane $\nu = 0$ in Lie$(G)^*$ consists of the single point orbits

$$\left\{\epsilon_{(\alpha,\beta)} \,\middle|\, (\alpha, \beta) \in \mathbf{R} \oplus \mathbf{R}\right\},$$

corresponding to the one-dimensional representations of G, or unitary characters of G/C. As the reconstruction plane it plays a fundamental role in the coherent optical processing of radar data [1], morphological MRI, and neurofunctional MRI detecting for the recording of brain activities [12]. It follows from this classification of the coadjoint orbits of G in Lie$(G)^*$ the highly remarkable fact that there exists no finite dimensional irreducible unitary linear representation of G having dimension > 1. Hence the irreducible unitary linear representations of G which are not characters are infinite dimensional. Their coefficient functions define the holographic transforms.

Inspection of the spectrum of G reveals a continuous open projection onto C, taking a copy of $\mathbf{R} \oplus \mathbf{R}$ onto $\{0\}$ and a single point onto each non-zero $\nu \in C$. Hence, the C^*-algebra of the Heisenberg group G is the algebra of continuous sections vanishing at infinity of a continuous field of C^*-algebras $(\mathcal{A})_{\nu\in\mathbf{R}}$, with $\mathcal{A}_\nu$ the quotient corresponding to the closed subset mapping into $\{\nu\}$. Since all points of the copy of $\mathbf{R} \oplus \mathbf{R}$ which maps into $\{0\}$ are one-dimensional representations of G, it follows $\mathcal{A}_0 \cong \mathcal{C}_0(\mathbf{R} \oplus \mathbf{R})$. For each $\nu \neq 0$, since the spectrum of $\mathcal{A}_\nu$ is a single point, and the irreducible unitary linear representation corresponding to the coadjoint orbit $\mathcal{O}_\nu$ is infinite-dimensional, it follows that $\mathcal{A}_\nu$ is isomorphic to the simple C^*-algebra of compact operators acting on the standard complex Hilbert space $L^2_{\mathbf{C}}(\mathbf{R})$. It includes the Hilbert- Schmidt operators on $L^2_{\mathbf{C}}(\mathbf{R})$ as a norm dense ideal.

Let C denote the one-dimensional center of G transversal to the plane carrying the spin excitation profiles or quantum holograms (Figure 2). Then

$$C = [G, G] = \mathbf{R}.\exp_G I$$

is spanned by the central transvection $\exp_G I$. In coordinate-free terms, G forms the non-split central group extension

$$C \lhd G \longrightarrow G/C$$

where G/C is transversal to the line C, and bicontinuously isomorphic to the symplectic affine plane $\mathbf{R} \oplus \mathbf{R}$.

- The group of automorphisms of G which induce the identity on the center C of G is isomorphic to the semi-direct product of the special linear group $\mathbf{SL}(2, \mathbf{R})$ and $\mathbf{R} \oplus \mathbf{R}$.

The irreducible unitary linear representations of G associated to the coadjoint orbit $\mathcal{O}_\nu$ are square integrable mod C. Indeed, it is well known that a coadjoint orbit is a linear variety if and only if one (and hence all) of the corresponding irreducible unitary linear representations is square integrable modulo its kernel. It is reasonable to regard square integrability as an essential part of the Stone-von Neumann theorem of quantum mechanics, because a representation of a nilpotent Lie group is determined by its central character χ_ν if and only if it is square integrable modulo center. Thus χ_ν allows for selection of the tomographic slice $\mathcal{O}_\nu$ with coordinate frame rotating at frequency $\nu \neq 0$.

In analogy to the Kepplerian conchoid construction [13], the C^*-algebra of the Heisenberg group G may be naturally viewed as the crossed product of $\mathcal{C}_0(\mathbf{R} \oplus \mathbf{R})$ by the action of the central factor C which in each line $\{\nu x\} \times \mathbf{R}$ parallel to the second axis of $\mathbf{R} \oplus \mathbf{R}$ consists of translation, but scaled by the first coordinate νx, so that each point of the second coordinate axis itself is left fixed under the scaling procedure.

- The kernel function $K^\nu \in L^2_{\mathbf{C}}(\mathbf{R} \oplus \mathbf{R})$ associated to the central character χ_ν represents a planar spin excitation profile or quantum hologram. It defines a multichannel coherent wavelet perfect reconstruction analysis-synthesis filter bank of matched filter type. The reconstruction of the phase histories in local frequency encoding subbands of K^ν is performed by the symplectically reformatted two-dimensional Fourier transform.

The kernel functions admit the Hermitian symmetry

$$K^\nu(x, y) = \bar{K}^\nu(y, x) \quad ((x, y) \in \mathbf{R} \oplus \mathbf{R})$$

so that the associated Hilbert-Schmidt integral operators are self-adjoint. It is this kernel symmetry which reflects the symmetry inherent to the geometric quantization approach. The compact self-adjoint integral operator defined by the Hermitian kernel $K^\nu \in L^2_{\mathbf{C}}(\mathbf{R} \oplus \mathbf{R})$ has a singular value decomposition associated to to its discrete spectrum as a least square fit. The Karhunen-Loève basis is known to form the minimizer of non-negligible amplitudes in the computational realization of the planar spin excitation profiles or quantum holograms (Figure 2) attached to U^ν. The Karhunen-Loève transform (KLT) can be applied in order to reparametrize the spatial coordinates of the quantum hologram via the normalizing action of the metaplectic group $\mathbf{Mp}(2, \mathbf{R})$ which projects onto $\mathbf{SL}(2, \mathbf{R})$ with kernel $\mathbf{Z}/2\mathbf{Z}$.

The KLT both decorrelates the input and optimizes the redistribution of the wavelet energy in the L^2-sense. As an adaptive image transform algorithm [16] it allows for extraction of the prominent features of the final image with the fewest number of measurements. Such a block compression design performed by the best zonal sampler is of particular importance for use in hospital picture and archiving systems (PACS).

The geometric quantization approach leads to non-locality phenomenon of quantum mechanics [10], and to major application areas of pulsed signal recovery methods, the corner turn algorithm in the digital processing of high resolution SAR data [15], the spin-warp procedure in clinical MRI [6], [11], and finally

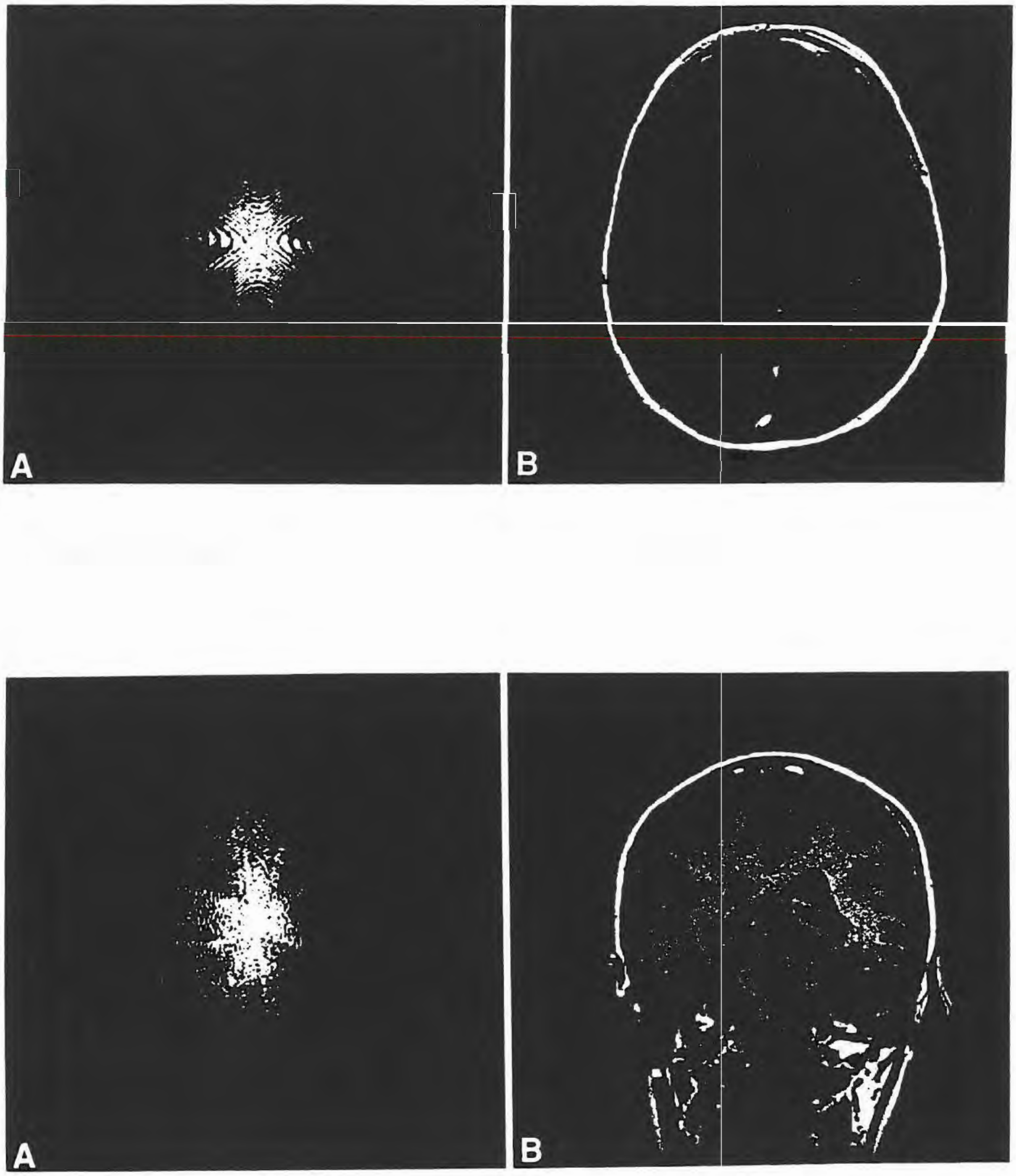


Figure 2: Quantum Holograms (A) and their Reconstructions (B): Transverse and Coronal Tomographic Slices of the Human Brain

to the variants of the ultra-high-speed echo-planar imaging (EPI) technique of neurofunctional MRI [12], and MRI-guided interventions [5]. The capacity of MRI to obtain both structural and functional information promises to dramatically alter the way of studying the human brain *in vivo*. MRI-guided surgical management recently allowed for the first time to perform brain operations without craniotomy. Combined with multi-slice acquisition, it is the spin-warp version of Fourier transform MRI which is used almost exclusively in current routine clinical examinations [4] and now forms a general basis for the majority of MRI studies and preoperative assessment [3], [12].

References

[1] L.J. Cutrona, E.M. Leith, L.J. Porcello, and W.E. Vivian, On the application of coherent optical processing techniques to synthetic-aperture radar. Proc. IEEE 54, 1026-1032 (1966)

[2] E.R. Davies, Electronics, Noise and Signal Recovery. Academic Press, London, San Diego, New York 1993

[3] D.G. Gadian, NMR and its Applications to Living Systems. Second edition, Oxford University Press, Oxford, New York, Tokyo 1995

[4] R.J. Gillies (Editor), NMR in Physiology and Biomedicine. Academic Press, San Diego, New York, Boston 1994

[5] F.A. Jolesz, MRI-guided interventions. The Coolidge Sci. Rev. 2, 1-25 (1994)

[6] P.J. Keller, How big is k-space? Int. J. Neuroradiology 2, 274-289 (1996)

[7] S. Mallick, H. Rajbenbach, Photorefractive nonlinear optics and optical computing. In: Optical Phase Conjugation, M. Gower, D. Proch, editors, pp. 342-363, Springer-Verlag, Berlin-Heidelberg New York 1994

[8] A.W. Rihaczek, S.J. Hershkowitz, Radar Resolution and Complex-Image Analysis. Artech House, Boston, London 1996

[9] W. Schempp, Harmonic Analysis on the Heisenberg Nilpotent Lie Group, with Applications to Signal Theory. Pitman Research Notes in Mathematics Series, Vol. 147, Longman Scientific and Technical, London 1986

[10] W. Schempp, Geometric analysis: The double-slit interference experiment and magnetic resonance imaging. Cybernetics and Systems '96, Vol. 1, pp. 179-183, Austrian Society for Cybernetic Studies, Vienna 1996

[11] W. Schempp, Wavelets in high resolution radar imaging and clinical magnetic resonance imaging. Proc. IWISP '96, Manchester, United Kingdom: Third International Workshop on Image and Signal Processing on the Theme of Advances in Computational Intelligence, B.G. Mertzios, P. Liatsis, pp. 73-80, Elsevier, Amsterdam, Lausanne, New York 1996

[12] W. Schempp, Magnetic Resonance Imaging: Mathematical Foundations and Applications. John Wiley & Sons, New York, Chichester, Brisbane (in print)

[13] B. Stephenson, Kepler's Physical Astronomy. Princeton University Press, Princeton, NJ 1994

[14] T. Tschudi, Phase conjugation in optical signal processing. In: Optical Phase Conjugation, M. Gower, D. Proch, editors, pp. 364-380, Springer-Verlag, Berlin-Heidelberg New York 1994

[15] D.R. Wehner, High Resolution Radar. Artech House, Norwood, MA and London 1987

[16] M.V. Wickerhauser, Custom wavelet packet image compression design. Proc. IWISP '96, Manchester, United Kingdom: Third International Workshop on Image and Signal Processing on the Theme of Advances in Computational Intelligence, B.G. Mertzios, P. Liatsis, pp. 47-52, Elsevier, Amsterdam, Lausanne, New York 1996

APPLICATION OF MATHEMATICAL MODELING FOR SUBSTANTIATION OF THE COMPUTER THEORY OF ONE CLASS OF STRAPDOWN INERTIAL SYSTEMS OF ORIENTATION AND NAVIGATION

P.K.Plotnikov. V.B.Nikishin, A.S. Plotnikov
Saratov State Technical University
410054 Russia, Saratov, str. Politechnicheskaya 77
(E-mail: pribor@sstu.saratov.su; alex@gyro.pvrr.ru)

Abstract. In first part of the report elements of the theory of one class of strapdown inertial systems of orientation and navigation (SISON) are stated. In second part of the report a procedure of modeling of SISON functioning is stated, results are resulted and analysis of received results of modeling, proving main states of the theory is given.

Introduction.

The structure of SISON includes the on-board computer (OBC), and also typical for SINS a set of measuring devices, fixed onboard mobile object: three-component gyroscopic measurer of angular speed (TGMAS) and three-component measurer of apparent acceleration (TMAA). The difference of a considered class of SISON from classical SINS is made by algorithms of OBC functioning. It bases on kinematic equations of frame of reference in Euler-Krylov's angles, referred to basic basis, with horizontal correction, generated on signals of TMAA and also referred to basic basis. In result of integration of specified kinematic equations on determined during integration estimations of angles of yaw, pitch and roll of object in OBC is simulated horizon accompanying free in azimuth frame of reference. These equations have described functioning of strapdown inertial system of orientation (SISO) in SISON structure, which is realized independent on functioning of system of navigation, that distinguishes the given class SISON from known. By virtue of it SISO can be applied independently and be based on TGMAS and TMAA of moderate accuracy. At the same time at presence of exact measurers in OBC a problem of navigation can be decided by determination of estimations of absolute angular speeds specified above simulated frame of reference on the basis of SISO information. Then used known algorithms of half-analytical inertial navigation systems, fair for model of movement of mobile object (MO) on Earth sphere, for purpose of determination the geographical longitude and latitude of MO site.

The elements of the computer theory of an offered class SISON include in structure the formulation of principles of construction of the theory, formation of algorithms of OBC functioning, research of their stability, analysis of properties of SISO, opportunity of fulfilment of Shuler's conditions, made on equations of linear approximation and also disturbed motion. After it algorithms of modeling of SISON functioning are formed by setting up of parameters of the Earth motion and MO movement on Earth sphere, program of modeling is developed, modeling, results are printed, then their analysis is made. It is shown, that the mathematical modeling has proved reliability of main states of the SISON theory, already on the base of full equations of motion of SISO.

The part of the report, named "Elements of the computer theory of one class of SISON" is written by P.K.Plotnikov, other parts are written by all co-authors, specified in heading.

Elements of the computer theory of one class of SISON

The MO movement in space is set by a movement of centre of weights "O" and by angles of MO orientation refer to accompanying horizon free in azimuth frame of reference η ($O\eta_1\eta_2\eta_3$) axis $O\eta_2$ of which is directed on vertical of a site (fig. 1), where ξ ($O_E\xi_1\xi_2\xi_3$) - inertialy unrotative frame of reference; Λ, φ - absolute longitude and latitude of a site, thus the geographical longitude λ is a difference $\lambda=\Lambda-u(t-t_0)$, where u - angular speed of the Earth. With MO is associated frame of reference x ($Ox_1x_2x_3$), and Ox_1 - longitudinal, Ox_2 - normal axis of MO. All frames of reference - right orthogonal. The information from TGMAS and TMAA receive by OBC in kind of estimations $\hat{\omega}_{xi}$, $\hat{W}_{xi}$ (i=1,2,3) appropriate components of vectors $\bar{\omega}$ ($\omega_{x1}\omega_{x2}\omega_{x3}$) and $\bar{W}$ ($W_{x1}W_{x2}W_{x3}$) of absolute angular speed of MO and apparent acceleration of a point O on axes of frame of reference x. Turns of frame of reference x refer to frame of reference η we shall describe with help of the conventional scheme of turns [1]

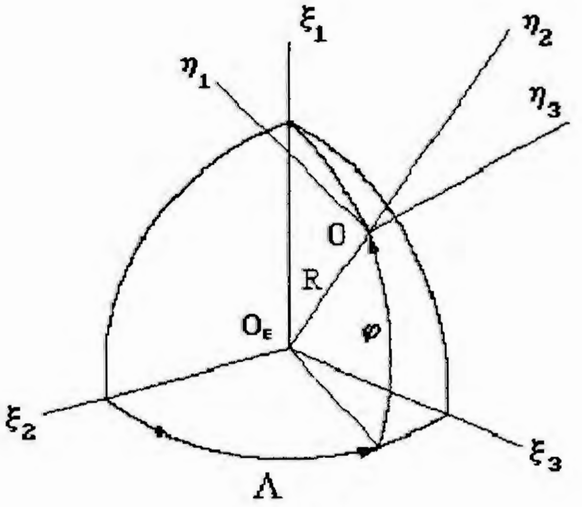


Fig. 1

$$\eta \xrightarrow[O\eta_2, Ox_3, Ox_1]{\psi,\theta,\gamma} x, \tag{1}$$

where ψ, θ, γ - the angles of jaw, pitch and roll, under arrow in (1) are designated axes, around which referred appropriate angles.

SISON task is determination by current primary inertial information $\hat{\omega}_{xi}$, $\hat{W}_{xi}$ (i=1,2,3) estimations $\hat{\lambda}$, $\hat{\varphi}$, but also $\hat{\psi}$, $\hat{\theta}$, $\hat{\gamma}$ by resolving in OBC the equations of orientation and navigation.

The following equations of a problem of determination of orientation MO, received on basis of Euler's kinematic equations, referred to frame of reference η, with entered in them by the members of horizontal correction are in the beginning integrated:

$$\begin{gathered}
\frac{d\hat{\theta}}{dt} = \left(\hat{\omega}_{\eta1} + \omega_{\eta1}^{k}\right)\sin\hat{\psi} + \left(\hat{\omega}_{\eta3} + \omega_{\eta3}^{k}\right)\cos\hat{\psi} \\
\frac{d\hat{\gamma}}{dt} = \left[\left(\hat{\omega}_{\eta1} + \omega_{\eta1}^{k}\right)\cos\hat{\psi} - \left(\hat{\omega}_{\eta3} + \omega_{\eta3}^{k}\right)\sin\hat{\psi}\right]\cos^{-1}\hat{\theta} \\
\frac{d\hat{\psi}}{dt} = \left(\hat{\omega}_{\eta1}\cos\hat{\psi} - \hat{\omega}_{\eta3}\sin\hat{\psi}\right)\tan\hat{\theta} + \hat{\omega}_{\eta2} \\
\omega_{\eta1}^{k} = -K_{\gamma}\frac{\hat{W}_{\eta3}}{g} - \int_{t_0}^{t}\frac{K_{\gamma}^{I}\hat{W}_{\eta3}}{g}d\tau \,; \\
\omega_{\eta3}^{k} = K_{\theta}\frac{\hat{W}_{\eta1}}{g} + \int_{t_0}^{t}\frac{K_{\theta}^{I}\hat{W}_{\eta1}}{g}d\tau \,; \\
\left[\hat{\omega}_{\eta1} \quad \hat{\omega}_{\eta2} \quad \hat{\omega}_{\eta3}\right]^{T} = \hat{A}^{T}\left[\omega_{x1} \quad \omega_{x2} \quad \omega_{x3}\right]^{T};\ \left[\hat{W}_{\eta1} \quad \hat{W}_{\eta2} \quad \hat{W}_{\eta3}\right]^{T} = \hat{A}^{T}\left[W_{x1} \quad W_{x2} \quad W_{x3}\right]^{T}; \\
\hat{A} = \hat{A}_{\gamma}\hat{A}_{\theta}\hat{A}_{\psi},
\end{gathered} \tag{2}$$

Where $\hat{A}$, $\hat{A}_{\gamma}$, $\hat{A}_{\theta}$, $\hat{A}_{\psi}$ - matrix of resulting and elementary turns, appropriate to estimations of angles of turns $\hat{\gamma}$, $\hat{\theta}$ and $\hat{\psi}$ [3]; $K_{\theta},\dots K_{\gamma}^{I}$ - factors of transfer of horizontal correction.

The entry conditions can be any, by modes of an initial alignment on $\hat{\theta}$ And $\hat{\gamma}$ they during functioning SISO will be adjusted independently, the angle $\hat{\psi}$ (t_0) can be given from external source of the information.

The equations of a navigation problem are taken from [1] in view of that the absolute angular speeds $\Delta\omega_{\hat{\eta}_i}$ of frame of reference $\hat{\eta}$, simulated in OBC, are determined by formulas

$$\Delta\omega_{\hat{x}_i} = \hat{\omega}_{xi} - \omega_{xi}\,; \quad \left[\Delta\omega_{\hat{\eta}_1}\Delta\omega_{\hat{\eta}_2}\Delta\omega_{\hat{\eta}_3}\right]^{T} = \hat{A}^{T}\left[\Delta\omega_{x1}\Delta\omega_{x2}\Delta\omega_{x3}\right]^{T}, \tag{3}$$

where $\hat{\omega}_{xi} = \hat{\omega}_{xi}\left(\hat{\psi},\hat{\theta},\hat{\gamma},\dot{\hat{\psi}},\dot{\hat{\theta}},\dot{\hat{\gamma}}\right)$ (i=1,2,3) - estimations of angular speeds of frame of reference $\hat{\eta}$.

At research of stability of the decisions of equations (2) equations of undisturbed and disturbed movements were derived by representation variable $\hat{\gamma}$, $\hat{\theta}$, $\hat{\psi}$, $\hat{\omega}_{xi}$, $\hat{W}_{xi}$ in kind of components of undisturbed and disturbed movement. On equations of an disturbed movement of linear approximation was shown asymptotic stability of undisturbed movement. On same equations Shuler's conditions [2] for SISO were derived.

From principles of construction of the computer theory considered SISON, in report first of all we selecte following: analogy of modeling in OBC dynamic of frame of reference and some real system; maintenance primary inertial and received from OBC information and algorithms, necessary for initial and current (at failures) alignment of this frame of reference.

Mathematical modeling of SISON functioning. Substantiation of basic states of the theory.

Problems are put: a) to confirm that SISO is able to functioning on algorithms (2) with estimation of accuracy of determination of angles of MO orientation at various modes of a movement; b) to confirm an opportunity of fulfilment of Shuler's conditions of indifference (period of fluctuations 84.4 mines.) to action of linear accelerations; c) to confirm asymptotic stability of the solutions on angles $\hat{\theta}$ and $\hat{\gamma}$, and also stability on angle $\hat{\psi}$, on which horizontal correction is entered and there is no correction on angle $\hat{\psi}$; d) to show efficiency of a mode of an initial alignment.

For solving of these problems modes of MO movement to North with speed $V_{\zeta 1}$ were given:

$$\begin{array}{lll} t=0...1200\ s; & V_{\zeta 1}=0; & W_{\zeta 1}=0; \\ t=1200...1220\ s; & V_{\zeta 1}=W_{\zeta 1}t; & W_{\zeta 1}=20\ m/s^2; \\ t=1220...6500\ s; & V_{\zeta 1}=400; & W_{\zeta 1}=0; \\ \varphi=45^0. & & \end{array} \tag{4}$$

Since $1000\ s$ and up to end of modeling there were oscillations according to the law:

$$\begin{array}{l} \psi=0,2Ex\cdot\sin(0,2\pi t+\pi/2); \\ \theta=0,1Ex\cdot\sin(0,314\pi t); \\ \gamma=0,1Ex\cdot\sin(0,2\pi t), \end{array} \tag{5}$$

where $Ex=1-e^{-0,1(t-1000)}$

The modeling is made on computer IBM PC 486, with step of integration *0.05 s*. The primary information in the view of the errors of TGMAS and TMAA are given by the formulas:

$$\hat{W}_{xi} = \left(W_{xi} + \Delta W_{xi}\right)\left(1+\delta W_{xi}\right); \quad \hat{\omega}_{xi} = \left(\omega_{xi} + \Delta\omega_{xi}\right)\left(1+\delta\omega_{xi}\right); \quad (i=1,2,3), \tag{6}$$

where $\Delta\omega_{xi}$, ΔW_{xi} - zero shifts, $\delta\omega_{xi}$, δW_{xi} - errors of factors of transfer, and

$$\begin{array}{ll} \delta W_{x1} = \delta W_{x3} = -\delta W_{x2} = 10^{-5}; & \Delta W_{x1} = \Delta W_{x3} = -\Delta W_{x2} = 10^{-3}\ , m/s^2; \\ \delta\omega_{x1} = \delta\omega_{x3} = -\delta\omega_{x2} = 10^{-5}; & \Delta\omega_{x1} = \Delta\omega_{x3} = -\Delta\omega_{x2} = 5\cdot 10^{-8}\ s^{-1}. \end{array} \tag{7}$$

Factors of transfer of horizontal correction

a) In mode of an alignment

$$K_\gamma = K_\theta = 1,5\cdot 10^{-2}; \qquad K_\gamma^I = K_\theta^I = 5,625\cdot 10^{-3}\ , s^{-2}. \tag{8}$$

b) In mode of normal functioning (Shuler's conditions)

$$K_\gamma = K_\theta = 0; \qquad K_\gamma^I = K_\theta^I = g/R = (1,24\cdot 10^{-3})^2\ , s^{-2} \tag{9}$$

On fig. 2 ...5 are submitted diagrams of errors of SISO, made by results of modeling of MO movement and SISO, that determined by formulas:

$$\Delta\psi = \hat{\psi} - \psi; \quad \Delta\theta = \hat{\theta} - \theta; \quad \Delta\gamma = \hat{\gamma} - \gamma. \tag{10}$$

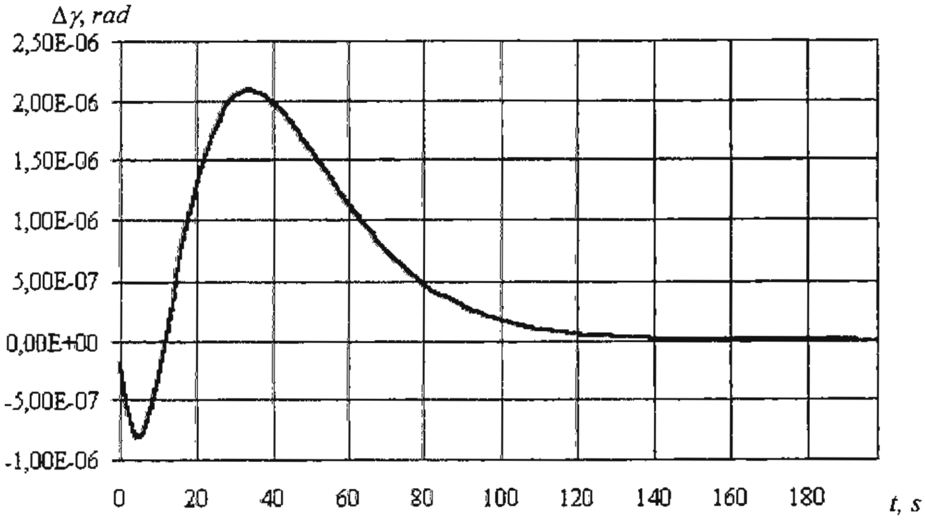


Fig. 2

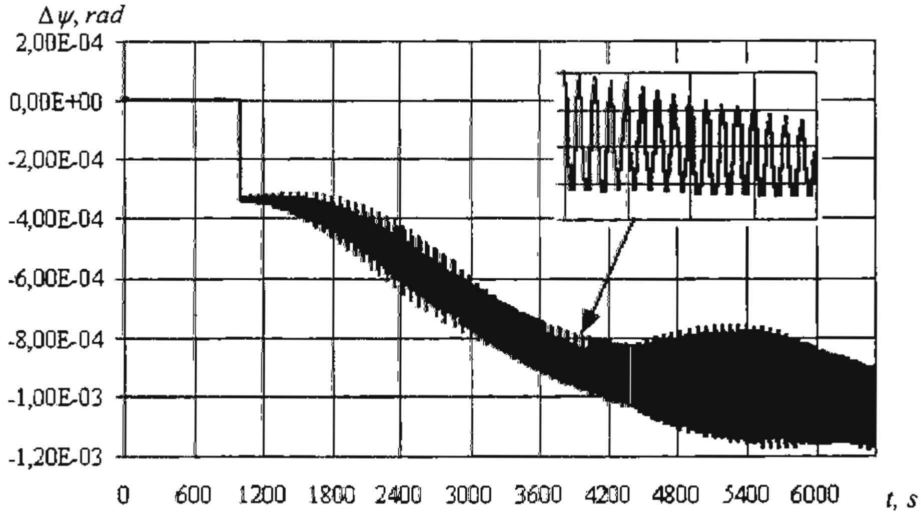


Fig. 3

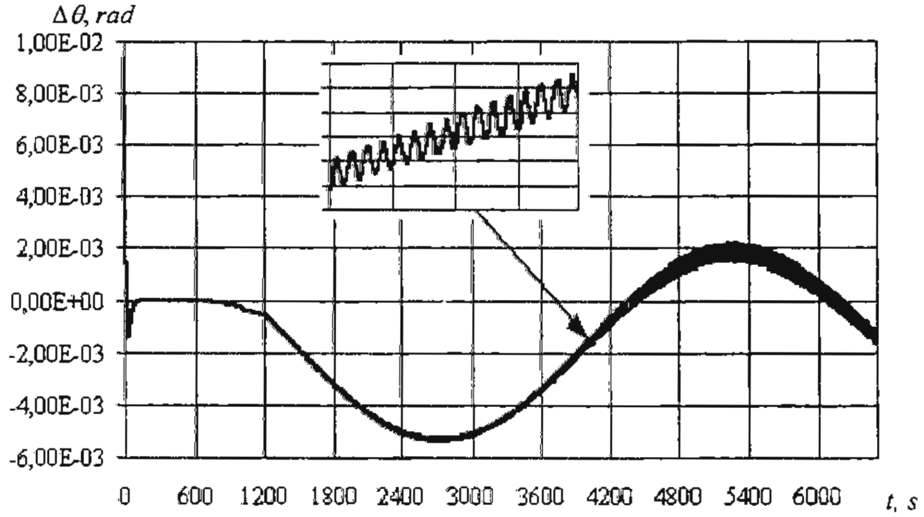


Fig. 4

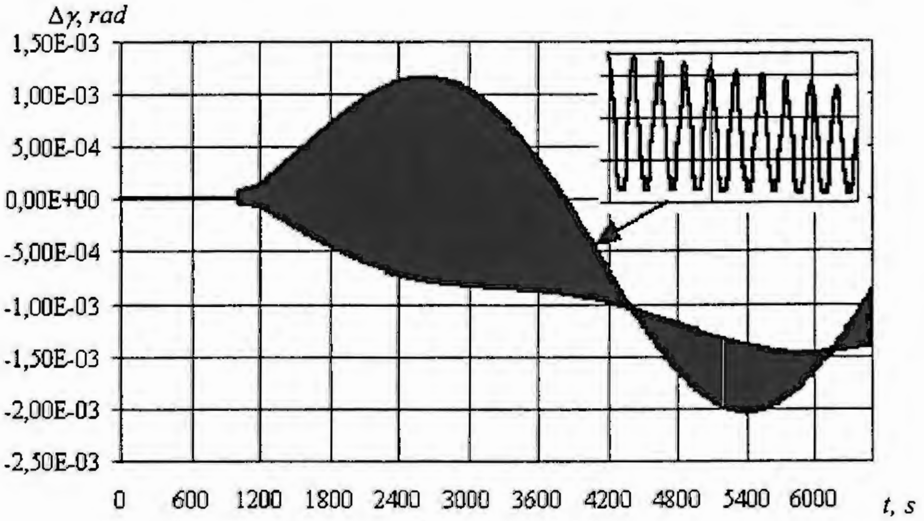


Fig. 5

The diagram on fig. 2 represents reaction SISO by angle $\Delta\gamma$ on influence of non-zero initial conditions at aligning. It is easy to see, that the transient attenuates on exponential law, that testifies the asymptotic stability of the solutions of equations. The similar result has a place by angle $\Delta\theta$. On angle $\Delta\psi$ (fig. 3) processes slowly diverges, that is explained by errors of calculations and presence of shifts of zero in TGMAS. This result does not contradict the theoretical conclusion about stability of a movement on coordinate $\hat{\psi}$. The diagrams on fig. 3 and 4 prove that SISO is able to function on equations (2), because the errors on angle $\Delta\theta$ (which are surpassed the errors on angle $\Delta\gamma$) satisfy for numbers of applications of system in view of errors of gauges and method of calculations. It is easy to see, that since $t=0,6\cdot10^3$ in system are fluctuations with period of Shuler, character of an error is not changed during action of linear acceleration *2g* (*1200<t≤1220 s*). This result confirms the theoretical precondition about efficiency of an undisturbance condition in given class of SISO. The diagrams on fig. 2-4 testify the efficiency of an initial alignment.

The resulted results of modeling, made on full equations (2), prove main states of the theory of one class SISON on SISO part. The modeling of functioning of navigation system was not made, as the properties of this part of SISON all-round are investigated [1], [2].

Conclusion

In report some principles of a computer theory of SISON are formulated, equations of a part of SISON, resolving a problem of determination of orientation angles of MO are introduced. For part of SISON, resolving a navigation problem for MO, known equations of half-analytical navigation systems [1], [2] are used. The new approach thus consists in method of determination of angular speeds of accompanying horizontal free in azimuth frame of reference. Theoretically on linearised equations of disturbed movement questions of stability, accuracy of SISO, Shuler's condition are investigated. Then with help of mathematical modeling the same results are justified, but already on complete equations of SISO, in which errors of TGMAS, TMAA and computing errors of OBC are discounted.

References

1. Ishlinskiy A.Yu. Orientation, gyroscopes and inertial navigation. Science, Moscow, 1976.
2. Kurt Magnus. Kreisel. Theorie und Anwendungen. Springer-Verlag Berlin. Heidelberg. New York, 1971

AN APPLICATION IN POSTAL AUTOMATION: TWO WAYS OF MODELING A TRANSPORT PROCESS

Boris Lohmann
AEG Electrocom, 78459 Konstanz, Germany
e-mail: Boris.Lohmann@aec.aeg.kn.DaimlerBenz.com

Abstract. In modern postal sorting centres, the singulation, address reading, transport and sortation of mail pieces is done automatically by machines which are equipped with automatic address readers. The letter sorting machine considered here, can be modeled either as a discrete event system with the help of a petri net, or as a time continuous system in terms of differential equations. The steps of modeling are illustrated, the pros and cons are discussed, and an approach for throughput control by feedback is outlined.

Introduction

Figure 1 shows the principal components of a modern mail piece sorting machine. With the help of a special *feeding device*, mail items are fed separately into the machine, and are passed across to constantly moving transport belts. The mail items then pass an *image scanner*, which transfers the image information to the *address reading device* (ARD). This ARD can be either an automatic address reader, based on optical character recognition, or a so-called *video coding system*, where the image is displayed on a monitor and the required sorting information is keyed in by an operator. In both cases the ARD processes the images in the sequence of arrival (i.e., FIFO, see [5]) while the mail items move at a constant speed through a *delay line*. The task of this delay line is to provide the ARD with enough time so that the address reading result is available before the mail item reaches the *sorting section* of the machine. In the sorting section the mail items are separated into different stackers or trays.

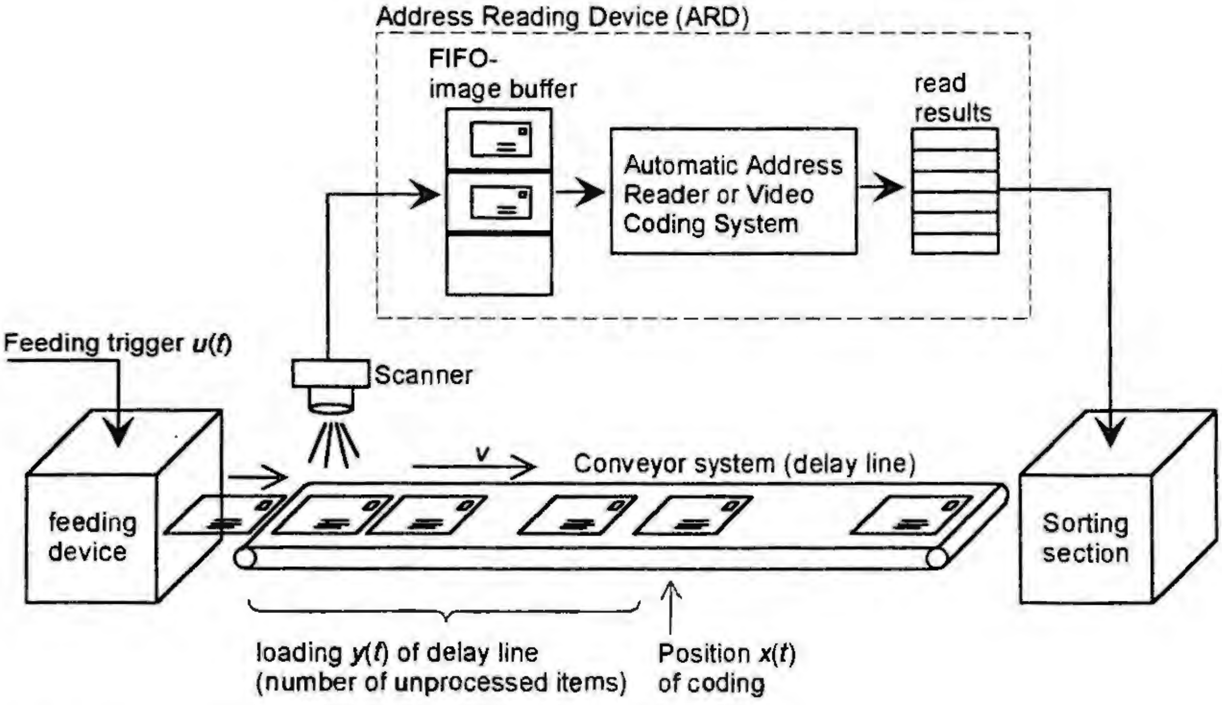


Fig. 1 Components of an automatic postal sorting machine

A modeling of this system is required for the design of a throughput control or for simulation studies in order to predict the performance of different configurations. Obviously, the system is driven by discrete events, namely

- the *feeding of a mail piece*. This is a *control input*, since this event can be triggered from outside, for example by a controller.
- the *coding of a mail piece*, i.e. the event by which the ARD finishes processing of an item and the read result becomes available. This event depends on the structure of the ARD, on the loading situation of the ARD and on the complexity of the individual image being processed. It can be considered as a *disturbance input*, since the time of its occurance cannot be predicted precisely.

The system variables of interest are

- the *coding position* x, i.e. the position of a mail piece at the moment when its coding result becomes available. Sorting of the mail piece is only possible if this coding position is *left* from the sorting section (see Fig. 1).
- the *numbers of mail pieces* being in the different stages of processing, like "uncoded and within delay line", "coded and within delay line", "successfully sorted", "sorting not possible". These give the statistical information on the performance of the machine.

Modeling by a Petry Net

From Fig. 1 and the knowledge of the system, a petry net can be derived quite easily (Fig. 2, see [2] for an overview). T1 is a source transition, representing the feeding of an item to the machine. Concurrent processes follow: the *image processing and address reading process* represented by P2, P3, P4, T2, T3, and the *transport process* represented by P1. The place P1 is a *timed place* which means that each marking entering P1 leaves it after a constant given time T. Depending on wether a read result is available at the end of the delay line or not, either T4 or T5 fires; the inhibitor arc is required for this decision. This firing synchronises the two concurrent processes and starts the sortation.

The place P3 has finite capacity $\kappa = 1$ while all other places can be considered infinite capacity. The two inputs of the system, feeding of items and coding of items, relate to the firing of T1 and T3. The other transitions of the model fire as soon as they are enabled.

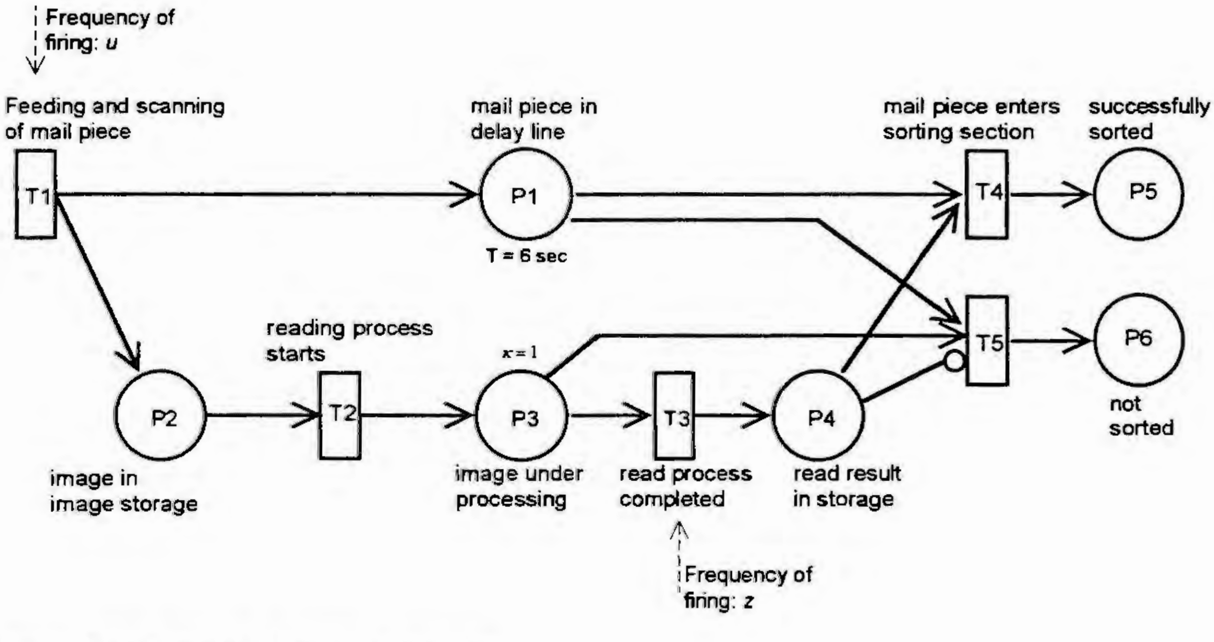


Fig. 2 Petri net model of the sorting machine

The *numbers of mail pieces* being in the different stages of processing are simply given by the numbers of markings in the different places.

The *coding position* x_i of a mail piece *i* can be calculated from the time $t_{i,enter}$ at which T1 fires, and the time $t_{i,code}$ at which the corresponding mail piece is coded, by $x_i = v \cdot (t_{i,code} - t_{i,enter})$. Note that x_i is *not* a time-continuous function, but has to be calculated for each single mail piece. The underlying (time-continuous) differential equation is $\dot{s}(t) = v$, and x_i relates to the end of the considered time interval. This is not represented by the petri net.

Time-Continuous Modeling

If the mail pieces are considered as "dividable" so that a time continuous feeding and coding process results, then a time continuous modeling of the hole process is possible. The input "feeding" is then represented by a *feeding rate u(t)* (in mailpieces per second), and the disturbance "coding" is represented by the *address reading capacity z(t)* (in mailpieces per second). These two "frequencies of events", *u* and *z*, are also illustrated in the petri net, Fig. 2.

The most important system variable is the *position x* within the delay line at which mail items switch from "unprocessed" to "processing completed", see also Fig. 1. In a time-continuous consideration of the process, *x* is no longer related to single mail pieces, but also becomes a time-continuous variable. The velocity $\dot{x}(t)$ of this *position of coding* is

$$\dot{x}(t) = v - \frac{z(t)}{c(x,t)} . \tag{1}$$

The constant transport speed *v* is from left to right. From right to left the ARD proceeds with the *capacity z(t)* (in items/sec). This capacity is divided by the concentration *c* of mail items at *x*, which results in the corresponding velocity. The concentration *c* is

$$c(x,t) = c(0,t - x/v) = \frac{u(t - x/v)}{v} , \tag{2}$$

which is a solution of the *1st order hyperbolic partial differential equation*

$$\frac{\partial c(s,t)}{\partial t} + v \frac{\partial c(s,t)}{\partial s} = 0 \tag{3}$$

with the *boundary condition* and the *initial condition*

$$c(0,t) = \frac{u(t)}{v} , \tag{4}$$

$$c(s,0) = c_0(s) , \tag{5}$$

discribing the distributed parameter system considered here [1, 4, 6]. The variables are

$c(s,t)$ the *concentration* of mail items on the transport belt (measured in pieces per meter),
v the constant *transport speed,*
t the *time,*
s the *spatial coordinate variable,*
$u(t)$ the *feeding rate* (in pieces per second); this is the control input,
$c_0(s)$ the *initial covering* of the transport belt with mail items.

With the solution (2) of equation (3), equation (1) becomes

$$\dot{x}(t) = v \left(1 - \frac{z(t)}{u(t - x/v)} \right) , \tag{6}$$

which is a nonlinear differential equation with the varying delay *x/v* in the input *u*.

Figure 3 shows the structure of the model representing eq. (6). The input u affects the nonlinear block via the varying delay. From the delayed input $u(t)$ and from the address reading capacity $z(t)$ the nonlinear block produces the velocity $\dot{x}(t)$. The state variable x itself defines the varying delay. The system reacts in an unstable manner, for example, on two different but constant values of the inputs u and z.

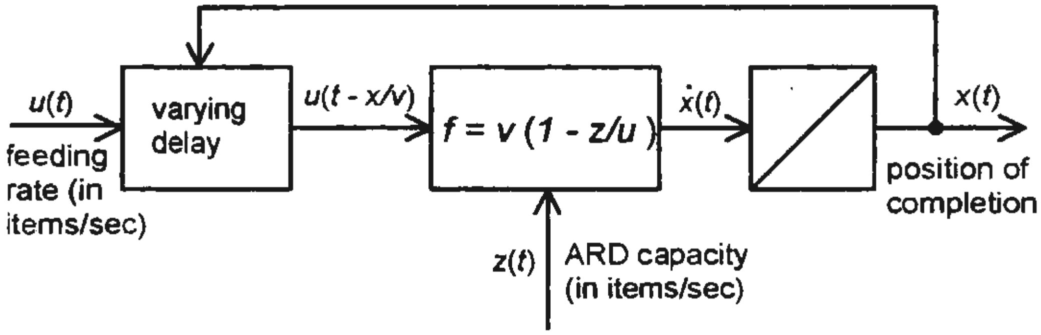


Fig. 3 Part system "coding position x"

Another system variable of interest is the number of mail items being in the delay line and waiting to be coded. In a time-continuous formulation this is a quantity y representing the *loading of the delay line*. For y, the simple differential equation

$$\dot{y}(t) = u(t) - z(t) \text{ for } y \geq 0 \tag{7}$$

is obtained, see Fig 4.

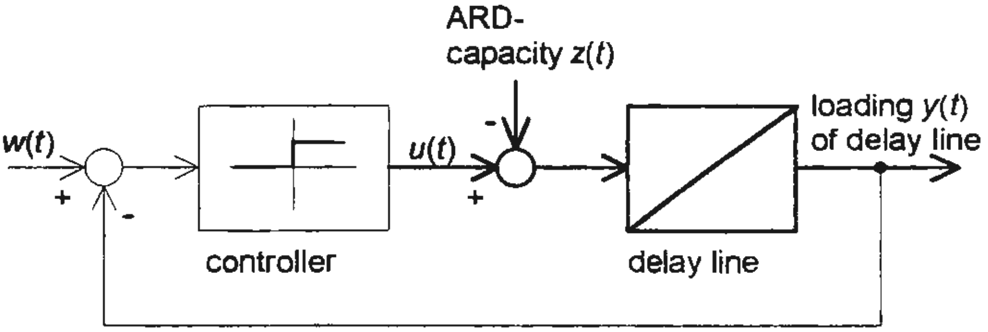


Fig. 4 Part system "loading y of delay line" (thick lines) and stabilizing feedback control (thin lines)

Discussion

Although the technical process considered here is dominated by discrete events, a purely time continuous modeling is possible, since an equivalent continuous interpretation of the process and its input and output variables exists. Nevertheless, the models differ significantly, not only in representation, but also in the usefulness for different tasks like controller design, representation of special situations, and simulation.

1. System insight and representation of system variables: The *position* x of coding, and an important characteristic of the model, the *varying delay*, are only represented in the *continuous* model. The reason for this is that the movement of mail items along the delay line is in fact a continuous process. For the representation of such processes, the petri net would need to be extended in a cumbersome manner. However, the petri net is superior in representing the number of items in the different stages of the process: The number of markings in the different places directly gives the required information. The *unstable behaviour* of the plant can be analized by the petri net as well as by the time continuous model. For instance, if the ARD capacity z is small compared to the feeding capacity u, then, in the petri net the number of markings in P2 will exceed any limits; in the continuous model, the outputs of the integrators in figures 3 and 4 will exceed any limits.

2. Controller Design: The *design of a stabilizing controller* is possible based on either models: the most simple approach based on the petri net would be, to give P2 a limited capacity κ, i.e. the control rule would be: "Feed a mail piece as soon as the number of mail pieces waiting for ARD processing falls short of the value of κ". A corresponding time-continuous control law can easily be found [4], and is illustrated in fig. 4. However, the design of a controller which stabilizes the system *and* keeps x within certain limits of the delay line has only been successful based on the continuous model: the idea presented in [3, 4] is to implement a disturbance feedback by measuring the disturbance z and by choosing the input $w(t)$ of the stabilizing control loop (fig.4) to $w(t) = kz(t)$. In the petri net this would mean to vary the finite capacity κ of the place P2 depending on the frequency z of firing of T3, which is not an obvious approach.

3. Special situations: The continuous model does not correctly represent the fact that y cannot become smaller than 0. The petri net does: the corresponding place runs empty and the subsequent transitions are disabled (which enforces $z = 0$). Furthermore, the continuous model cannot represent the feature that the ARD process is interrupted if the corresponding mail item leaves the delay line without read reasult (i.e., the firing of T5 in Fig. 2 is not represented in the continuous model).

4. Simulation: For the simulation of the model, with or without controllers, a *hybrid* representation is adequate. It includes the petri net as shown in Fig. 2 with an added time continuous layer describing the movement of each single mail piece through the delay line, see Fig. 5. This presupposes that the markings can be distinguished (*coloured* petri net). At *AEG Electrocom* in Konstanz, Germany, the system model, together with different controllers, has been simulated extensively based on this hybrid approach, before the final control algorithm was implemented in the field.

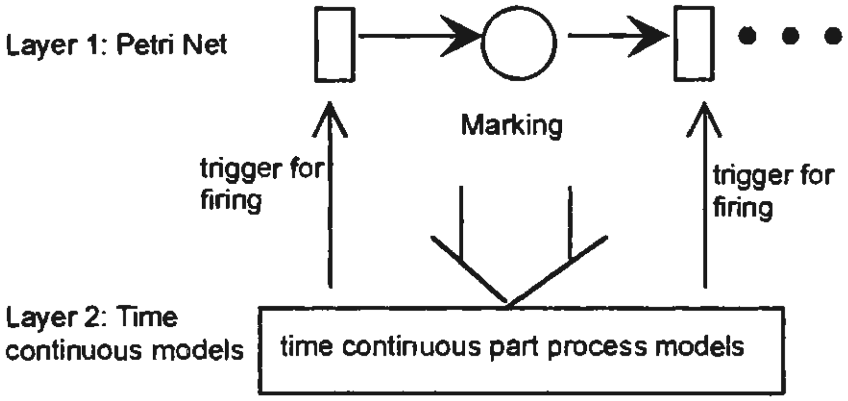

Layer 1: Petri Net
Marking
trigger for firing
trigger for firing
Layer 2: Time continuous models
time continuous part process models

Fig. 5: Hybrid modeling in two layers

References

[1] Ahmed, N.U. and K.L. Teo (1981). *Optimal Control of Distributed Parameter Systems.* North Holland, New York, Oxford.

[2] David, R. and H. Alla (1994). Petri Nets for Modeling of Dynamic Systems - A Survey. *Automatica* **30**, pp. 175-202.

[3] Lohmann, B. (1994). *Verfahren zur Steuerung der Eingabestation für eine Briefsortieranlage.* Patent filed, Germany, P 44 19 430.7

[4] Lohmann, B. (1996): Throughput Control for a Transport Process and Application in Postal Automation Machines.To appear in the IFAC Journal "Control Engineering Practice", Nov. 1996.

[5] Panwalkar, S.S. and W. Iskander (1977). A survey of scheduling rules. *Operations Research* **25**, pp. 45-61.

[6] Ray, W.H. and D.G. Lainiotis (1978). *Distributed Parameter Systems. Identification, Estimation, and Control.* Marcel Dekker, New York, Basel.

In-Ground Effect Inflow Models For Lifting Rotors Using A Finite State Modeling Approach

J.V.R. Prasad[1], Hong Zhang[1] and D. A. Peters[2]
[1]School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA
[2]School of Mechanical Engineering
Washington University
St. Louis, MS

Abstract. Helicopter behavior is significantly affected when flying near the ground due to interactions between rotor wake and the ground plane. Existing ground effect simulation models use a simple factor to modify the out-of-ground effect inflow to get in-ground effect inflow, which is found to be rather approximate. A new ground effect model, which is based on the generalized wake theory is proposed in this study. The inflow at the rotor disc including the effect of ground is represented as a series of normalized Legendre functions of radial station and a series of harmonics of azimuth angle. The magnitude of each term is determined from first-order differential equations in time domain. The coefficients of the differential equations depend on the rotor height above the ground plane, rotor inclination and flight speed. The forcing functions are user-supplied, radial integrals of the blade loading. Initial validation results demonstrate the validity of the proposed model.

Introduction

Modern day helicopters are frequently required to operate close to ground (or ship deck at sea) and their behaviors are greatly affected when flying near the ground surface. The flow around the rotor disc is greatly altered in order to meet the no penetration boundary condition at the ground surface. Since obtaining realistic flow distribution over the rotor disc is a prerequisite to accurate and detailed analysis of helicopter performance and handling qualities, there is a need for developing accurate inflow models that include ground effect. Though comprehensive in-ground effect models based on computational fluid dynamics (CFD) exist in the literature, such CFD models are not very useful for real-time simulation studies. Instead, if one can develop a model in the form of a set of differential equations to describe ground effect on rotor inflow, such a model can easily be integrated with existing helicopter flight simulation programs for studying helicopter flight dynamic behavior when flying close to ground.

Several theoretical and experimental investigations [1-10] have been carried out in the past to understand ground effect on rotor inflow and to develop in-ground effect inflow models. It is conceivable that inflow distribution over the lifting rotor disk should be a function of radial station, azimuth angle and time for the general case. Hence, instead of using a simple factor, a radial and azimuthal as well as time dependent modification function is needed to obtain in-ground effect inflow precisely. A finite state in-ground effect inflow model for the forward flight case is developed in [11] using the generalized wake theory. This study considers extension of the model developed in [11] for the hover case. The organization of the paper is as follows. First, a brief description of the modeling approach for the forward flight case is presented followed by extension of the modeling approach for the hover case. Then, results are presented that illustrate validity of the proposed model for both hover and forward flight cases followed by general conclusions and recommendations.

Ground effect modeling in forward flight

It has already been shown that the pressure distribution of a lifting rotor can be represented in ellipsoidal coordinates as [12]

$$\Phi(\nu,\eta,\bar{\psi},\bar{t}) = -\frac{1}{2}\sum_{m=0}^{\infty}\sum_{n=m+1,m+3,\cdots}^{\infty} \bar{P}_n^m(\nu)\bar{Q}_n^m(i\eta)\;[\tau_n^{mc}(\bar{t})cos(m\bar{\psi}) + \tau_n^{ms}(\bar{t})sin(m\bar{\psi})] \quad (1)$$

where $\bar{P}_n^m(.), \bar{Q}_n^m(.)$ are normalized associated Legendre functions of the first and second kind respectively, τ_n^{mc}, τ_n^{ms} are the disc loading coefficients corresponding to the cosine and sine harmonic distributions, respectively, $(\nu, \eta, \bar{\psi})$ are ellipsoidal coordinates with origin at the rotor center and $\bar{t}$ is nondimensional time.

In order to model ground effect using a pressure potential representation, the following conditions must be met for the new pressure potential Φ :

- Φ must satisfy the basic governing equation (Laplace equation).
- Φ goes to zero at infinity (upstream and downstream).
- Φ renders the desired disk loading.
- Φ must be chosen in such a way that there is no normal component of flow at the ground surface. Suppose two different pressure discontinuities representing rotor and its image are placed in the flow field, pressure distribution of such a system can be obtained through superposition of individual solutions. Thus, the combined pressure potential can be written as

$$\Phi_{Total}(\nu,\eta,\bar{\psi},\bar{t}) = -\frac{1}{2}\sum_{m=0}^{\infty}\sum_{n=m+1,m+3,\cdots}^{\infty} \bar{P}_n^m(\nu_1)\bar{Q}_n^m(i\eta_1)\ [\tau_{n\ 1}^{mc}(\bar{t})cos(m\bar{\psi}_1) + \tau_{n\ 1}^{ms}(\bar{t})sin(m\bar{\psi}_1)]$$

$$+\frac{1}{2}\sum_{m=0}^{\infty}\sum_{n=m+1,m+3,\cdots}^{\infty} \bar{P}_n^m(\nu_2)\bar{Q}_n^m(i\eta_2)\ [\tau_{n\ 2}^{mc}(\bar{t})cos(m\bar{\psi}_2) + \tau_{n\ 2}^{ms}(\bar{t})sin(m\bar{\psi}_2)] \tag{2}$$

where: $(\nu_1, \eta_1, \bar{\psi}_1)$ are ellipsoidal coordinates for the rotor with its origin at the rotor center and $(\nu_2, \eta_2, \bar{\psi}_2)$ are ellipsoidal coordinates for the image rotor disk with its origin at the image rotor center. Using appropriate transformations, one can express the combined pressure potential in terms of a common set of ellipsoidal coordinates $(\nu, \eta, \bar{\psi})$ or a common set of Cartesian coordinates (x,y,z) as shown in Fig. 1.

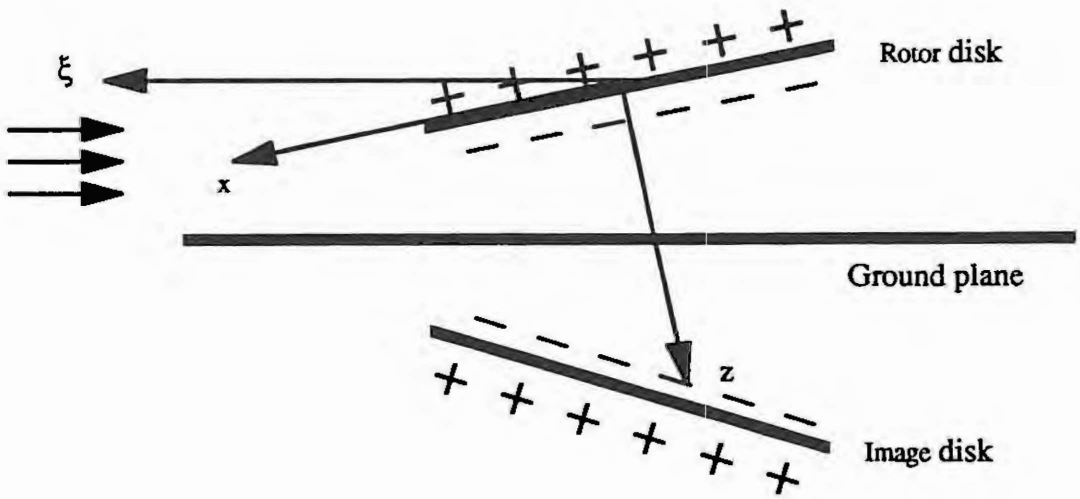


Figure I. Ground effect modeling in forward flight.

The combined pressure potential (Φ_{Total}) can be decomposed into two parts

$$\Phi_{Total} = \Phi_{Total}^{V} + \Phi_{Total}^{A} \tag{3}$$

where Φ_{Total}^{V} is the convection part and Φ_{Total}^{A} is the unsteady part. It can be shown that both parts satisfy Laplace equation and the relation between non-dimensional induced inflow of a lifting rotor and pressure potential can be written as

$$w = -\frac{1}{V}\int_{0}^{\infty}\frac{\partial\Phi_{Total}^{V}}{\partial z}d\xi \tag{4}$$

$$\frac{\partial w}{\partial \bar{t}} = \left.\frac{\partial\Phi_{Total}^{A}}{\partial z}\right|_{\eta=0} \tag{5}$$

where ξ is upstream line from the rotor disk as shown in Fig. 1 and V is the so called mass flow parameter as defined in [6].

If the inflow at the rotor disk is expanded in terms of harmonic variation with respect to rotor azimuth angle (ψ) and a series of associated Legendre function of the first kind of the ellipsoidal coordinate ν as

$$w=\sum_{m=0}^{\infty}\sum_{n=m+1,m+3,..}^{\infty}\frac{\bar{P}_n^m(\nu)}{\nu}[\alpha_n^m\cos m\psi+\beta_n^m\sin m\psi] \tag{6}$$

then, expressions that relate the unknown coefficients in the expansion for inflow at the rotor disk, i.,e., α_n^m, β_n^m with the coefficients of the disk loading harmonic distributions, i.e., τ_n^{mc}, τ_n^{ms} can be obtained by combining Eqs. (2) through (6) as

$$[M]^{-1}\frac{d}{d\bar{t}}\begin{Bmatrix}\cdot\\ \cdot\\ \alpha_n^m\\ \cdot\\ \cdot\\ \beta_n^m\\ \cdot\\ \cdot\end{Bmatrix}+V\begin{bmatrix}\tilde{L}^c & 0\\ 0 & \tilde{L}^s\end{bmatrix}^{-1}\begin{Bmatrix}\cdot\\ \cdot\\ \alpha_n^m\\ \cdot\\ \cdot\\ \beta_n^m\\ \cdot\\ \cdot\end{Bmatrix}=\frac{1}{2}\begin{Bmatrix}\cdot\\ \cdot\\ \tau_n^{mc}\\ \cdot\\ \cdot\\ \tau_n^{ms}\\ \cdot\\ \cdot\end{Bmatrix} \tag{7}$$

Detailed expressions for $[M]$, $[\tilde{L}^c]$ *and* $[\tilde{L}^s]$ are given in [11].

Ground effect modeling in hover

A straightforward extension of the finite state in-ground effect inflow modeling approach described in the previous section to hovering flight is not possible due to the lack of a common freestream for the rotor and its image in order for the superposition of pressure potentials to be valid. However, it is still possible to treat the rotor and its image as two separate pressure discontinuities and superpose the velocities induced by such discontinuities. A straightforward application of the generalized wake theory for computing induced velocities at points that are far away from the rotor disk downstream results in twice the induced velocity at the rotor disk whereas in reality, the induced velocity should go to zero at infinity. Hence, a decay function based on viscous flow theory [13] is needed when computing induced velocity at the rotor disk due to the image rotor. The coefficient of the decay function is adjusted by matching the uniform part of the induced velocity predicted by the proposed model with experimental data [8] when the height of the rotor above the ground equals one rotor radius. The following decay function is used in this study:

$$f_{decay}(z)=e^{-1.35z} \tag{8}$$

Thus, induced inflow at the rotor disk including the decay function is given by

$$w=\frac{1}{V}[-\Phi_1|_{z=0^-}+(2\Phi_2|_{z=2h^-}-\Phi_2|_{z=0})f_{decay}(2h)] \tag{9}$$

Validation and discussion

An initial validation is carried out by simplifying the model and comparing the results predicted from the proposed model with those predicted using experimental data available in the literature. For this purpose, only one radial function and only the first harmonic variations are considered, i.e.,

$$w=\sqrt{3}\alpha_1^0+\frac{\bar{P}_2^1(\nu)}{\nu}[\alpha_2^1\cos\psi+\beta_2^1\sin\psi] \tag{10}$$

or expressed in conventional notation as in the widely used Pitt/Peters dynamic inflow model [6]

$$\lambda=\lambda_0+\frac{r}{R}(\lambda_c\cos\psi+\lambda_s\sin\psi) \tag{11}$$

Denoting the ratio of in-ground-effect to out-of-ground effect uniform inflow components as g, i.e.,

$$g=\frac{(\lambda_0)_{ige}}{(\lambda_0)_{oge}} \tag{12}$$

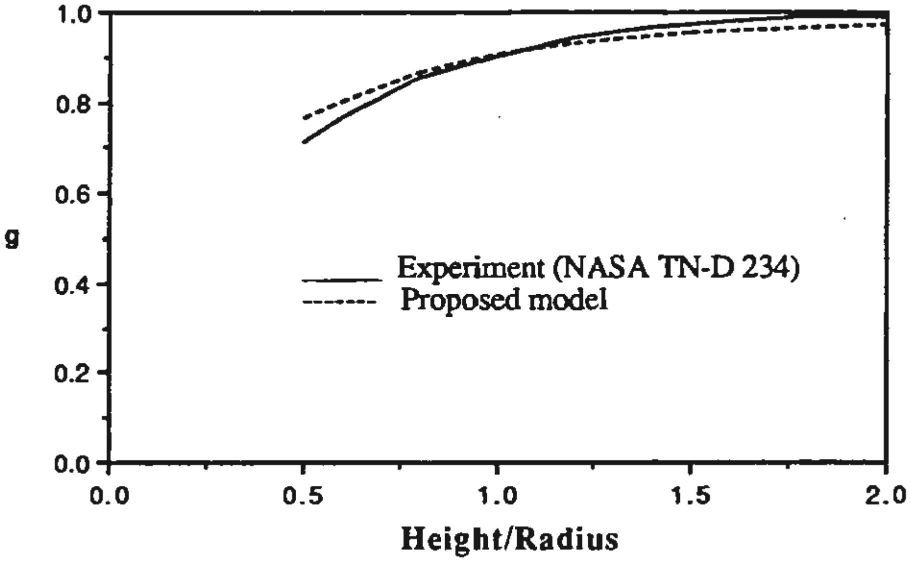


Figure 2. Variation of g with rotor height above the ground for a forward flight case (advance ratio = 0.08)

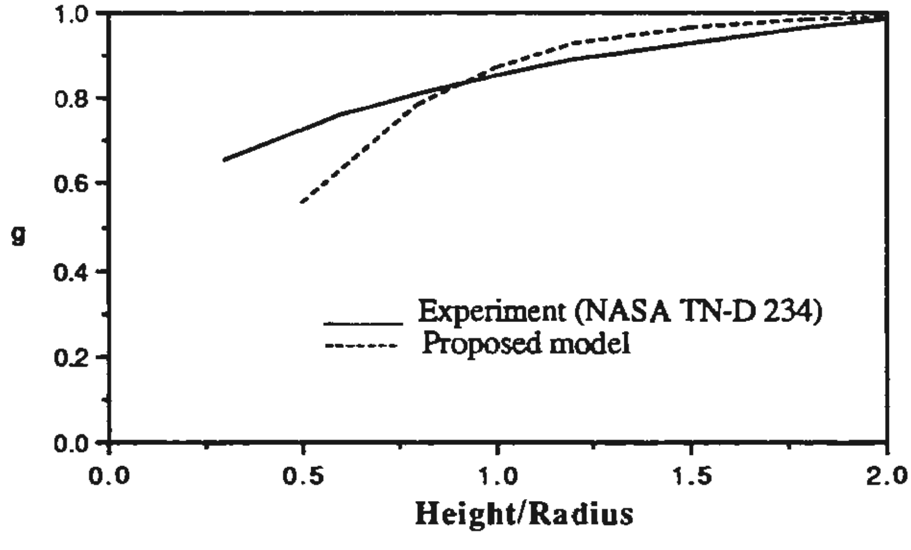


Figure 3. Variation of g with rotor height above the ground for hover case.

Figures 2 and 3 show variation of g with rotor height from the ground as predicted by the proposed model for the forward flight (advance ratio = 0.08) and hover cases, respectively. Also, these figures compare results predicted by the proposed model with experimental data available in the literature [8]. As expected, the predicted results show that ground effect decreases as the rotor height above the ground is increased. Also, the variation of g with rotor height above the ground matches well with experimental data except for small values of height above the ground in hover.

Conclusions

A finite state ground effect model for predicting rotor inflow for the hover case is presented by extending the previously developed forward flight ground effect model using a decay function. Comparisons of uniform component of rotor inflow results obtained using the proposed model with experimental data for both hover and forward flight cases demonstrate the validity of the proposed model. The modeling method presented in this paper considers ground effect in the presence of a flat ground. Further work is needed to extend the finite state inflow modeling approach to the inclined and dynamic ground effect cases to include helicopter flight involving hovering over inclined surfaces or hovering over a pitching, rolling and heaving ship deck.

Acknowledgments. This study is funded under the NRTC Center of Excellence in Rotorcraft Technology program at Georgia Tech.

References

1. Cheeseman, I. C. and Bennett, W. E., The Ground Effect for Lifting Rotor in Forward Flight, British ARC RM No. 3021, 1955.
2. Peters, D.A. and He, C.J., A Closed-Form Unsteady Aerodynamic Theory for Lifting Rotors in Hover and Forward Flight, 43rd Annual Forum of the American Helicopter Society, 1987.
3. Curtiss, H.C.,Jr., et al, Rotor Aerodynamics in Ground Effect at Low Advance Ratios, 37th Annual Forum of the American Helicopter Society, 1981.
4. Curtiss, H.C.,Jr., Erdman, W. and Sun, M., Ground Effect Aerodynamics, Vertica, 11(1987).
5. Gaonkar, G.H. and Peters, D.A., Review of Dynamic Inflow Modeling For Rotorcraft Flight Dynamics, Vertica,12(1988).
6. Peters, D. A. and HaQuang, N., Dynamic Inflow for Practical Applications, Journal of the American Helicopter Society, 33(1988).
7. Pitt, D.M. and Peters, D.A., Theoretical Prediction of Dynamic Inflow Derivatives, Vertica, 5 (1981), 21-34.
8. Heyson, H. H., Ground Effect for Lifting Rotor in Forward Flight, NASA TN D-234, 1960.
9. Hayden, J.S. , The Effect of the Ground on Helicopter Hovering Power Required, 32nd Annual Forum of the American Helicopter Society, 1976.
10. Heyson, H. H., Theoretical Study of the Effect of Ground Proximity on the Induced Efficiency of Helicopter Rotors, NASA TM-X-71951, 1977.
11. Zhang, H., Prasad, J.V.R. and Peters, D.A., Finite State Inflow Models for Lifting Rotors in Ground Effect, 52nd Annual Forum of the American Helicopter Society, 1996.
12. He, C., Development and Application of a Generalized dynamic Wake Theory for Lifting Rotors, Ph.D. Thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, July 1989.
13. Schlichting, H. Boundary Layer Theory. McGraw-Hill Classic Textbook Reissue, Translated by Dr. Kestin, Seventh Edition, 1979.

DISCONTINUOUS-CONTINUOUS MINING SYSTEM OPERATION AS MARKOV PROCESS

V. Pavlovic, University of Belgrade
Djusina 7, 11000 Belgrade

Introduction

Discontinuous-continuous system functioning in opencast mining is a highly complex working environment function represented by mining and geological properties, openpit mine geometry, equipment, operating flow-sheet in excavation, transport and disposal, i.e. storage and production process control. Mass flow on this systems representing the basics for output calculation, as well as operation and renewal times after complex system functioning enables an analysis of system addition to output calculation. System output was defined on the basic of operating reliability calculation of a complex system and its elements connected in series, parallel and combined, functioning of which in real time was stated as a random Markov process.

The reliability of discontinuous-continuous system operation in real time represents a basic for defining their output and structure. System reliability parameters are obtained by analysis of its possible states related not only to equipment operating and failure times, but also conditioned failures due to working environment properties in real space, manpower and other scheduled and unscheduled delays. Discontinuous-continuous mining systems functioning is defined as random process with exponentially distributes operating and renewal times and material flow.

A Random Process of Mining System Elements Functioning

Operation in increasingly complex geological conditions and the need for costs reduction resulted in significant development of discontinuous-continuous systems in opencast mining. Mineral material and overburden excavation and local transport on openpit mines are carried out by discontinuous subsystems consisting of power shovels servicing dumptruck fleet. The dumptrucks are discharged into mobile and semi-mobile crushers which represent the first link of a continuous subsystem while the further masses flow is completed by conveyors to disposal areas or storages. Failure of a discontinuous subsystem element does not cause a failure of the complete mining system (except if the subsystem has only a single element), but leads to an output decrease. Failure of anyone of a continuous subsystem element leads to complete system failure. Determination of all system elements reliability parameters and operation simulation allow an analysis system functioning and in turn afford efficient production planning and design of a new systems in opencast mining.

Functioning of each element of a opencast mining systems presented as a Markov random process where the operating time flow with mean time (T_s) has the below exponential distribution:

$$f(t) = \lambda \exp(-\lambda t), \quad \text{for } t>0, \quad \text{where is } \lambda=(T_s)^{-1} \tag{1}$$

Since the system over time (d_t) passes from state (s_i) into state (s_j) it is possible to make out (n) homogenous equations of the possibility transfer probability with (n) unknown forms:

$$dp_i/dt = \Sigma\lambda_{ji}p_j - p_i\Sigma\lambda_{ij} \tag{2}$$

The following equations serve for calculation of operating state probability (P_0) and system renewal state (P_1) with operating and renewal intensities λ and β:

$$dp_o/dt = -\lambda P_0 + \beta P_1, \qquad dp_1/dt = \lambda P_0 - \beta P_1, \qquad P_0 + P_1 = 1 \tag{3}$$

Since $dp_o/dt = (\lambda + \beta) P_0 + \beta)$ below equations are obtained for P_0 and P_1 as a function of (t)

$$\begin{aligned} P_0(t) &= \beta(\lambda+\beta)^{-1} + \lambda(\lambda+\beta)^{-1} \exp(-(\lambda+\beta)t), \\ P_1(t) &= \beta(\lambda+\beta)^{-1} - \lambda(\lambda+\beta)^{-1} \exp(-(\lambda+\beta)t) = 1 - P_0(t) \end{aligned} \tag{4}$$

When $t \rightarrow \infty$, marginal stationary probabilities are as follows :

$$P_0 = \beta(\lambda+\beta)^{-1} \quad \text{and} \quad P_1 = \lambda(\lambda+\beta)^{-1} \tag{5}$$

Marginal probabilities for (n) system states may be generally obtained by below formulas:

$$P_1 = \lambda_0 P_0/\beta_1, P_2 = \lambda_0\lambda_1 P_0/\beta_1\beta_2, \ldots, P_n = \lambda_0\lambda_1 \ldots \lambda_{n-1}/\beta_1\beta_2 \ldots \beta_n \ \text{i}$$
$$P_0 = (1 + \lambda_0/\beta_1 + \lambda_0\lambda_1/\beta_1\beta_2 + \ldots + \lambda_0\lambda_1 \ldots \lambda_{n-1}/\beta_1\beta_2 \ldots \beta_n)^{-1} \tag{6}$$

The probability of operation of a system with combined connected subsystems and elements is as follows:

$$P_s = (1 - \Pi(1-P_{0j}))\, \Pi P_{0i}\text{, where:} \tag{7}$$

j - number of elements of subsystems connected in parallel (j = 1,...,m)
i - number of elements or subsystems connected in series(i = 1,...,r)

Discontinuous Subsystem

A discontinuous subsystem for opencast mining of mineral material deposits includes a series of independent elements which define its properties. System elements include deposit geological properties, opencast mine geometry, its facilities and mechanization, as well as the flow-sheet and work organization. From the aspect the theory of waiting queuing a discontinuous subsystem as a random process consists of (C) loading units and (K) dumptrucks with exponentially distributed loading times and travel cycles with parameters (b) and (a = const) in a stationary Poisson's process.

Intensity averaging is carried out in regard with varying dumptruck fleet travel cycle times (i = 1,...,C) for loading units according to :

$$a_s = (k_1 a_1 + \ldots + k_i a_i)(k_1 + \ldots + k_i)^{-1} \tag{8}$$

For operation of one loading unit or one with a limited number of dumptrucks (k) as a real technological whole the probability of (n) dumptrucks being in the system is:

$$P_n = P_0 k!(a/b)^n ((k - n)!)^{-1},\ (n = 1, \ldots, k)\text{ , where} \tag{9}$$
$$P_0 = (\Sigma k!\ (a/b)^n ((k - n)!)^{-1})^{-1},\ (n = 0, 1, \ldots, k)$$

The expected number of dumptrucks waiting to be serviced:

$$r = \Sigma\,(n - 1)\, P_n = k - (a - b)(1 - P_0)\, a^{-1} \tag{10}$$

The expected number of dumptrucks in the system:

$$z = n\, P_n = k - a(1 - P_0)b^{-1} \tag{11}$$

The expected dumptruck waiting time:

$$t_w = r(a\,(k - z))^{-1} \tag{12}$$

while the expected time of dumptruck residence in the system is as follows:

$$t_s = z\,(a\,(k - z))^{-1} \tag{13}$$

The structure of a discontinuous mining process as a part of an opencast mining system may be stated as a loading or discharge subsystems within which the dumptrucks are serviced. Both subsystems are cyclic and closed due to a constant number of constant number of machines and equipment, while change of the number of any property changes the structure of mining system output calculation. In each subsystem parallel servicing may be made, and the first dumptruck arriving in the subsystem is serviced.

The output of a discontinuous subsystem is obtained as a product of stationary probabilities of all elements operation and dumptruck output obtained on the basis of body volume, filling ratio, broken-to-solid ratio and cycle time, taking into account waiting for loading and discharge.

Continuous Subsystem

Operation of a serial continuous subsystem as well as operation of all system elements may be stated as an area of elementary states $S(A_{ij})$ where the sums of elementary states A_{ij} are for $i = 0$ operating states and $i>0$ for renewal states. In the case when $1 \leq i \leq n$ and $j = 1$, states sums are obtained in which the system or element are waiting for renewal. For $j = 2$ renewal is completed, while for $j = 3$ the system is waiting for start-up after renewal. An available system is non-operative due to renewal of auxiliary technological operations in the sum of elementary states $A_{n+1,1}$ and is also non-operative due to the effect of transported material in states sum $A_{n+2,1}$. The sum of elementary states $A_{n+3,1}$ represents the manpower influence, while sums $A_{n+4,1}$ represent the system states related to other specific or climatic delays. System state $A_{n+5,1}$ includes disposal prestorage delays. System state $A_{n+6,1}$ and $A_{n+7,1}$ inelude scheduled delays defined into times of repair, reconstruction and other scheduled delays.

Upon system failure the failured element is renewed. At a random time moment the system fails to be renewed during time interval T_{o1} and restarted at moment T_{o1}. Operating times $T_1, T_2,..., T_i$ as well as renewal times $T_{o1}, T_{o2},...,T_{oi}$ have appropriate distribution functions $F_i(t)$ and $F_{oi}(t)$. The moments of failure initiation are $t_l = T_i$; $t_2 = T_1 + T_{ol} + T_2$; $t_i = T_1 + T_{o1} + T_2 +... + T_i$, while the moments of renewal end are $t_{o0} = 0$; $t_{ol} = T_l + T_{ol} +... + T_{oi}$.

Determination of system reliability parameters in real time requires only definition of system existence in an operating state in some time moment t if the system was available in time moment t=0. System operating time t1 is defined by the shortest element operating time $t_r = \min (T_1, T_2,..., T_n$ and hence: $P (t_r \geq t) = \exp (-\lambda_1 + \lambda_2 + ... + \lambda_n) t)$. A continuous mining subsystem output is defined by masses flow of a discontinuous subsystem, but it also affects the whole system output by its operating probability.

Output Calculation

System output as a basic efficiency measure depends on the output of excavators excavating materials with varying properties in real openpit mine state. On the other hand, it is also conditioned by the reliability of all system elements operation defining system states and their probability to be realized in real time used to define the rate of time utilization rate. The product of variable values of output and operating probability of serially and parallel-serially organized subsystems and elements yields the real exploitation output of a complex system.

System output was defined on the basis of the calculation of operating reliability of a fully renewed system and its elements as a random process.

The product of obtained effective operating time of discontinuous equipment together with operating cycles parameters obtained on the basis of the theory of waiting queuing yields the exponentially possible output of an openpit mine discontinuous subsystem. With the aid of continuous subsystem operating probability it is possible to obtain the system true output and allows a comparative analysis of operating results in the case that system changes are introduced.

The results of comparative simulation analysis without participation of the times of scheduled technological and organizational delays, are given in the tables 1 and 2 on the basis of marginal values of failure and renewal time intensities, loading and servicing. The system consists of one power shovel, three dumptrucks, a semi-mobile crusher, conveyor belt and a spreader. System failure occurs with the failure of anyone continuous subsystem element or power shovel which is at the same time the most unreliable element. During the second period of operation upon the first power shovel failure a dumptruck also fails this leads during 1.1 hours to system output decrease (Table 1). During the second power shovel renewal listing 4.42 hours the dumptruck is renewed and hence during the next period of 8.44 hours to a failure the system operates with a full output. The continuous subsystem is the most reliable one whose failure is expected after 25.74

hours of effective operation. The results of the analysis of waiting queuing of a discontinuous subsystem are given in Table 1 for characteristic cases of power shovel operation with two (1.1 hours) and three dumptrucks (5.43 hours), The designed total system output of 20500 m^3 of overburden material may be realized during the analyzed time (26 hours). The overall system operating time is 18.82 hours, the failure time is 7.18 hours and the rate of time utilization is 0.72.

Table 1

Waiting queues of a limited number of dumptrucks	(1)	(2)
Intensity of dumptruck arrivals	10.00	07.00
Number of dumptrucks	03	02
Number of power shovels	01	01
Loading intensity	20	20
Working time (h)	05.43	01.10
Dumptruck body volume (m3)	50	50
Probabilities of (K) dumptrucks appearance in the subsystem:		
K = 0	0.210	0.512
K = 1	0.316	0.362
K = 2	0.316	0.126
K = 3	0.158	0
Expected number of dumptrucks in the queue	0.632	0.126
Expected number of dumptrucks in the subsystem	1.421	0.612
Expected waiting time in the queue	0.040	0.013
Expected residence time in the system	0.090	0.063
Subsystem ability (dumptrucks/h)	22.105	16.915
Subsystem output (m$^{3/}$t)	6005	930

Table 2

Simulation analysis of system functioning	Power shovels	Dumptrucks	Conveyors	Output
Failure intensity (min)	0.05	0.01	0.02	
Failure intensity (max	0.50	0.30	0.20	
Renewal intensity (min)	0.06	0.07	0.10	
Renewal intensity (max)	1.00	0.35	0.25	
Operating time (T_1)	**4.94**	9.27	25.74	5459
Renewal time (T_{o1})	**1.49**	3.97	5.46	
T_2	5.43	**4.33**	20.80	4785
T_{o2}	4.42	3.87	5.46	
T_3	**1.1**	1.1	16.47	930
T_{o3}	**4.42**	2.77	5.46	
T_4	**8.44**	27.74	15.37	9326
T_{o4}	**1.27**	9.01	5.46	

Conclusion

The developed software package does not enable only calculation of outputs of complex discontinuous-continuous opencast mining systems and simulation of process functioning with the possibility of changing system parameters. It also enables optimization of the schedule of machine groups in the openpit mine as a function of grade and excavation dynamics. The realized practical results, particularly in Cementworks Novi Popovac openpit quarries, indicate great possibilities for appropriate analysis of discontinuous-continuous systems operation.

References

1. Pavlovic V.: Reliability of discontinuous system in opencast mining, Faculty of Mining and Geology - Belgrade 1989.
2. Pavlovic V.: Continuous mining reliability, Ellis Horwood, Chichester, 1989.
3. Pavlovic V.: Model of Openpit mines operating Reliability, Faculty of Mining and Geology, Belgrade, 1994.

DISCRETE PROBABILISTIC MODELLING OF MINERAL PROCESSING

Anna Walaszek-Babiszewska
Silesian Technical University
Akademicka2, 44-100 Gliwice

Abstract. The paper presents some aspects of modelling of mineral processing. A discrete probability of occurrence of grains in raw materials and in the products of separation are considerated in the paper. The probability of occurrence a single grain in the general population is a basic parameter for calculating of sampling accuracy.

Introduction

It is known that materials as coal, ores, wastes of mineral preparation, some municipal wastes are heterogeneous in character. There are two basic characteristics which are commonly used by process engineers. These are:
- a characteristic of the grain composition,
- a densimetric characteristic.

In preparation plants there are the sequence of operations in which the separation of grains takes place according to their diameter or density.
The technology and control many of preparation processes are based on the information relating to material characteristics determined by a sampling method.

A discrete probability distributions of grains in the material

Physical features of a single grain such as a diameter and a density considered from the point of view of observation of the whole population are random quantities. In case of experimental investigations a certain finite number I of separable grain classes i.e. of separable intervals of grain diameter

$$(d_{i,min}\ ,\ d_{i,max}),\quad i=1,2,...,I$$

and a certain finite number J of separable density fractions

$$(\delta_{j,min}\ ,\ \delta_{j,max}\ ),\quad j=1,2,...,J$$

are assumed. For two features of grains (d,δ) we can defined a discrete probability distribution of a two-dimensional random variable:

$$p_{ij} = P\ [\ d\in(d_{i,min}\ ,\ d_{i,max}),\ \delta\in(\delta_{j,min}\ ,\ \delta_{j,max}\ )\ ]\ ,$$
$$i=1,2,...,\ I;\quad j=1,2,...,J$$

as a probability of occurrence a grain of the diameter d_i and the density δ_j in the examined population:

$$p_{ij} = \frac{N_{ij}}{N},\quad i=1,2,...,I;\ j=1,2,...,J; \tag{1}$$

where: N_{ij} - number of grains of features (d_i, δ_j) considered in the material, N -a total number of grains in the material.
In a distribution of the two-dimensional random variable there are marginal distributions:
-$p_{i.}$ - probability of occurrence a grain of the diameter d_i in the general population

$$p_{i.} = \frac{\sum_{j=1}^{J} N_{ij}}{N}\quad ,i=1,2,...,I \tag{2}$$

-$p_{.j}$ - probability of occurrence a grain of the density δ_j in the general population

$$p_{.j} = \frac{\sum_{i=1}^{I} N_{ij}}{N}\quad ,\ j=1,2,...,J \tag{3}$$

and conditional probability distributions, for example:

$$p_{j/i} = \frac{N_{ij}}{\sum_{j=1}^{J} N_{ij}} = \frac{p_{ij}}{p_{i.}}\quad ,\qquad j=1,2,...,J;\ i=1,2,...,I; \tag{4}$$

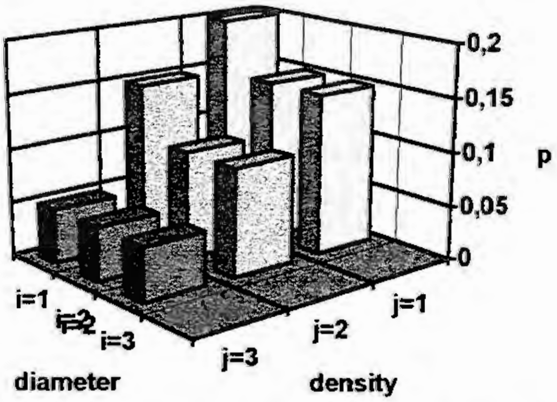


Fig.1. Empirical probability distribution p_{ij}, i- a diameter fraction number, j- a density fraction number.

We can calculate a discrete probability distribution of grains in a mixture of two grain materials, when the characteristics of the both materials are known:

$$p_{ij}^{(s)} = p_{ij}^{(1)} Q + p_{ij}^{(2)} (1-Q) \tag{5}$$

$$i=1,2,...,I;\ j=1,2,...,J;$$

where: $p_{ij}^{(1)}$ - probability of occurrence a grain of the diameter d_i and the density δ_j in the first material,
$p_{ij}^{(2)}$ - probability of occurrence a grain of the diameter d_i and the density δ_j in the second material,
$p_{ij}^{(s)}$ - probability of occurrence a grain of the diameter d_i and the density δ_j in the mixture,
Q - probability of sampling a grain belonging to the first material from the mixture

$$Q = \frac{N^{(1)}}{N^{(1)} + N^{(2)}} \tag{6}$$

These above named material properties can change in different points of space (for example in a stock pile) or in different moments of time, when the mass of the grained material is transported by a conveyer. Then probability distributions (1) - (4) depend on time t and coordinates vector s.

The discrete probability distributions of grains in products of separation

In a processing plant the raw grain material is subject to separation in a separator or a classifier. An ideal separation of grains according to the diameter d_{is} means e.g. that the grains of a set:

$$\{(d_i,\delta_j);\ i=1,2,..,i_s,\ j=1,2,...,J\}$$

pass into the screen undersize. The probability, that any grain from the midst of all grains of the feed will occur in the product of classifying amounts to

$$P(d_i \le d_{i_s}) = \sum_{i=1}^{i_s} \sum_{j=1}^{J} p_{ij} \tag{7}$$

An ideal separation of grains according to the density means that grains which the values of a density are lower than the separation density δ_{js} pass to the light product:

$$P(\delta_j \le \delta_{j_s}) = \sum_{i=1}^{I} \sum_{j=1}^{j_s} p_{ij} \tag{8}$$

If the probability of passing of a grain into the product is less than a unit and is equal R_{ij}, the total probability that any grain of the feed will occur in a given product can be presented in the formula [2]:

$$p^P = \sum_{i=1}^{I} \sum_{j=1}^{J} R_{ij} p_{ij} \tag{9}$$

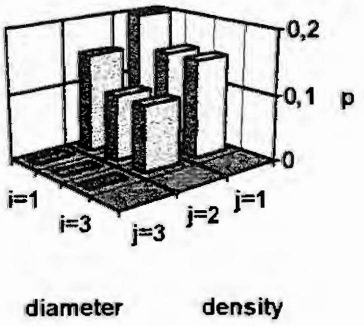


Fig.2. An ideal separation of grains according to the diameter d_2 from the material presented in Fig.1.

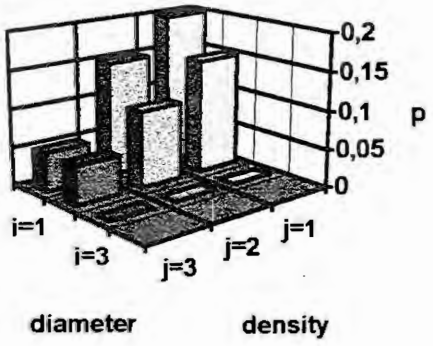


Fig.3. An ideal separation of grains according to the density δ_2 from the material presented in Fig.1.

The probability distribution of occurence of grains with the features (d_i, δ_j) in the product is determined by the relationship

$$p^P_{ij} = \frac{R_{ij}p_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} R_{ij}p_{ij}} \quad ,i=1,2,\dots,I;\ j=1,2,\dots,J. \tag{10}$$

therein

$$\sum_{i=1}^{I}\sum_{j=1}^{J} p^P_{ij} = 1.$$

The probability R_{ij} is the static characteristic of the preparation process.

The Markov processes have been used in many previous papers deal with descriptions of mineral preparation processes The dynamics of that preparation processes may be described by changing in time probability functions of passing the grains to the product. For a certain dynamics type of a batch preparation process we can describe that probability by a formula:

$$p_{ij}(t) = p_{ij}(0)[1-\exp(-\frac{t}{T_{ij}})]; \quad T_{ij}=T;\ p_{ij}(0)=0;\ i=1,2,3;\ j=1;\ t=0,\ T,\ 2T,\ 3T; \tag{11}$$

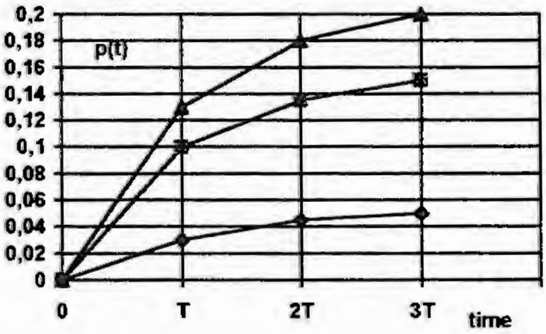


Fig.4. The dynamic characteristic of a certain batch preparation process

Statistical properties of the sample characteristics

The probability distributions of raw material are in general not known. This is the important problem to determine estimators of discrete probability distributions on the basis of sample characteristics.
During sampling tests numbers of grains in particular grain classes constitute a multivariate random variable $(X_1, X_2, ..., X_I)$ which can take the values

$$(X_1 = n_1, X_2 = n_2, ..., X_I = n_I),$$

where:

n- number of grains in a sample, $n_1 + n_2 + ... + n_I = n$.

The probability distribution with independent sampling being maintained is determined by a formula of multinomial distribution:

$$P(X_1, X_2, ..., X_I) = \frac{n! p_1^{n_1} p_2^{n_2} ... p_I^{n_I}}{n_1! n_2! ... n_I!} \tag{12}$$

Numbers $p_1, p_2, ... p_I$ are marginal probabilities (2) of occurrence a grain of the diameter d_i in the general population.
Frequencies of occurence of grains belonging to stated classes in a sample coposed of n elements which is under testing also constitute a multivariate random variable assuming the values

$$(Y_1 = \frac{n_1}{n}, Y_2 = \frac{n_2}{n}, ..., Y_I = \frac{n_I}{n})$$

with multinomial distribution (12).
Thus, in case of multiple drawing of samples, the most probable characteristic of material is determined to the probability

$$P(Y_1 = m_1 / n, Y_2 = m_2 / n, ..., Y_I = m_I / n) = \frac{n! p_1^{m_1} ..... p_I^{m_I}}{(m_1)! ... (m_I)!} \tag{13}$$

where:
m_i - the most probable number of grains in i-th fraction.

Each of random variables X_i or Y_i at independent sampling with constant number n of grain in the sample is subject to binomial distribution:

$$P[X_i = k_i] = P[Y_i = k_i / n] = \binom{n}{k_i} p_i^{k_i} (1 - p_i)^{n-k_i} \tag{14}$$

and expected values and variances are:

$$E(X_i) = np_i, \qquad D^2(X_i) = np_i(1 - p_i), \quad Cov(X_i, X_{i'}) = np_i p_{i'} , \tag{15}$$

$$E(Y_i) = p_i, \qquad D^2(Y_i) = p_i(1 - p_i)/n , \quad Cov(Y_i, Y_{i'}) = p_i p_{i'} / n \tag{16}$$

i, i'=1,2,...,I.

Identification of grain material characteristic by sampling methods

Repeated measurements of granular characteristic of fine-grained materials: magnesia and quartz were carried out by means of a laser analyzing diffractometer. The instrument makes it possible to determine the content of 31 grain classes within the range 0.9 to 200μm. The content p of each class is given as percentage. On the basis of series of measurements of characteristic in samples mean values, variances, coefficient of variation (relative error,%) and histograms of the content of particular grain classes were calculated. Fragments of results are shown in Table.1 and in Fig.4.

where summation extends over all combinations of numbers $n_1,n_2,...,n_j$ with constant size of the sample n.

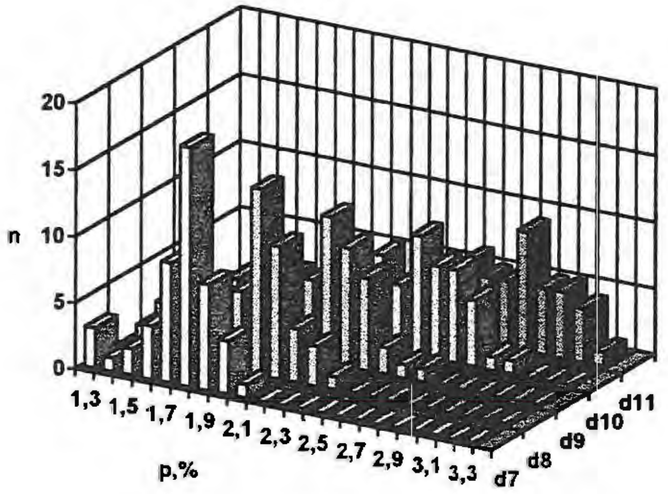
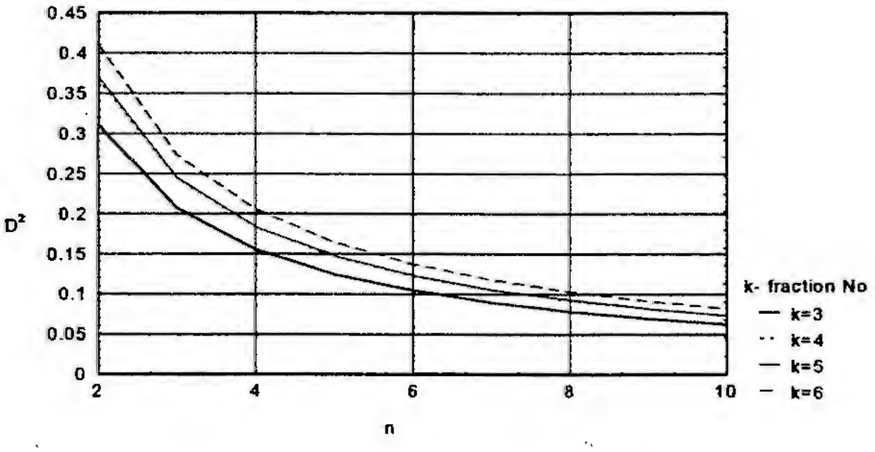


Fig.4. Histograms of the contents of grain classes No.7-11 (Table.1)

Fig.5. Measure of accuracy in simulation tests



References

[1]. Walaszek-Babiszewska A.: Modele stochastyczne opróbowania węgla. Zeszyty Naukowe Politechniki Śląskiej, seria Górnictwo, z.203.Gliwice 1992r.

[2]. Walaszek-Babiszewska A.:Stochastic models of a characteristic of the grain composition and a densimetric characteristic in samples. Archives of Mining Sciences.1993 Vol.38, Is.2.

MODELLING AND SIMULATION OF THE NEW EUROPEAN TRAIN CONTROL SYSTEM

A. Janhsen[1], K. Lemmer[1], B. Ptok[2], E. Schnieder[1]
[1] Institut für Regelungs- und Automatisierungstechnik,
Technische Universität Braunschweig,
Langer Kamp 8, D-38106 Braunschweig,
K.Lemmer@tu-bs.de

[2] Deutsche Bahn AG
Zentralbereich Neue Systeme, ZNS 12
Arnulfstraße 9-11, D-80335 München,

Abstract. This paper deals with a description of the modelling and simulation of the new European Rail Traffic Management System (ERTMS). The general structure of the project is illustrated and the modelling approach is explained.

Different system views were integrated in the entire model which was designed and evaluated by Petri Nets. Based on an informal (natural language) specification a formal net representation was created. The investigation of static as well as dynamic system behaviour is possible. The model consists of a process, scenario and functional account. The aspects of reusability of subnets are mentioned.

1 Introduction

The aim to realise an European Rail Traffic Management System (ERTMS) is motivated by the problems of the frontier crossing of trains. Up to now each railway administration forces its own train control system. In other words it is necessary to change the train traction for passing the border or to equip the engine with many different systems on board as national solutions of the countries it will pass. In consequence the first solution costs a lot of time the second a lot of money. In order to avoid this it is important to define an unique train control system. It means in detail that the train can pass the frontier without changing the engine. Furthermore it is not necessary to change the engine-driver, because an uniform signalisation is provided for the man-machine-interface (MMI). This case is called interoperability.

The development of an interoperable future train control system is characterised by high complexity. Different kinds of interests related to the implementation strategy and to the existing functional environment of the system has to be reflected on. Furthermore it is necessary to look at the safety critical aspects for the train control system. The interests in modelling and simulation of the new system are focused on the verification of the standardised interface description and the investigation on the integration of the new system to the existing interlockings and regulation systems.

The implementation of the new train control standard will be the basis for the unrestricted border crossing services as well as for the liberal using of the railway network by several railway companies. Appropriate European directives are in preparation.

2 The system structure

The main units of the ERTMS are the onboard and trackside systems. The onboard system supervises all functions realised on the train and communicates with the trackside system. So, for the communication purposes, the onboard components cover the radio installation and devices for reading of beacons, so-called balises. Beside this there exist special interfaces to the existing national train control systems, called Specific Transmission Module (STM). Also the movement of ERTMS fitted out trains on conventional tracks is possible by the STM's.

The heart of the trackside system of ERTMS is the Radio Block Center (RBC). The RBC communicates with the train, the trackside installations (for example level crossing), the interlocking and the regulation. Evaluating the information given from the interlocking and the regulation the RBC gives movement authorities, orders and instructions to the train.

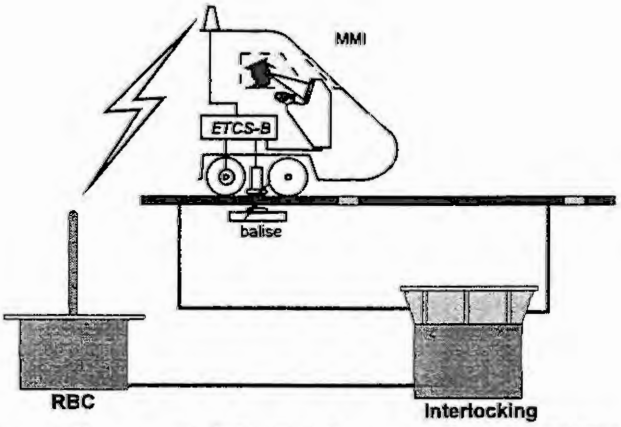


Figure 1: European Rail Traffic Management System (ERTMS)

The ERTMS identifies three functional application levels. They are defined to provide the railway companies with systems which are adequate for their typical national environment. They also ensure downward compatibility. Using the different levels, fall back strategies and migration processes are supported. The main characteristics of these three levels are listed beneath:

	Level 1	Level 2	Level 3
Technical Interoperability	**Yes**	**Yes**	**Yes**
Operational Interoperability	**Only with MMI signalling**	**Yes**	**Yes**
Physical signal trackside	**Yes**	**Optional**	**No**
Driver information	**Signalling trackside and optional cab signalling / MMI**	**Cab signalling / MMI and optional signalling trackside**	**Only cab signalling / MMI**
Radio Transmission	**No**	**Yes**	**Yes**
Train Integrity	**Checked trackside**	**Checked trackside**	**Checked by train itself**
High performance blocks (virtual short sections) possible	**No**	**Yes**	**Yes**
Moving Block possible	**No**	**No**	**Yes**

Table 1: Characteristics of the different ERTMS levels

Under certain circumstances the ERTMS could provide also mixed traffic of different application levels.

3 The Project: The formal system specification

Based on the informal (textual) System Requirement Specification (SRS) it is the goal to create a formal system model in order to verify the specification. The requirement of a safe system demand the proof of the correct system behaviour specification. The most important part of the model is the interaction between onboard and trackside. Furthermore the interface to the environment for the system stimulation is modelled.

In order to proof the system behaviour a formal description in a mathematical sense is important. Either the correctness, logic behaviour and concistency of the model have to be checked. Not only the static aspects moreover the system dynamic have to be considered. On account of this Petri nets were used for system modelling.

3.1 Specification in three phases

The base for the modelling was the system requirements specification at the state of the art. The specification was available in textual form. This informal representation is not easy to percept and has all adjectives of the natural language: incompleteness, incorrectness, ambiguousness and contradictions. Moreover the static structure and dynamic behaviour is not provable.

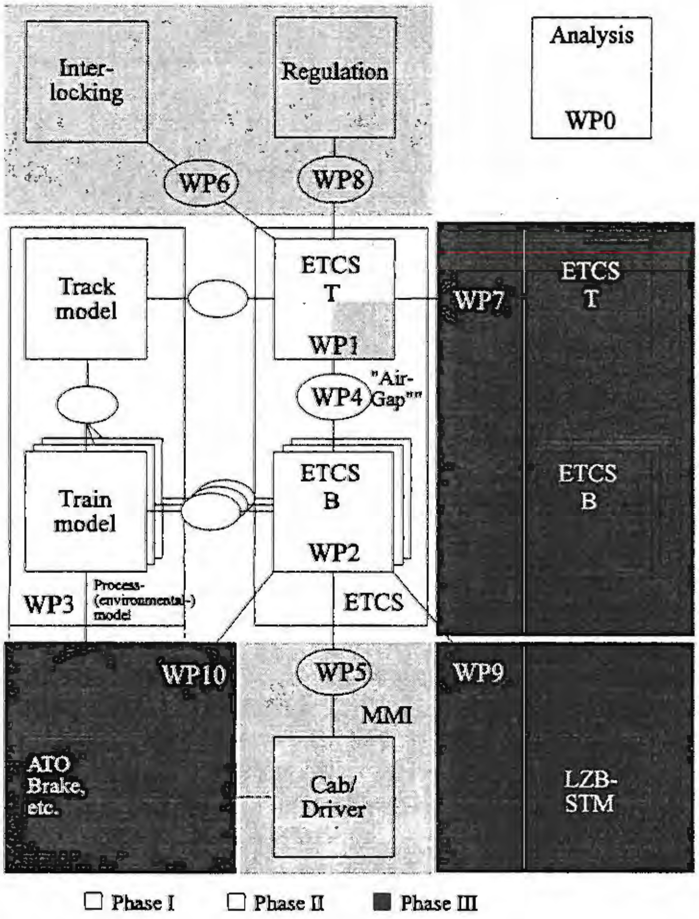


Figure 2: Project phases and workpackages

The aim was to model the system with Petri Nets. The benefit is a formal representation and the integration of static and dynamic aspects in an unique model. Moreover it is possible to simulate and to analyse the model. In order to handle the project efficient three phases were defined as shown in figure 2. A central part is the interaction of the on board and trackside equipment and the communication over the 'air gap'. For the environmental behaviour no explicit specification exists, but it was important to model that interface for the dynamic check.

3.2 Modelling of the ERTMS

On different levels the model represents different appearance of the system. The model incorporates the following paradigms:

- *Process*

  The representation of the (technical) process is oriented at the system behaviour. In the general view the interfaces, the objects (like e.g. regulation, interlocking, train or radio block center) and the global behaviour is included.

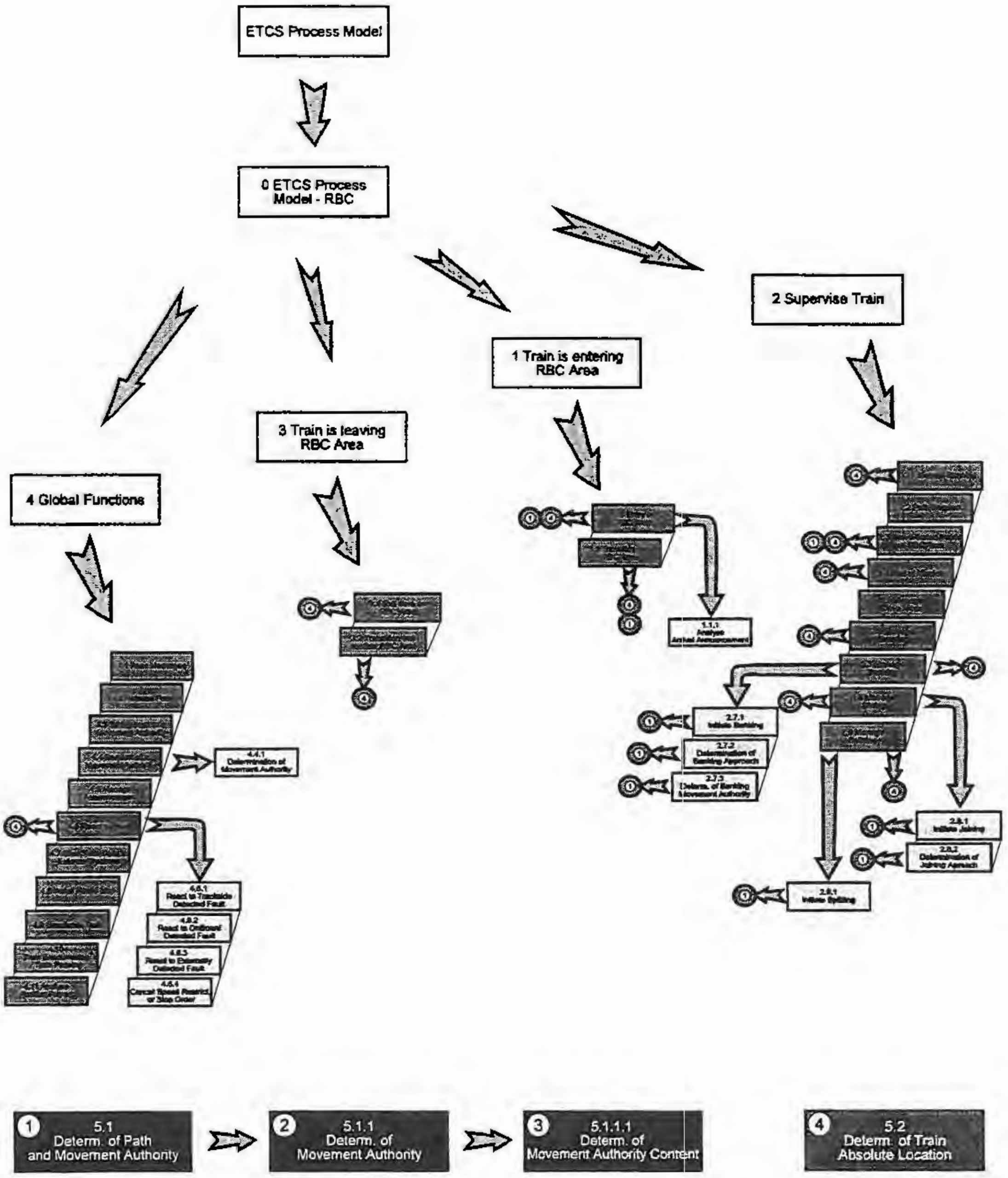
  For example the process view in phase I (compare fig. 2) represents the interaction between the Trackside/RBC (ETCS-T, WP1), On Board (ETCS-B, WP2) and environmental part (WP3) of the model.

- *Scenarios*

  The scenario view consists of walk through strongly connected activities. That means in detail the closely related activities of mainly operational type which are necessary for e.g. entering to RBC, RBC area change, joining, splitting, shunting, etc.

- *Functions*

  The functionality is shown in this more detailed level of the model. Some functional modules can be used in different scenarios. They are stored separately for reuseability. The are then called functional blocks.



ETCS Process Model
0 ETCS Process Model - RBC
2 Supervise Train
1 Train is entering RBC Area
3 Train is leaving RBC Area
4 Global Functions
1.1.1 Analyse Actual Announcement
2.7.1 Initiate Banking
2.7.2 Determination of Banking Approach
2.7.3 Determ. of Banking Movement Authority
2.8.1 Initiate Joining
2.8.2 Determination of Joining Aproach
2.9.1 Initiate Splitting
4.4.1 Determination of Movement Authority
4.6.1 React to Trackside Detected Fault
4.6.2 React to Onboard Detected Fault
4.6.3 React to Externally Detected Fault
4.6.4 Cancel Speed Restrict. or Stop Order
1
5.1 Determ. of Path and Movement Authority
2
5.1.1 Determ. of Movement Authority
3
5.1.1.1 Determ. of Movement Authority Content
4
5.2 Determ. of Train Absolute Location

Figure 3: Structure of the RBC Petri Net model

Figure 3 shows the structure of the RBC Petri Net model. Without discussing details the composition of the nets can

[Eng93] English, S. L.: Coloured Petri Nets for Object-oriented Modelling, Thesis, University of Brigthon, 1993

[ERT96a] Interoperability and ERTMS Level definition, Doc. ref. 96E0132-.doc, 15.02.1996

[ERT96b] ERTMS, Preliminary SRS, Tracside Operating Priciples, Ref. EEIG:96058, Doc. ref. 96E0581-.doc, ERTMS Users Group, Brüssel, 03.04.1996

[FRS95] Functional Requirements Specification, FRS Synopsys, Ver. 3.0, UIC/ETCS, European Rail Reserach Institute, Utrecht, Netherland, 30.11.95

[Har87] Harel, D., et al.: On the Formal Semantics of Statecharts (Extended Abstract), Computer Science, Proc. on the 2nd IEE Symposium on Logic, 1987

[HaPi88] Hatley, D.J.; Pirbhai, I.A.: Strategies for Real-Time System Specification, Dorset House Publishing Co., New York, 1988

[JaLeSi96] Janhsen, A.; Lemmer, K.; Schnieder, E.: *Entire Design Process: From Human Ideas to Technical Implementation*, In: IFAC-World Congress San Francisco, 1996

[JaSi96B] Janhsen, A.; Schnieder, E.: *Entire Design Process: Introduction of a Mental Reference Model for Systems Specifications*, 11th conference Applications of Artificial Intelligence in Engineering, Clearwater / Florida, 1995

[Jen92] Jensen, Kurt: Coloured Petri Nets - Basic Concepts, Analysis Methods and Pratical Use, Volume 1 of EATCS Monographs of Theoretical Computer Science, Springer, New York, 1992

[Rum91] Rumbaugh, J.; et al.: Object-oriented Modelling and Design, Prentice Hall, New Jersey, 1991

[Pto96] Ptok, B., Kollmannsberger, F., Wojanowski, E.: ETCS - Kern des Europäischen Eisenbahn Verkehrs Management Systems, Eisenbahningenieur(47), 1996

[PtKoWo96] ERTMS - European Railway Transport Management System, Eisenbahntechnische Rundschau, ETR 45 Heft 3, 1996

[Sch93] Schnieder, E.: Universal Approach to the Design of Automation Systems, VDI-Berichte Nr. 1067, pp. 363-376, VDI Verlag, Düsseldorf, 1993

[Sch94] Schnieder, E.: Modelling in Transition to the Information based Automation, Lecture, Colloquium of control systems, Boppard, 1994

[Sch95] Schnieder, E.: Basics, Perspectives and Visions of Design Methodology of Automation Systems, Conference 'Design of Complex Automation Systems', Braunschweig, 1995

[Shl88] Shlaer, S.; Mellor, S.: Object-Oriented Systems Analysis: Modelling the World in Data, Yourdon Press Inc., New Jersey, 1988

[SiJa94] Schnieder, E.; Janhsen, A.; et al.: Impact of Advanced system engineering to the design of modern railway operations control system, Comprail, Computers in Railway IV, Vol. II Railway Operation, Section 2: Signaling, Communications, Advance Train Controls, Madrid, 1994

[You89] Yourdon, E.: Modern Structured Analysis, Prentice-Hall Inc., 1989

PETRINET BASED MODEL OF CONTAINER TERMINAL LOGISTICS

K. Müller and E. Schnieder
Technical University of Braunschweig
Langer Kamp 8, D-38106 Braunschweig

Abstract. It is not reasonable from ecological and economical points of view to transport long distance goods by trucks. Furthermore, the advantage of the flexibility of trucks is decreased by the amount of time lost in traffic congestions on highways.

1. Motivation

In the last years the traffic situation on the European roads has become more and more difficult, especially because of the opening of the eastern frontiers. The traffic statistics of the past, which can be seen in figure 1, show that the share of railway has become even smaller. One possibility of improving the situation would be a shift of the long distance traffic from the road onto the railway.

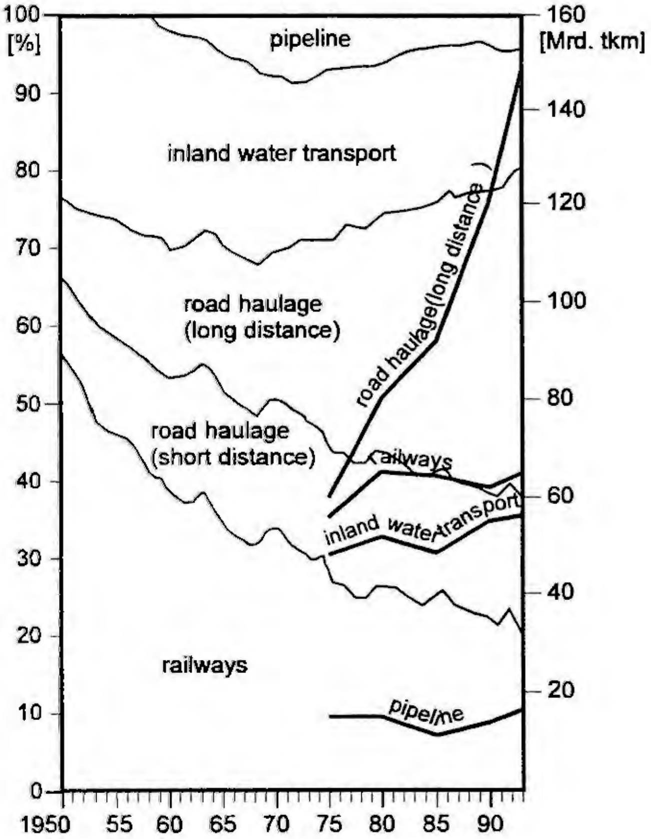


Figure 1.1: Distribution of freight traffic in germany

If the traffic situation will develop in the future as it did in the past, the flexibility of transport by trucks for long distance goods will come to nothing because of permanent traffic jams. Therefore, it is necessary to look for new traffic concepts and care for their quick realization. One possibility of improving the situation on the roads is to use the specific advantages of trains and trucks. These are the flexibility of trucks in the local traffic and the long distance transport capacity of the train with regard to ecology and economy. These advantages of both systems should be made use of in a combined transport system.

The combined goods traffic is characterized by trucks gathering the goods and loading them for long distance transport onto the train, and again by the trucks taking over the goods from the rail near the place of destination for distribution.

This process has its weak point in the fact that transport by rail is predominantly carried out by night, as the passenger traffic in Germany takes priority during the day.

The terminals operate at maximum capacity mainly in the early morning hours and in the early evening hours, because of the fact that the forwarders want to deliver their containers as late as possible in the evening, and like to pick up the goods as early as possible in the morning. Thus for many hours of the day the loading capacity of the terminals is scarcely profited by, which results in delays in the morning and evening.

As the international container traffic is continually growing, it is therefore absolutely necessary to develop more and - above all - faster operation modes of transloading goods in terminals, which will make traffic more attractive for the forwarding trade. The new systems will have to make possible not only shorter transfer times between train and truck, but should be integrated in the whole transportation chain.

The aim of this project is to model the entire transportation system with all possible transport chains to calculate in advance the best transportation chain and to generate all required transportation data. Hence a virtual transport process results. The whole model consists of several decomposition levels. In this paper it will only be possible to show the decomposition of the container handling in the terminal. The general structure of the whole model is shown in figure 1.2.

2. The Transportation Chain

In the last chapter we described that it is not only the aim to speed up the container transfer, but to optimize the entire transportation chain. In our opinion it is necessary to look at two parts. In the first it is necessary to generate possible transportation chains. Each of these chains had an effect on the current traffic situation if it were realized. In the second part it is necessary to look at the real transport.

Why is it necessary to look at the entire chain?
Each carrier has its own advantages and its own working capacity. In an optimal situation all carriers are working near 100% capacity. Another optimum is reached, if the carrier with the fewest ecological damage or the cheapest shipment transports the most cargo.

It is only possible to reach one of these peaks by considering all carriers and using those with the most suitable characteristics. For example: Assume that on the motorway to the destination there normally is a traffic jam. The time requirements for the transport are not very strict. If someone has all required information about the possibilities of the other carriers he is able to use a better alternative for example to transport the goods by train.

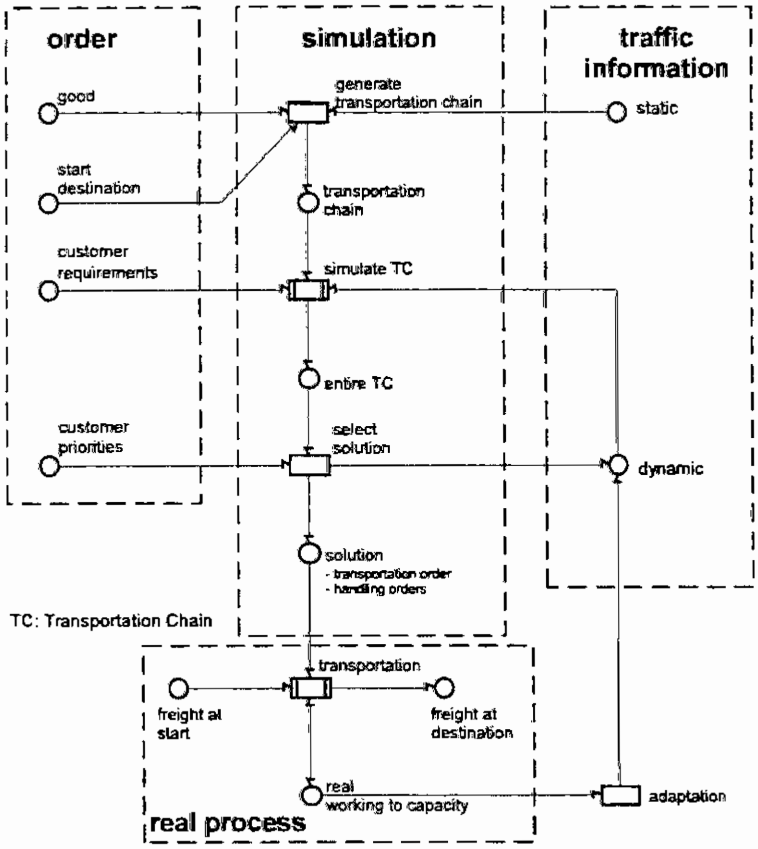


Figure 2.1: The entire transportation chain

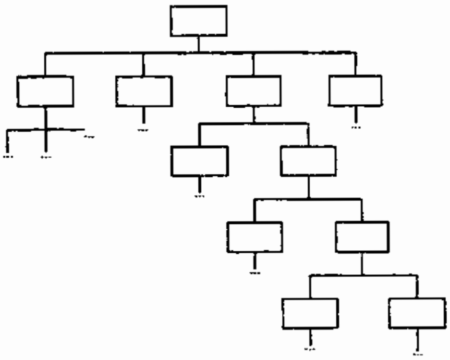


Figure 1.2: structure of the whole model

Description of the entire transportation chain
The transportation chain contains the following objects:

- the transportation order
- the wirtual traffic process
- the traffic information
- the real process

The entire transportation chain is shown in Figure 2.1.

Transportation order
The object 'transportation order' contains the information about the goods, for example weight, size, transport conditions and so on, as well as the starting point and the destination. Particulary important are the requirements of the customer. These could be for example a secure or a fast transport.

Traffic information
The traffic information has to be divided in two parts, static and dynamic information. The static information contains a list of all handling modules, a list of all roads, the traffic regulations and the reliability of the roads.
The dynamic information contains the present capacity of the handling modules, the current working capacity of the streets and the rails.

Virtual Transportiation Chain
The aim of the simulation is to generate an optimal transportation chain. To reach this, general steps are necessary. In the first step a transportation chain is built out of the static traffic information, the information from the goods and the starting point and destination.

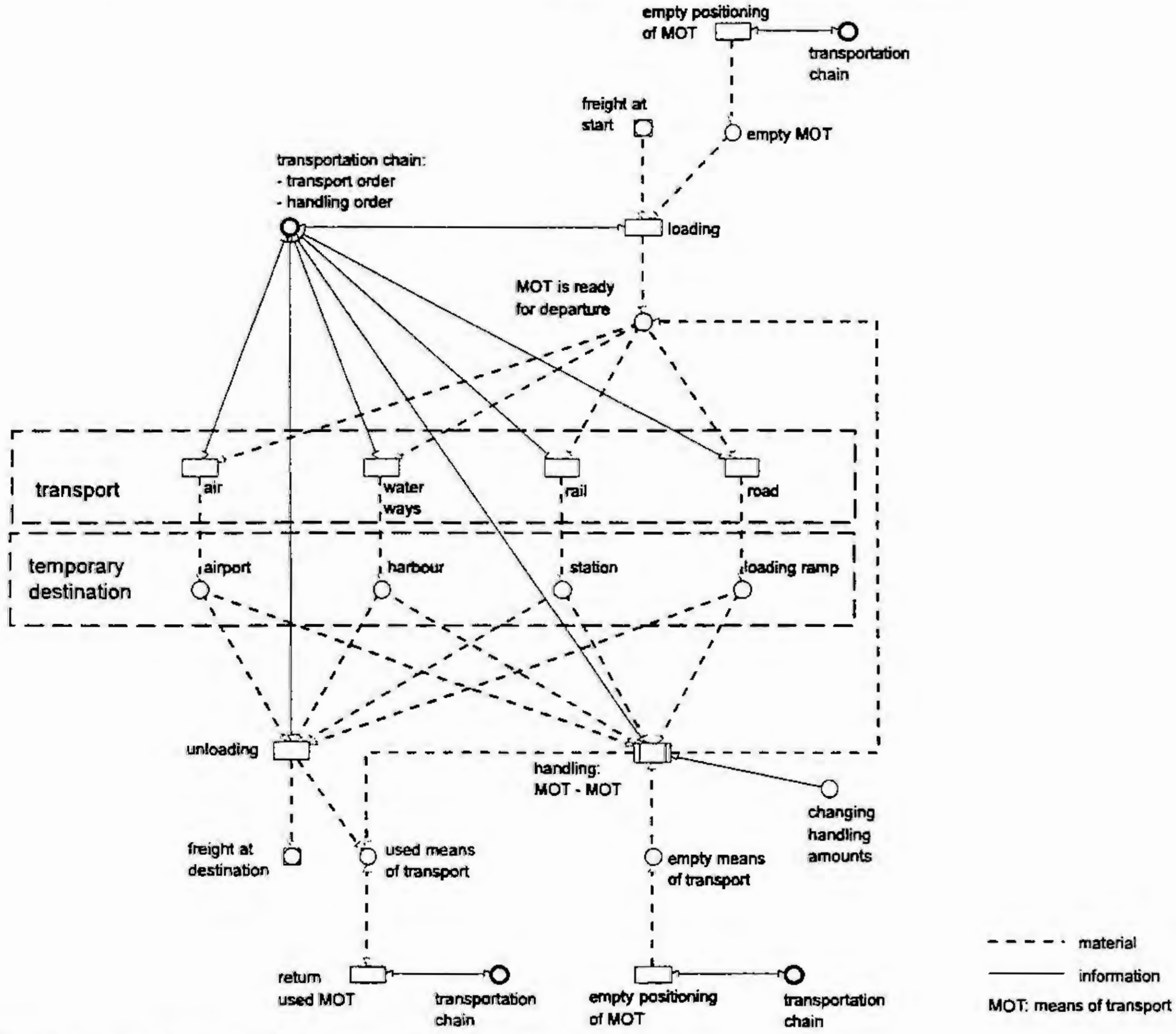


Figure 3.1: refinement of the transition transportation

This transportation chain contains no information about the actual traffic situation. Today this is the only chain that can be generated. In the future it will be possible to get information about the actual traffic situation with systems like RDS/TMS (Radio Data System / Traffic Message

Specification).

With the static transportation chain, the requirements from the customer, and the dynamic traffic information, the simulation calculates the chains which meet all requirements. At this point this information is necessary because the results affect the traffic situation.

The customer has to choose one of the calculated transportation chains. As soon as he chooses one solution this choice influences the real dynamic traffic information in the future. This solution is the chain which describes the use of all means of transportation from the start to the destination.

3. Refinement Transition 'transportation'

One of the advantages of petri-nets is the possibility to refine transitions. With this hierarchical models can be developed. The Figure 3.1 shows the refinement of the transition 'transportation'.

There are connections to the upper net with the places freight at start and at destination and the selected transportation chain. This net shows the entire process which begins with the provision of an empty means of transportation. In the combined transport this is usually a lorry. The loaded lorry drives to the loading ramp in the next terminal. There another empty means of transportation is needed in this example it is an empty rolling stock. Now the lorry is empty and can load for example another container for the return trip.

In general the freight could be transported via aircraft or ship or by rail or by street. The course of events is the same for each carrier. It it important that the token on the place 'transportation chain' contains all information and controls the entire transportation process.

If the last means of transportation reaches the destination, the freight will be unloaded and the means of transportation is ready to load another freight.

4. Refinement of the Transition 'handling'

The refinement of the transition 'handling' at the next level shows the kinds of ports of transshipment which connect the mode of transport with different means of transport. For example a harbour connects the modes of transport street, rail and waterway. The terminal 2000 is only used to connect street and rail.

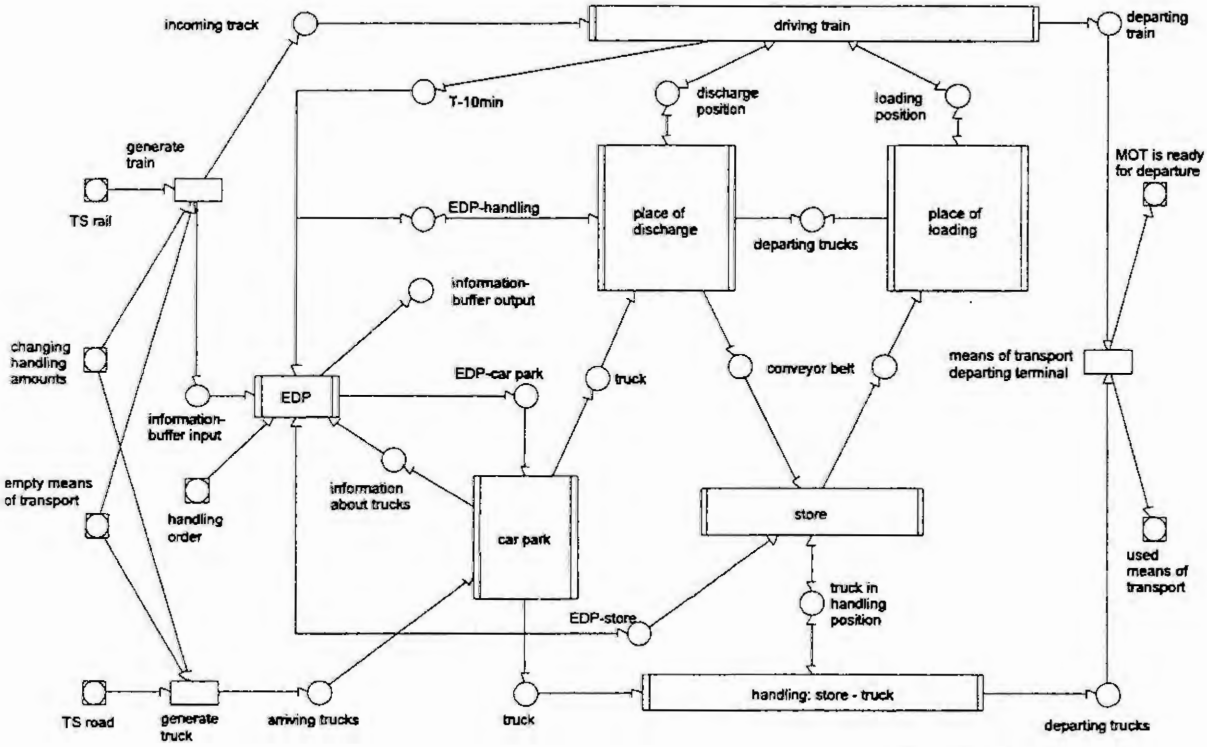
Concept of the terminal

When at the right moment a truck is ready at the transfer position to receive the container, the robot loads the container directly onto the truck. Should the truck be late, the transfer robot is programmed to drop the unit load immediately, so as to be ready for handling the next arriving unit load. For such a situation a conveyor belt has been chosen, on which such undispatchable containers can be placed in order to be transported for storage out of the transfer area. This process of unloading the containers from a slowly moving train can be reversed for loading.

The refinement of the handling module is the final detailed level in the transportation chain, see Figure 5.1. A refinement of one of these transitions would lead to the level of sensor and actuator models.

With the aim of improving the chances for the combined goods traffic a new concept is being developed at the Institute of Control and Automation Engineering at the Technical University of Braunschweig Germany. The realisation concept comprises the following aspects:

- automatic transfer of containers
- direct transport over the day
- optimized management based on simulation



TS: Traffic System
EDP: Electronic Data Processing
CB: Conveyor Belt
MOT: Means of Transportation

Figure 4.1: petri net model of the terminal

The terminal consists of several places and pieces of equipment. The main places are the place of discharge and the place of loading, car park, rail and stock. The main pieces of equipment are the handling module, the conveyor belt and the store. Figure 4.1 shows the function of the terminal with petri nets.

5. Automatic Unit Load Handling

Coresponding to the function of the transition 'handling' a reference modul has built in scale 1:45 in our institute. In a real terminal the handling module has the following job: While the train moves along, the containers are unloaded in direct transfer, which means without the disadvantage of any shunting, directly under the connecting line.

The transfer robot is composed of two swivel axles and two linear axles. By the main swivel axle an alternative movement between train, truck and drag chain conveyor is performed. The second swivel axle moves in opposite synchronisation to the main swivel axle, in order to keep the grab always parallel to the train. The first linear axle synchronizes itself with the train's speed, and the second linear axle lifts the containers from the train. The transfer robot' s job is to move its spreader between the connecting line and the train with the aim of carrying out a rendezvous with the container on the slowly moving carriage. As soon as the spreader has taken up the speed of the train and has found the right position above the container, it takes hold of it, lifts it and moves it sideways off the train.

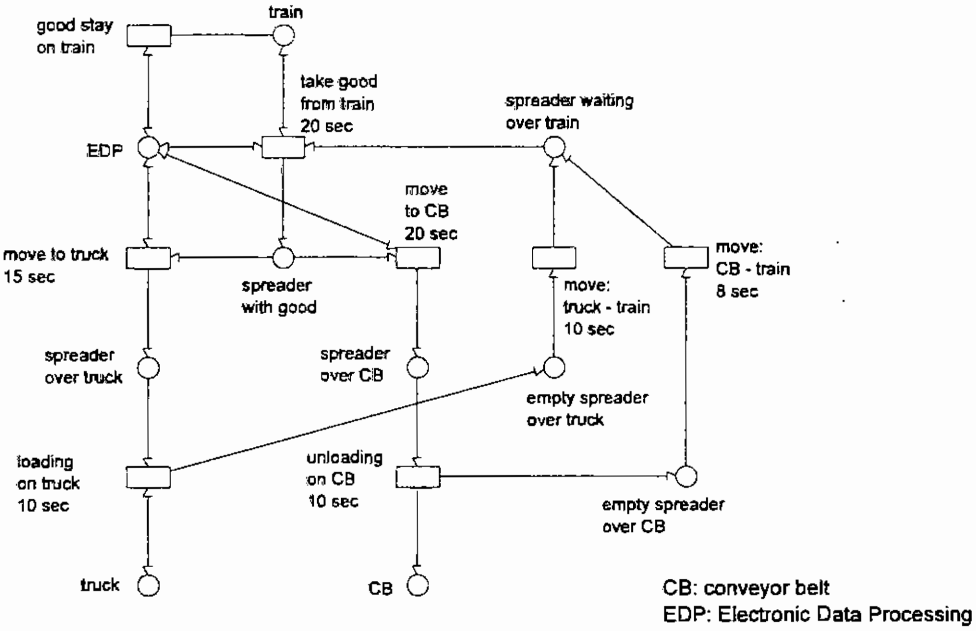


Figure 5.1: handling modul

6. The Tool INCOME

For the modelling of the described petri nets we use the tool INCOME which is basically a workflow management tool [Promatis] based on petri nets. INCOME uses predicate transition nets for modelling. The programming language for the description of the function of a transition is PROLOG. INCOME works together with an ORACLE database. For the described problem of combined goods traffic this is a big advantage in relation to some other tools, because there is a lot of information to be handled about the trains, the containers or the trucks.

7. Literature

Müller, K.; Wend, F.; Schnieder, E.
Schnellumschlagterminal und Logistikkonzept für einen leistungsfähigen Gütertransport auf der Schiene
In: Innovative Umschlagsysteme an der Schiene. VDI-Verlag, Düsseldorf. 1996, S. 79 - 101. (VDI-Berichte 1274)

Wend, F.; Schnieder, E.
Das Konzept "Terminal 2000" - Neues Umschlagkonzept für Container in der integrierten Transportkette
In: Der Nahverkehr. Nr. 5, 1994, S. 20 ff.

Wend, F.; Schnieder, E.
Terminal 2000 - A new transfer concept for the combined goods traffic survey
In: Large Scale Systems: Theory and Applications. Vol. 2, IFAC/IFORMS/IMACS Symposium, Pergamon Press, London. 1995, S. 881-886

PROMATIS Informatik
Descostraße 10 · D-76307 Karlsbad · Germany · Tel. 00497248 / 91 45 - 0 · Fax 91 45 - 19 · E-Mail HQ@PROMATIS.DE · http://www.promatis.de

SIMULATION AND VIRTUAL REALITY IN REQUIREMENTS ENGINEERING

A. Janhsen[1] **and M. Krone**[2]
[1] Institute of Control and Automation, Technical University of Braunschweig,
Langer Kamp 8, 38106 Braunschweig, Tel. +49-531-391-3317, janhsen@ifra.ing.tu-bs.de
[2] Siemens AG, Transportation Systems,
Postfach 3327, 38023 Braunschweig, Tel. +49-531-226-2977, Maren.Krone@bwg4.erl1.siemens.net

Abstract.. Developing large and complex railway systems is a difficult task due to the fact that lots of requirements and detailed aspects have to be considered. Thus it is important to emphasise on early phases and especially on the requirements engineering phase. To enhance the result of this stage, we will combine modelling methods with simulation and virtual reality representations of the later system. In this paper we introduce the structure of the design framework, relate simulation, visualisation and virtual reality representations to railway systems engineering and explain the idea of combining both.

Introduction

The development of large and complex railway systems is by far not a trivial task. Especially the requirements engineering phase is usually a big challenge for all participants. There are not only a huge amount of different requirements to take into consideration but also to fulfill the so-called RAMS requirements, that stands for reliability, availability, maintenance and safety of the later railway system. Additionally, problems between developers and customers like e.g. different semantics in vocabulary or ambiguity have bad influence in the requirements elicitation phase e.g. misunderstood requirements. That enormous complexity of this problem field leads to the use of formal methods [5,8]. However, due to the fact that customers normally do not understand any formal system description, they need an intuitive representation of it. Explanations about the syntax of the used formal languages to the clients does usually not lead to a better understanding, as it does not ensure an entire agreement on the semantics.

Visualised simulation (VS) and virtual reality (VR) is a more intuitive representation of complex aspects for human beings. Furthermore, there is no need for the customer to learn a new description language. Also customers can easily determine the completeness of their ideas. Thus, using different views of the system or different visualised scenarios reduces difficulties in understanding.

Such a combination of modelling methods based on formal methods with VS/VR in the requirements engineering phase needs a structural model together with methodological guidelines. In the following part we introduce the framework, relate virtual reality and visualised simulation to the scope of railway systems engineering, introduce the idea of combining simulation/virtual reality and modelling methods and give an outlook to further work.

Structure of the Design Framework

Common approaches for systems design often have a linear or quasi-linear sequence of phases, where each phase gives output to the next and receives requests for updates of the latter (e.g. [1, 10, 12]). However, large projects usually can not follow such a clear determination of steps and linear approach. Engineers and developers are forced by strong influences to organise work flows also including parallel phases and interactions between different steps.

The structural model proposed in this paper does not follow these quasi-linear approaches. It is rather a two-dimensional framework, where the kernel is a formal database containing all necessary information about requirements and systems specification (see figure 1). The framework is a code-based software architecture for solving specific problem classes with interaction of its components. Also the interfaces to related systems are included in the definition of the overall structure. As early as possible in the project, the requirements should be described in a formal language due to the resulting decrease of ambiguities and the increase of requirements quality. In the first step, starting with the requirements within the formal frame, the requirements specification is enhanced by additional formal descriptions of the domain context in order to receive an entire formal description of an scenario. The understanding of scenarios in this paper is different to the scenarios mentioned in literature, like e.g. [2, 13]. We introduce a new type of scenario as a formal description of subprocesses of the overall system, which has to be complete and consistent according to an introduced metamodel of specification languages, the so-called mental reference model [6]. If the description is still insufficient in relation to a reference model, it must be completed by performing guided discussions with the user or using generated check lists. Hereby the interface to the documentation and test components with the requirements part in figure 1 is evolved. This is the important point that is different to other approaches like mentioned above. Another advantage of using formal description languages in this early stage in the design process is that all other tasks of systems engineering like analysis, tests, simulation, structured documentation and implementation can be optimal performed with computer aid.

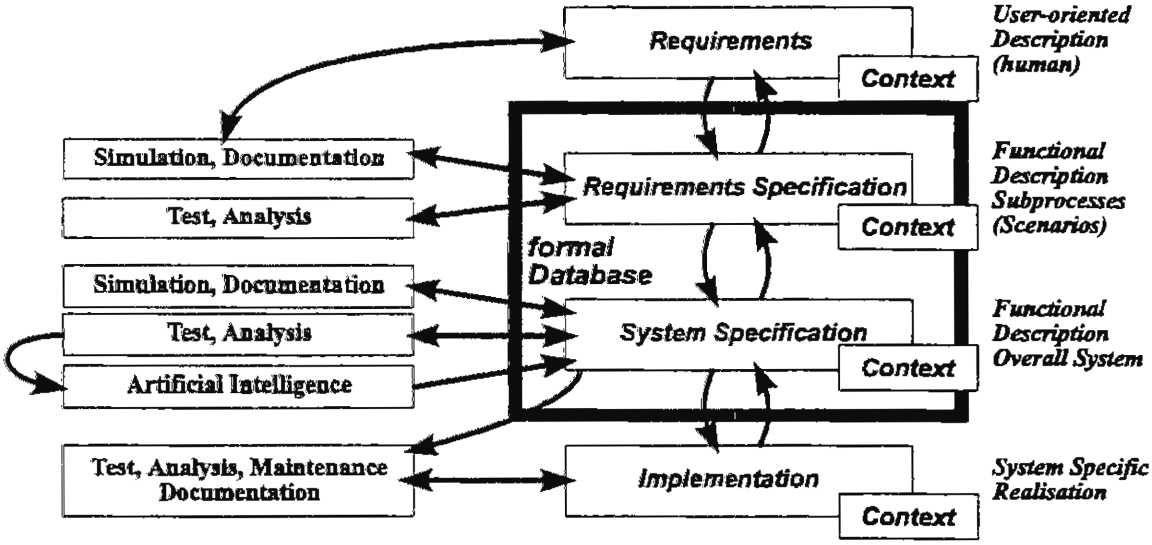

Requirements
Context
User-oriented Description (human)
Simulation, Documentation
Test, Analysis
Requirements Specification
Context
Functional Description Subprocesses (Scenarios)
formal Database
Simulation, Documentation
Test, Analysis
Artificial Intelligence
System Specification
Context
Functional Description Overall System
Test, Analysis, Maintenance Documentation
Implementation
Context
System Specific Realisation

Figure 1: Framework of the design process

In the next step, the formally described, consistent requirements build up the entire system specification with the integration of all predefined subprocesses. The entire system specification is the second component of the formal database. Both components are defining different description levels. They are extended by context information, which can be automatically integrated as domain specific knowledge, when also stored in the same data format. Of course also the user itself has context knowledge of the user specific application domain, which can not accessed by the systems analysts easily. The same increase of knowledge is carried out during the implementation phase by including context information. Only in the two other cases of the requirements and system specification inside the formal database we are able to have direct access to the hidden information and the possibility to integrate them in the given structure.

As it is difficult for the customers to understand these formal descriptions in general, the consistence description must be reduced in complexity and shown to the customer as simulations. This can be done through simulations which concentrate the information to certain aspects of interests for the user. This is the reason for the application of modern virtual reality approaches in this framework. Especially the use of user adapted graphical representations in the requirements phase is very important, as most of the reasons for system failures are related to errors during the requirements specification phase. We will focus on this point in the next chapter.

Using Simulation and Virtual Reality in Requirements Engineering

In this section we introduce visual simulation (VS) and virtual reality (VR) in requirements elicitation. A possible requirements engineering process with VS/VR is defined and a list of expected advantages of this new method is given.

In other fields e.g. architecture or computer aided design virtual reality became more common in the last years [3,7]. In Berlin e.g. the future buildings at the Potsdamer Platz and the new part of the government can be visited by virtual reality (VR) demonstrations. These intuitive presentations emphasise in most cases on structural aspects. On the other hand, simulation tools are seen as the „key technology for realistic support of decisions of dynamic aspects in the future“ [14]. However, the practical use of VR for nearly realistic support of decision in structural aspects and of simulation tools for dynamic issues has been commonly accepted. Thus the question arose whether simulation and virtual reality can be used even in the requirements elicitation phase in railway engineering projects where lots of decisions on dynamic processes (e.g. train schedule) have to be made.

A *VR (virtual reality) system* consists of several effectors like data glove, head-mounted display or force ball and the reality engine with its computer system kernel , several signal converters and control elements [11]. As *VS (visualised simulation) system* we define a reduced VR system based on a simulation system together with enhanced visualisation. It consists of a 2-D geometry database, simulation management and user application that considers environmental, process and technology aspects. It has a formal kernel where the system and environmental aspects can be specified and simulated e.g. based on formal mathematical theory. Thus the formal described subprocesses or scenarios can be carried out and presented to the customer by the VS system.

The development of a VR system can be very expensive. Therefore it has to be carefully decided whether to use a VR or a VS system. Regarding the application field of railway systems development, the main aspects to be considered are automatic train protection (ATP), automatic train operation (ATO) and the supporting systems like regulation and interlocking. In order to discuss these dynamic aspects we need a good simulation of the process, abilities for interacting with the simulation and an intuitive visualisation. Thus the VS system is appropriate.

Figure 2 shows the relation between simulation, visualised simulation and virtual reality with respect to .“kind of visualisation“ (1-dimensional, ..., multimedia) and „interaction“ (no-interaction, real time interaction). The concepts „simulation“ bases on an operational kernel, with 1-dimensional visualisation (tables) and no direct interaction. Visualised Simulation is an enhanced simulation with 2-D or 3-D visualisation and an interaction. The next enhancement classifies the concept „virtual reality“ that has additional multimedia visualisation and probably real time interaction.

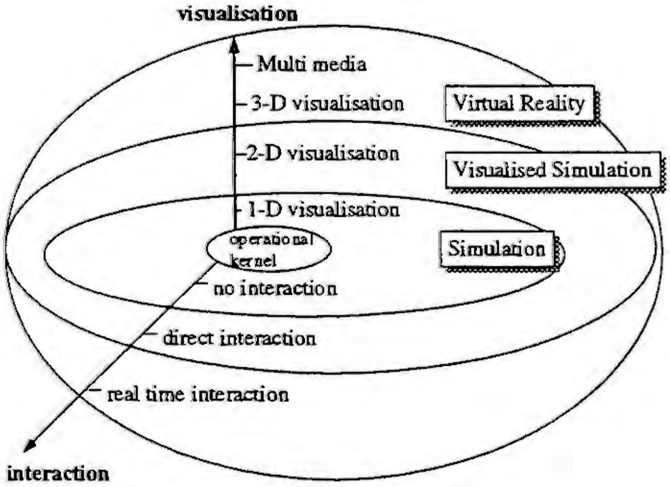


Figure 2: Dimensions of Visualisation

The train analysis and simulation tool set called TRANSIT [9] can be used as VS system. TRANSIT determines various features of performance in local traffic and automatic guided train areas and consists of the four main modules train schedule, control system, interlocking system and process that are connected to each. It models and simulates the process as well as the control and the interlocking system. Hence, TRANSIT considers reality aspects in railways.

An overview on typical application fields in railways systems engineering and their requirements toward visualisation and simulation is given in Tabular 1.

Application Field	*Discussion Partner Customer Side*	*Discussion Partner Supplier Side*	*Representation*
structural aspects: interior design of trains, comfort of disposition system	customer/user	supplier (marketing staff)	virtual reality
performance, accordance of train schedules, validation of schedules	user	developer (technical staff)	simulation
discussion of complex dynamic behaviour aspects as variants of train schedules, traffic load,	customer / user	supplier (marketing staff) /development staff	visualised simulation

Tabular 1: VS, VR and simulation in railway systems engineering

As depicted in figure 1 the introduced framework considers simulations of the specified system at both levels the requirements and the system specification level. In the following we focus on the requirements specification level. In order to achieve a requirements specifications as complete and customer-oriented as possible, the use of VS and VR systems have to be integrated in the entire requirements engineering process. By experience we know that new methods and tools do not improve the engineering process and its result unless they consider the usual applied development process of the project group. In conventional processes the developer elicits the requirements in discussions with the customer/user on basis of ER-diagrams, OOAD specifications, a.s.o.. All these representations are paper-oriented and non of them can be simulated on the computer. The human-oriented virtual reality presentation, we adapted to the framework, is natural for all participants so that it can be easily integrated in the process by substituting the paper documents.

Figure 3 shows the proposed new process with VS/VR that refines the triangle part (requirements, requirements specification, simulation and documentation) of figure 1. It is a cyclic iteration starting at step 1, where the customer provides the developer with a first collection of requirements formal or informal described. These requirements are transformed by the development team into formal scenarios, that can be carried out. The next step is to integrate these scenarios in the VS or VR system. Then the simulated scenarios based on the entire VS / VR System are presented to the customer for interactive discussion. Thus existing requirements can be correctly defined and new features can be additional demanded by the customer due to the presented system. In several iterations of these four steps both customer and developer refine the first rough system specification until they agree on all aspects.

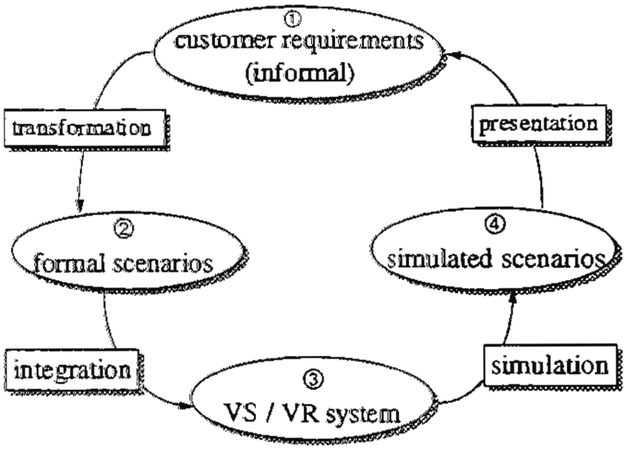


Figure 3: Process of Requirements Engineering

According Davis [4] the relationship between requirements and design becomes more complex in problems where solutions lie in a combination of software and hardware. This combination is typical for railway engineering projects where software (control system), hardware (trackside elements) and integrated firmware build up the required system. Davis provides the following six points of traits that problem analysis techniques should exhibit:

1. Facilitate communications
2. Provide a means of defining the system boundary
3. Provide a means of defining partitions, abstractions, and projections
4. Encourge the analyst to think in terms of real requirements not in solutions
5. Allow for opposing alternatives but alert the analyst to their presence
6. Make it easy for analysts to modify the knowledge structure

Our approach supports developer and customer with good communication facilities (animation) and regards the system boundary by simulating all scenarios with respect to the VS rsp. VR system. Abstractions and projections can be carried out chosing different scenarios. Thus it establishs a bases for agreement, provides a baseline for validation and verification and reduces the development effort (less errors, focussing on important requiremets).

By visualisation and animation of requirements the level of abstraction for discussions is pre-defined, concepts can be concretised and several views of the system can be interactively discussed. Furthermore, developer and customer are compelled to formalise their ideas due to the fact that both systems (VS and VR)have a formal kernel. This will bring out a lot of misunderstanding in early stages. Another advantage is the consideration of overall system aspects by the integration step. Though the complexity of the comprehensive system description increases, the discussion can still be easily followed due to the intuitive representation.

The following theses sum up our expectations on the improvement for requirements engineering based on VS/VR:

1. VS/VR leads to a communication between customer and developer with less misunderstandings, ambiguity and clearer semantics
2. The customer is better involved in the decision process. (It is much easier to involve him.)
3. The developed system is much more customer-oriented and adapted to their needs. (Due to these 2)
4. The developing process is more efficient (time, budget). (Faults and misunderstandings can be detected in an early stage of the project)
5. Easier integration of the system. (Based on the VS rsp. VR system in the integration step)

There are some items (4-6) our approach do not fulfil but we keep them in mind and work further on them. Anyhow, our approach provides customer and developer with lots additional advantages that will lead in our opinion to better system specifications.

Conclusion

In this paper we presented a first snap-shot on the idea of combining virtual reality representations rsp. simulation tools with modelling methods in the phase of requirements engineering in order to enhance the output of the phase and the quality of the later railway system. In the future we will further concretise our structural model and emphasise on the earliest part of the lifecycle respectively the incoming informal specification of the customer and its transformation in parts of formal and informal specifications. While the application of the introduced framework including the specification of requirements in scenarios gave us a good feedback during the specification of ETCS (European Train Control System), we hope to integrate soon the virtual reality aspects. This would be a further step to error-minimised specifications of complex and safety-critical railway systems.

References

[1] Boehm, B. W.: A Spiral Model of Software Development and Enhancement. ACM SIGSOFT Software Engineering Notes 11, 1986

[2] Booch, G.: Object Oriented Design with Applications, The Benjamin / Cummings Publishing Company, Redwood City, USA, 1991

[3] Chapin, W. and Lacey, T. A. and Leifer, L.: DesignSpace: A Manual Interaction Environment for Computer Aided Design. In: CHI'94, Boston, Massachusetts USA, 1994.

[4] Davis, A. M.: Software Requirements - Objects, Functions and States, Prentice-Hall,1993.

[5] Hansen, K. A.: Formalising railway interlocking systems. In Proc. Nordic Seminar on Dependable Computing Systems, Institute of Computer Science. Technical University of Denmark, DK-2800 Lynby, Denmark, 1994.

[6] Janhsen, A., Schnieder, E.: Entire Design Process: Introduction of a Mental Reference Model for Systems Specifications. International Conference on Applications of Artificial Intelligence in Engineering XI, Clearwater, Florida, Computational Mechanics Publications, Ashurts, 1996

[7] Kempfer, L.: Maximizing Visualization. In: Computer Aided Engineering, June, 1996.

[8] Naftalin, M. and Denvir, M. T. and. Bertrani, M, editors: FME'94: Industrial Benefit of Formal Methods. LNCS 873, Springer, Berlin, 1994.

[9] Nökel, K. and Rehkopf, A.: Betriebsführung von Nahverkehrssystemen mit dem Simulationswerkzeug TRANSIT, Signal+Draht , 88, 3, 1996, 15-18.

[10] Papaspyrou, N., Skordalakis, E., Vescoukis, V. C.: *A Logic-Based Framework for Reasoning Support in Software Evolution*, In: Advanced Information Systems Engineering, 8th International Conference, CAiSE, Proceedings, Springer Verlag, Berlin, 1996

[11] Pimentel, K. and Teixeira,K.: Virtual Reality - Through the new looking glass, Windcrest Books, 1992.

[12] Royce, W. W.: Managing the Development of Large Software Systems. Concepts and Techniques, Proceedings, WESCON, 1970.

[13] Rumbaugh, J.; et al.: *Object-oriented Modelling and Design*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1991.

[14] VDI Gesellschaft Fahrzeug und Verkehrstechnik: Simulation und Simulatoren für den Schienenverkehr, VDI Verlag, Germany, 1995.

MSO: Modeling - Simulation - Optimization in Large Scale Traffic Systems

A. Graber, MBC Model Based Consulting, 8703 Erlenbach, Switzerland
M. Mouthon, GfAI Group for Applied Informatics, 8105 Regensdorf, Switzerland

1. Abstract

This paper describes the methodology used to optimize the entire net structure for the cargo traffic of Swiss Rail. Swiss Rail has around 1800 stations for cargo, of which aprox. 30 are shunting stations (marshaling stations).

The transportation process is subdivided into three sub processes:

1) local collecting of wagons (from station of origin to shunting station).
2) transporting over a long distance (from shunting station to another shunting station), and
3) local distribution of wagons (from shunting station to destination station).

In a period of one week, around 90'000 wagons are transported.

The overall objective is the minimization of costs whilst maintaining given transport quality standards. However, the exact costs of shunting and transportation are not really known, as the cargo division of Swiss Rail shares resources with the passenger division . Therefore we concentrate here on the optimization of the utilization of resources.

Due to the combinatorial complexity of the problem a mesoscopic approach is chosen by which some parts of the system are modeled in detail and others only rudimentarily whereby part of the structure and some relevant parameters have to be iteratively tuned during the validation.phase..

The cyclic MSO-approach applies at two levels:

1) **Model optimization:** The results of a simulation run are compared with the real world. The structure of the models is adapted and the different parameters tuned in such a way that the accuracy of the results improves. This is repeated iteratively until a stable model with known and acceptable accuracy is achieved.
2) **System optimization:** The results of a simulation run with the stable model are analyzed. The network structure, corresponding time table and routing matrices are modified iteratively until an optimal performance of the overall system is reached.

In this paper emphasis is put on the second MSO cycle - optimization of the rail network - and we will only describe briefly the output of the model optimization process i.e. the structure of the models used.

2. MSO: Model Optimization

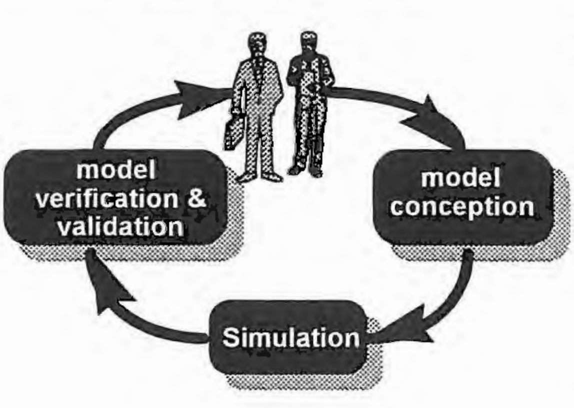


fig 1.
The model optimization cycle. The two persons represent the domain know how (SBB) and the operations research know how.

The output of the model optimization cycle is a set of 3 models which are used for the system optimization:

1. **Scenario generator**: The scenario generator enables the user to modify interactively the size and location of the collecting and distribution zones, the time tables, the volume of wagons to be transported, and other relevant parameters like shunting capacities etc.. Whenever zone modifications occur, time tables have to be generated automatically for local as well as for long distance traffic. The consistency of all relevant system parameters is examined on request before a new scenario is simulated.
2. **Simulation model:** The simulation model takes the consistent data from the szenario generator (network, time tables, capacities and the list of wagons to be transported) and simulates the behavior of the transport chain.

- *Local collection:* Based on the time table, the wagons are picked up at their station of origin and transported to the next shunting station.
- *Long distance transport:* At the level of shunting station is decided on the path wagons have to follow to their destination shunting station. The long distance trains are formed and run according to the time table. Restrictions are considered here like: capacity of shunting station, train length and weight (depending directly from some physical characteristics of the network and indirectly from the timetable which contains the relevant routing information).
- *Local distribution*: At the destination shunting station distribution trains are formed and wagons are transported to their final destination according the time table. No restriction apply here to train length and weight.

3. **Report generator**
   The report generator generates standardized editable text documents (MS-Word) that contains statistics (MS-Excel) at three levels:
   a) static overview (flow related statistics),
   b) statistics about the system dynamic c) detail information about subnets.
   The report generator can display and compare several scenarios in one report. The comparison of a new scenario with a standard well known reference scenario, proved to be one of most efficient methods.

The following figures show some typical graphics produced by the Report Generator.

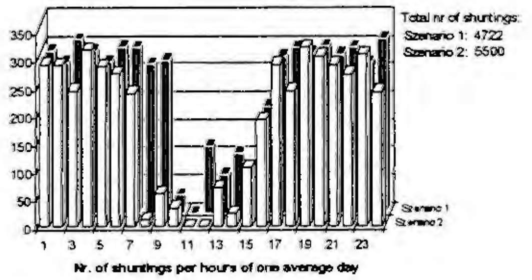


fig. 2:
Comparison of the load per hour in one specific shunting station.

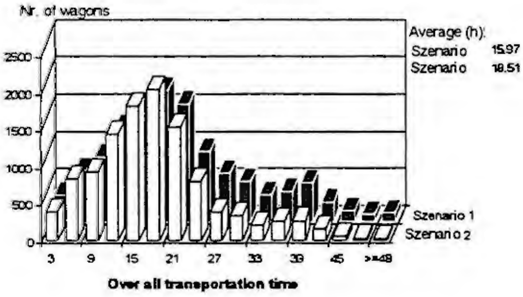


fig. 3:
Comparison of the over all transportation time.

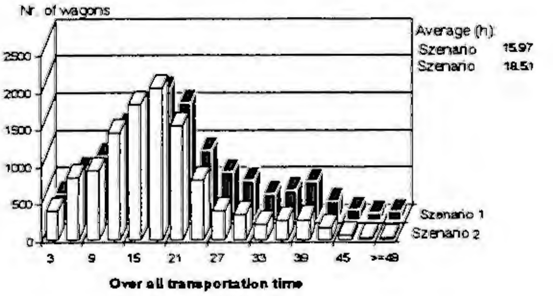


fig. 3:
Comparison of the over all transportation time.

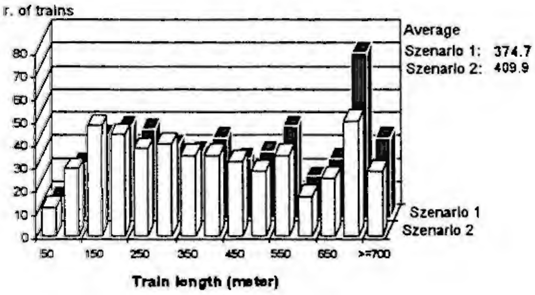


fig. 4:
Comparison of the distribution of the train length.

3. MSO: System optimization

The network optimization results from comparing a new scenario with the reference scenario or from comparing successive scenarios among themselves. The main control variables in this optimization process are:

1. Number of active stations.
2. Number and location of the shunting stations, related local zones and time tables for local and long distance traffic.

3. Time dependent volume of goods to be transported from and to the active stations.

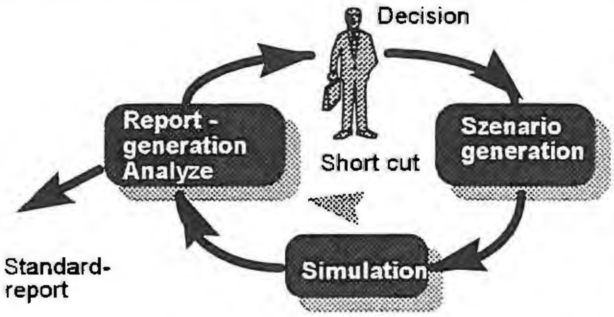


fig. 5
The cycle to optimize the railway system. The person represents the domain know how (SBB).

Consistency requirements among the structure of the modeled rail network, the train time tables and the source and destination of the goods to be transported; make the building of a new scenario an extremely time consuming operation !

Usually classical sensitivity analysis fails to produce meaningful results whenever the model is built on a weakly defined or a highly complex problem situation. The support of a scenario generator is then required to guide the trial and error process towards some optimized solution. This is particularly true here, where more than 20'000 systems parameters could theoretically be varied to produce new scenarios.

The main function of the scenario generator is to produce in a relatively short time, new and consistent scenarios in a format as near as possible from the one used in reality. Among others: consistent routing matrices and time tables have to be generated !

3.1. Modeling the Network

The scenario generator is imbedded in a graphical information system (GIS) platform in which all available functions can be called by a mouse click (Fig. 6).

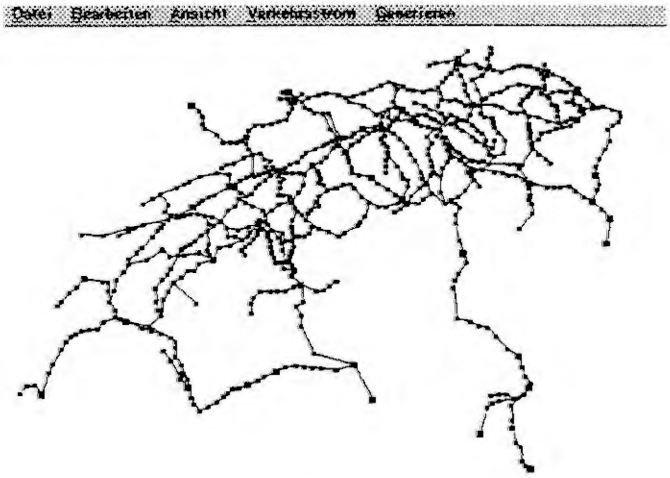


fig. 6.
The graphical user interface

Modifying the Network

Typical modifications of the structure of the local zones consist in :

- eliminating an existing shunting station or defining a new one,
- reassigning a mother station to a different shunting station.

Changing the structure of a zone requires usually new time tables and routing matrices to be defined. By small changes it is possible to modify the existing matrices and time tables manually. Nevertheless relevant modifications of the network within an optimization cycle require usually a full recalculation of routing matrices and time tables.

3.1.1 Modifying the Network

Typical modifications of the structure of the local zones consist in :

- eliminating an existing shunting station or defining a new one,
- reassigning a mother station to a different shunting station.

Changing the structure of a zone requires usually new time tables and routing matrices to be defined. By small changes it is possible to modify the existing matrices and time tables manually. Nevertheless relevant modifications of the network within an optimization cycle require usually a full recalculation of routing matrices and time tables.

3.1.2 Routes

Routes describe the connections between shunting stations. These can be direct, or via other shunting stations. They are time dependent and can be adjusted hourly (two stations can be connected directly in the morning and via a third one the rest of the day for example).

The calculation of the routes represents a complex combinatorial problem. Pure mathematical optimization algorithms - as described in [1] - tend to produce an enormous amount of combinations or to create unstable solutions.

For the calculation of routes in the scenario generator a modified flooding algorithm [2] plays a central role. Trials with genetic algorithm were abandoned, due to poor performance and unsatisfactory results.

3.1.3 Main Time Tables

Collection-, long distance and distribution time tables can be created automatically. Restrictions in the form of time windows and railway line capacities can be given what allows indirectly capacity problems due to other traffic types to be considered

The itinerary of collection and distribution trains is calculated on the basis of the shortest way to and from the shunting stations. Whenever minimizing train kilometers conflicts with minimizing wagon kilometers priority is given to train kilometers.

The long distance train time tables are treated similarly, taking into account the restrictions formulated in the route matrices.

Time tables calculated in this way can only be approximate. Their quality becomes apparent only in the results of a simulation run according to which they can be modified manually. The consistency of such changes is tested automatically.

The simulation model itself contains some functionality to improve the quality of the train time table by overloading trains systematically, or by generating additional trains.

3.1.4 Transport Orders

Transport orders for new scenarios are generated automatically on the base of real life transport orders. The transport demand can be adjusted globally or locally (for example a percentage increase in transport demand of a specific good between two regions). By doing so special attention is paid to the fact that an increased transport demand from region A to region B implies usually an increased transport of empty wagons from B to A.

3.2. Simulation of Traffic

The purpose of the simulation is to generate dynamic results of the transport process The functionality of the model and the generated output is described in the previos chapter .

3.3. System - Optimization

3.3.1 Static Optimization

A powerful characteristic of the scenario generator is the possibility of eliminating unrealistic scenarios by ways of rough flow analysis prior to simulation, accelerating substantially the optimization cycle. This flow analysis produces values for

- Work load in each shunting station due to collecting and distribution traffic
- Work load in each shunting station due to long distance traffic
- Traffic density on each railway line
- etc..

These values can only be calculated approximately by means of flow analysis i.e. ignoring the dynamical dimension of the network. These approximations are nevertheless precise enough to detect very soon unrealistic scenarios. It has nevertheless to be born in mind that due to the time dependency of some routes and of the capacity of some resources the actual usage rate of trains and shunting stations facilities will generally be higher as found by analyzing flows.

The static optimization is performed within the scenario generator and by cycling through the "short cut path" (figure 5), where static results of scenario generator are evaluated in details by means of the report generator.

3.3.2 Dynamic Optimization

With a scenario optimized on the basis of static considerations only, no statements can be made about the overall transport time and the critical train length. This type of results will only be obtained through simulation.

3.3.3 Optimization Cycles

Fig 7 shows all theoretical possible optimization cycles to achieve the best result. As stated before, the cycle containing costs is not uses, as the cost structure is not accurate enough, however reports containing costs have been generated, to demonstrate theoretical savings achieved by new scenarios.

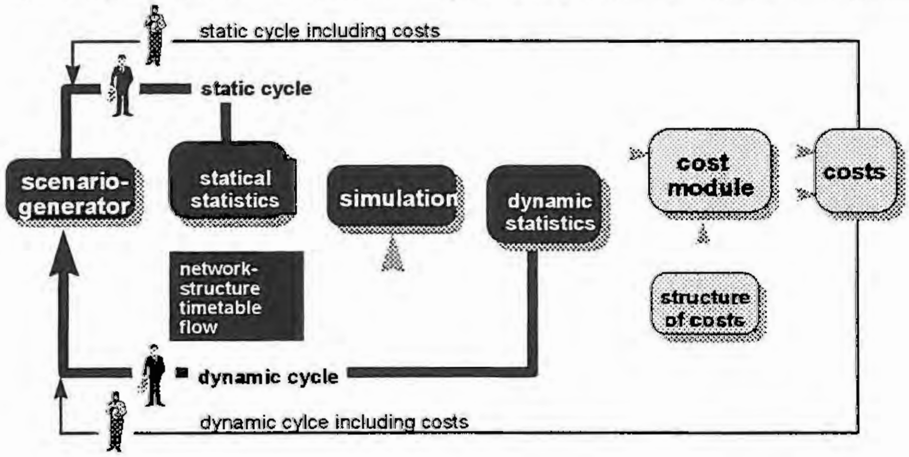


fig 7.
All theoretical possible optimization cycles uses with the model

4. Experience in practice

4.1. General

Systems of the size of a national railway like Swiss Rail are weakly defined systems with many special cases to be treated specifically. The more emphasis is placed on special cases, the more likely we are to loose control over the combinatorial complication of the model. On the other side if relevant special cases are not considered, the relevance of the results will suffer.

The scenario generator is currently intensively used by Swiss Rail in its effort to increase productivity in the good traffic sector. The productivity of the optimization process itself has been substantially increased by using the above described pragmatic and.iterative MSO approach.

Parallel to small network adjustments - such as the removal of singular shunting stations - more involved network adjustments - such as the reduction from 47 to 5 shunting stations - have been evaluated.

Within half of a day, radically new scenarios could be created and simulated, allowing for an immediate first analysis. Through further iterations the required more detailed statements were produced . leading to an optimal solution within a few hours.

User acceptance of the results has been remarkably high due to the interactivity of the optimizing process. The user has been very quickly able to use its deep expertise of the real network at different abstraction levels in the optimizing cycle. Due to the cyclic way of working, the learning success of the user had a positive effect on the efficiency of the search for solutions.

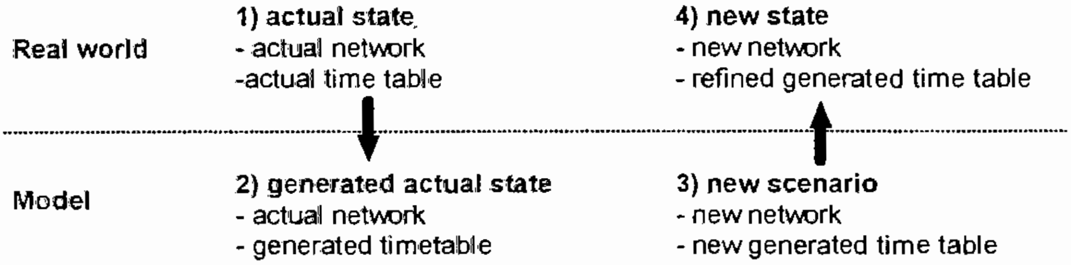
4.2. Accuracy of Simulation Results

At the time the model was validated, no detailed data about the movement of wagons were available at Swiss Rail. For validation purposes 2000 wagons were marked and their movements were recorded during 24 hours.

The discrepancy between model data and validation data ranged from 5% up to 30%, depending on the type of results. At a first glance, this seems to be not precise enough for optimization purposes. Investigations showed however, that the absolute degree of accuracy is a less important factor for the optimization process as the fact that results with the same degree of accuracy are compared.

In order to compare a new scenario with the actual state, the actual network structure is entered into the scenario generator. Actual routing matrices and time tables are ignored and new ones are generated with the built-in algorithms. These matrices and time tables are of lower quality than the real ones, but they enable a meaningful comparison of the "transformed" actual-state with the new scenario.

Experiences in the last years have shown, that the improvement tendency from a "actual state" (1) to a "new state" (4) is equivalent to the one observed from a "generated actual state" (2) to a "new scneario" (4).



Real world	**1) actual state** - actual network -actual time table	**4) new state** - new network - refined generated time table
Model	**2) generated actual state** - actual network - generated timetable	**3) new scenario** - new network - new generated time table

fig. 8
Real world and model

5. Hard- / Software, Performance

Hardware:	IBM PC (486 / 66 MHz)	
Software:	OS:	OS2
	Model kemel:	SIMAN IV
	Model enhancements:	c / C++
	Reports: C++ / MS-Word / Excel	
Performance:	Simulation of one relevant Scenario (1 Week, entire network, full traffic): 55 Minutes	

6. Literature

1) Weißleder D.U. (1989). Das Transportmodell TPM. Eisenbahntechnische Rundschau, 38(11), 685-690.

2) Tannenbaum, A.S., Computer Networks, Prentice Hall 1988

3) A. Graber / U. Ulrich (1992) Modeling and Simulation of the Swiss Railways Good Traffic with SIMAN. 7th. ASIM Symposium on Simulation in Hagen, Band 4, 128-132

4) A. Graber / U. Ulrich / M. Mouthon (1994) Computer Aided Generation of Scenarios for the Optimization of Complex Transport Systems by Means of Simulation, CISS - First Joint Conference of International Simulation Societies Proceedings, SFIT (ETH) Zurich Switzerland, pages 454 ff.

The Impact of Dynamic Speed Adaption on Macroscopic Traffic Flow Simulation

Frank Meißner
Technical University of Hamburg-Harburg
Lohbrügger Kirchstraße 65
D-21033 Hamburg

Abstract

During the last years the technique of macroscopic traffic simulation has gained more importance within the field of online traffic control. Mostly, simulation is used as a means of prediction for traffic behavior in order to test rerouteing decisions before they are applied to traffic network. Hence simulation execution time is a critical resource which has to be dealt with. The simulation model, presented in this article, used to calculate dynamic adaption terms for the simulated speed in any traffic situation. Due to the high calculation cost of these terms a new hybrid simulation model is introduced, which is capable of doing a low cost calculation for fairly homogeneous traffic situations with low to medium load and applying the full speed calculation scheme whenever a significant loss of precision threatens. Different traffic scenarios are investigated and the results for a German motorway are shown. Moreover, the behavior of dynamic speed adaption in an extremely inhomogeneous situation is demonstrated. Finally, the computational profit of the new hybrid model will be shown and discussed.

Introduction

In recent years new discussion has arisen about the usefulness of dynamic speed adaption in macroscopic modeling. The basic approach on traffic flow modeling of Lighthill and Witham [3] treats density as the only dynamic variable and assumes instantaneous adaption of speed according to a static speed-density characteristic ("first order model"). This led to a number of deficiencies which prompted an extension of the model by Payne[4] to include dynamic speed adaption to the static speed-density characteristic and thus giving the model more realistic features. However, this extension to a "second order model" was counterbalanced by doubling the numerical effort, which might become a problem for online applications in larger networks. In addition, while improving some deficits of first order models, dynamic speed adaption included other malfunctions of its own, as pointed out by several critics.

In this paper we present the results of a thorough comparison of both modeling approaches based on real data ranging from free flow to full congestion. While critics from both sides so far have focused on extreme and somewhat artificial situations we put emphasis on the behavior of both approaches in critical but real situations.

Introduction of the basic simulation model

The simulation is based on a macroscopic model which has been initially introduced by Payne[4] and later on been refined by Cremer[1]. Three variables *density*, *mean speed*, and *traffic volume* are chosen to describe traffic behavior. Each of the variables is computed for so called *segments* at discrete time steps of appropriately ten seconds. Segments are formed by subdividing a road into smaller stretches of about 500 meters length. Time step and segment length are dependent on each other by the precondition, that no vehicle shall be able to pass a full segment length during one time step, in order to avoid a loss of vehicles while calculating. The variables are calculated by the following time difference equations which have been derived from partial differential equations proposed by Lighthill, Witham[3] and Richards[5].

Traffic density in segment i at time step $n+1$ is calculated using the mass transport equation:

$$c_i(n+1) \quad = \quad c_i(n) + \frac{T}{\Delta_i}[q_{i-1}(n) - q_i(n)] \tag{1}$$

The calculation of mean speed for same segment and time step is done by:

$$\begin{aligned} v_i(n+1) \quad = v_i(n) \quad & + \frac{T}{\tau}[V(c_i(n)) - v_i(n)] \\ & + \frac{T}{\Delta_i} v_i(n)[v_{i-1}(n) - v_i(n)] \\ & + \frac{\nu T}{\Delta_i \tau}\left[\frac{c_i(n) - c_{i+1}(n)}{c_i(n) + \kappa}\right] \end{aligned} \tag{2}$$

And finally volumes are computed via:

$$q_i(n) \quad = \quad \alpha c_i(n) v_i(n) + (1-\alpha) c_{i+1}(n) v_{i+1}(n) \tag{3}$$

The significant behavior of the model is defined by the steady–state speed–density relation:

$$V(c) \quad = \quad v_f u_2 \left[1 - \left(\frac{c}{c_{max}}\right)^{l(3-2u_2)}\right]^m \tag{4}$$

with:

v_f the free flow maximum speed

c_{max} the maximum density at full congestion

l, m parameters used to define the form of the speed–density relation

α weighing factor

κ density parameter

ν sensitivity factor

τ time constant

$u2$ speed limit from the interval $[0, 1]$

The parameters mentioned above have been properly adjusted for the simulations on the following pages to:

	speed–density parameters				model parameters			
No. of lanes	v_f	c_{max}	l	m	α	κ	ν	τ
1	65	100	1.4	3.8	0.8	10	21.6	0.00944
2	122.4	200	1.4	3.8	0.8	20	21.6	0.00944
3	126.0	300	1.4	3.8	0.8	30	21.6	0.00944
	km/h	veh/km				veh/km	km^2/h	h

Table 1: Adjusted simulation parameters

Most interesting for the following discussion is the calculation of speed through the three rightmost terms. The first term, the speed–density relation, which is shown in Figure 1, defines the main behavior of traffic.

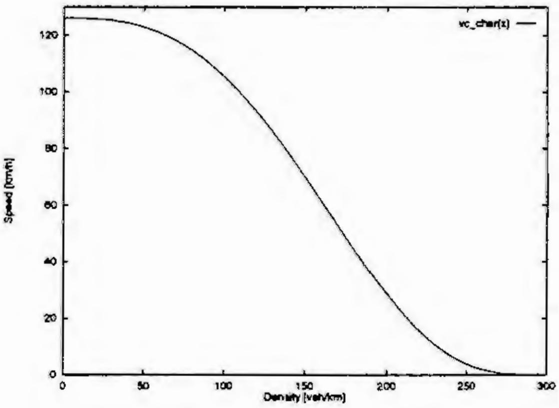


Figure 1: Speed–density relation of a three lane motorway

The second term represents the convection of speed within a given distance whereas the third term reveals the driver's anticipation of traffic flow in the next downstream segment. The latter two terms are called dynamic speed adaption terms.

These basic equations are further on refered to as *complex calculation* or *complex model.*

Simple Model

Simpler calculation models have been favored by several authors [4][6] and supported by others [2]. We have chosen to investigate a model, which differs from Equation (2) of the complex model, by the omission of any speed adaption through dynamic terms. The right hand side of this model consist of the speed–density relation only, which is dependent on density value c_i of time step $i+1$.

$$v_i(n+1) \quad = \quad V(c_i(n)) \tag{5}$$

This model will be refered to as the *simple model* or *simple computation.*
The decision to use the speed equation of the form as mentioned above was done for implementation reasons of the algorithm. In order to use both methods in parallel switching from one to another depending on traffic conditions this scheme proofed to be a solution of less computational cost.

Comparison of the two models

Obviously, the complex model is almost twice as expensive to calculate as the simple model, which makes it valuable to investigate whether it is acceptable to skip dynamic terms for the sake of a calculation time speed up. Hence, it is necessary to classify different situations in order to identify potential to spare calculation cost.
Therefore both dynamic terms of the complex model speed equation have to be looked at. The speed convection evaluates to zero if speed in the last upstream segment is identical to the actual segment's speed. The driver anticipation term also vanishes if actual segment's and downstream following segment's density are identical. This leads to the assumption that for homogeneous situations the calculation of simple and complex model will not differ too much, thus making the cheaper calculation of the simple model for these situations applicable.

First, we chose two almost homogeneous situations with low and high traffic load conditions on the German motorway A9, which is very well surveyed including entering and leaving volumes for any on/off ramp[1]. For this paper we focus on a section of this motorway with three lanes which is 30.34 km long and equipped with a speed control system. Data from the street has been collected by ten detectors, which are placed in irregular intervals in the road. During most of the scenarios we will refer to, no speed limits have been enabled. Whenever a limit was active we have taken it into consideration for the simulation.

Figure 2 shows a rough sketch of the chosen motorway section.

[1] SIEMENS AG/Germany provided this excellent data.

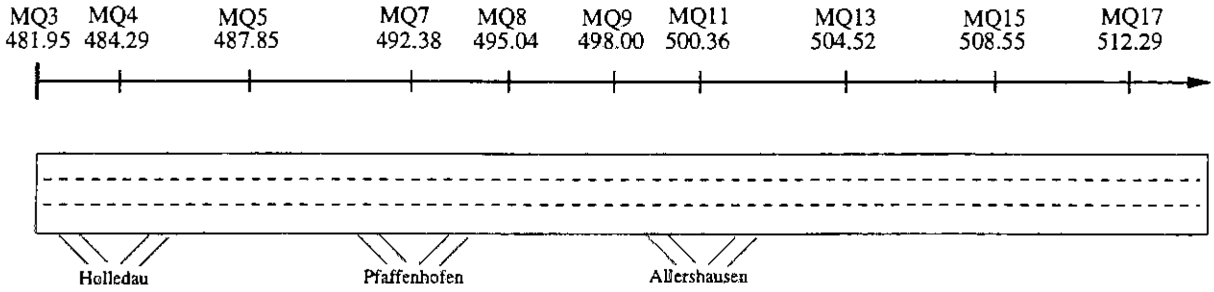


Figure 2: Topology for our investigation from motorway A9 near Munich

This topology was easily translated into a topology suitable for our simulation program SIMONE[2] by subdividing the road into segments of 500 meters with three lanes. On/off ramps are represented by traffic junctions of no length of their own. Therefore different flows had to be separated very well to avoid any non natural interference such as making vehicles leave at the same ramp they have entered.

The results of our simulations from Sunday 1994/04/03 at MQ7, which is at a distance of 10.5 kilometers from network entry, show a good approximation of volumes for complex and simple model. Flow conditions have been very moderate reaching at maximum of one third of full capacity.

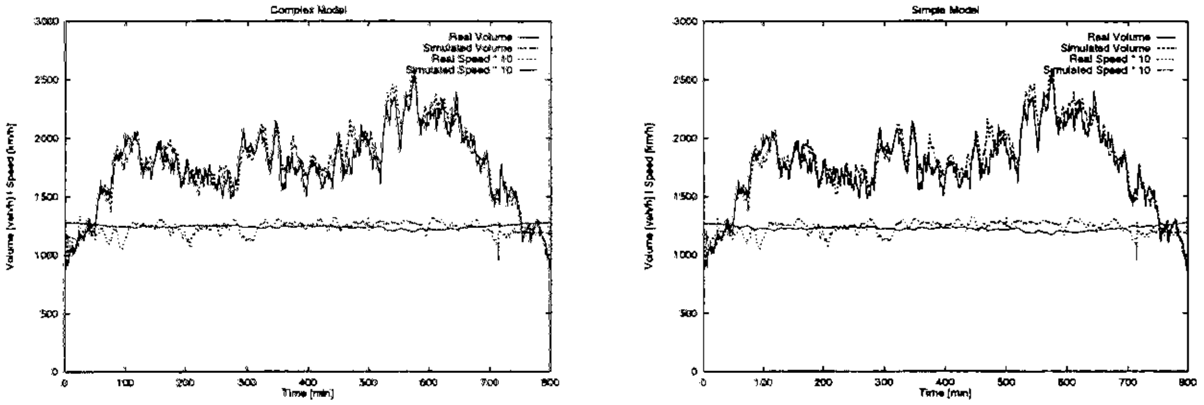


Figure 3: Volume and speed at low capacity demand for complex and simple model calculation, measured at MQ7

Simulated mean speed is always a little smoother than reality due to both model's tendency to dispersion. If the measurement point was situated closer to the network entry, the speed approximation would have been much better than shown above. But the results are still good enough, to reveal the real situation.

For conditions of higher traffic demand the situation looks quite similar even if traffic flow is a little more unstable. As a means of comparison we needed to calculate a performance index for every simulation run by:

$$perf = \sum_{i=3}^{17} \sum_{n=0}^{timesteps} (q_{real_{mq_i}}(n) - q_{sim_{mq_i}}(n))^2 * \gamma + (v_{real_{mq_i}}(n) - v_{sim_{mq_i}}(n))^2 \qquad (6)$$

with $q_{real_{mq_i}}(n)$, $v_{real_{mq_i}}(n)$ measured volume and speed and $q_{sim_{mq_i}}(n)$, $v_{sim_{mq_i}}(n)$ simulated volume and speed at detector MQi, $\gamma = 0.001$, and for nonexisting detectors we define $q_{real_{mq_i}}(n) = q_{sim_{mq_i}}(n) = v_{real_{mq_i}}(n) = v_{sim_{mq_i}}(n) = 0$

The performance indices of both models compared to real data are shown in Table 2.

[2] *Simulation of Motorway Networks*

Model	performance index	
	low demand	high demand
Complex [scalar]	2169948	2325740
Simple [percent of Complex]	104.1 %	106.1 %

Table 2: Comparison of performance between complex and simple model

Obviously, simple model's computation produces acceptable results for homogeneous — or almost homogeneous — situations, but complex model is superior in any case. In case of higher traffic load the simple model's performance is a little worse compared to cases of lower demand, because higher traffic load always tends to less stable behavior than low load conditions.

In contrary simple and complex model differ quite more if traffic volumes and speed are not as smooth as they used to be in the cases shown so far. The following scenario has happened on 1992/09/13 which was a Sunday as well. Starting in the early evening at 19:25 two lanes were blocked for about two hours. The simulation results and the real traffic behavior are shown in the following figure:

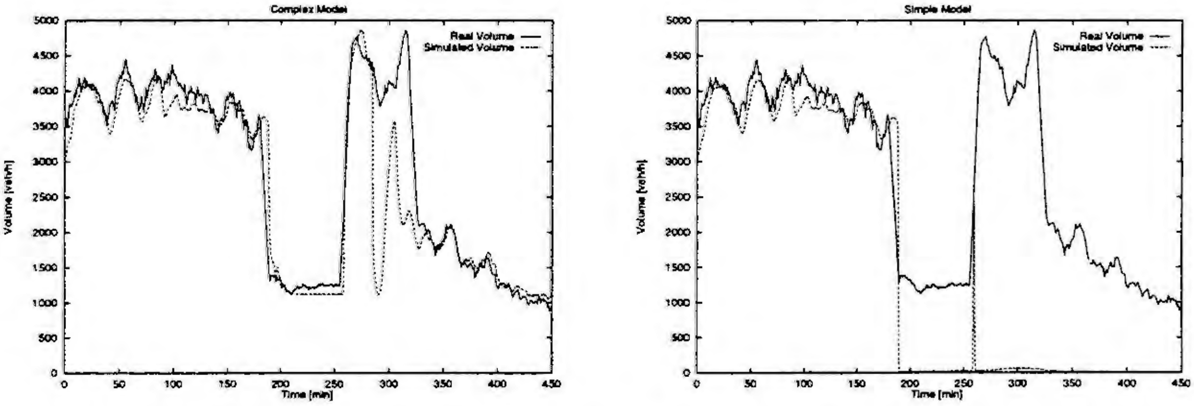


Figure 4: Real and simulated volumes (complex and simple) of MQ7 with two lanes blocked for about two hours

In the case of this two lane blocked situation both models cannot completely succeed in reflecting the congestion. The "raw" simple model did not recover from zero speed at all, because speed only relies on density, which had reached its maximum. As a result the speed kept evaluating to zero, thus allowing nobody to leave the congested area. To overcome this disadvantage often a minimum speed of around 5 km/h is introduced to the simple model, which we did not do because we favor complex model's capabilities in such cases.

In contrary, the complex model's simulation always regains normal flow because of the dynamic speed computation. But in reality the leaving volume at congestion head is not at theoretical full capacity which is an inherent assumption done through speed–density relation, which results in too fast congestion dissipation and some over reactions.

Nevertheless, the results of the complex model are still much better than the simple model's results.

Conclusions

Both models deliver pretty good results for homogeneous traffic situations at low to medium traffic demand, whereas simple model's computation scheme is preferably cheaper. Inhomogeneous and higher load conditions are better reflected by the complex model, even for an extreme situation as described above. As a result we combined both schemes in our simulation tool by introducing a simple mechanism to decide which method to apply. This new *hybrid model* changes computation from simple to complex, whenever mean speed drops or density exceeds a prespecified level. It appeared that if a precision loss of 5% was considered tolerable, dynamic speed adaption has to be done whenever speed lowers to less than 110 km/h or density exceeds 10 $veh/lane$. Applying these parameters to the simulation scenarios delivered computation times as shown in Table 3.

	Execution Time	
Scenario	Complex Model [sec]	Hybrid Model [Percent of Complex Time]
low demand	30	95 %
high demand	32	104 %
two blocked lanes	31	97 %

Table 3: Execution times for different scenarios

Obviously the speed gain is terribly bad, which is on one hand a consequence of the mechanisms which had to be added to the program in order to decide whether to calculate simple or complex model. On the other hand the cost for managing and storing segments, roads, and intersections appears to be remarkably high compared to dynamic term computation.

As a matter of fact the poor impact of avoiding dynamic terms' evaluation on execution time does not justify the loss of precision in simulation results. But as computational effort is still high we are looking forward to reduce data management cost within our program. To our minds further emphasis should be laid on improvement of dynamic terms to increase simulation performance.

References

[1] Michael Cremer. *Der Verkehrsfluß auf Schnellstraßen — Modelle, Überwachung, Regelung.* Band 3 der Reihe: Fachberichte Messen, Steuern, Regeln. Springer–Verlag, Berlin — Heidelberg — New York, März 1979.

[2] Carlos F.Daganzo. Requiem for Second–Order Fluid Approximations of Traffic Flow. *Transportation Research*, 29B(4), 1995.

[3] M.J. Lighthill and J.B. Witham. On kinematic waves. I. Flow movement in slow rivers. II. A Theory of traffic flow on long crowded roads. A229:281–345, 1955.

[4] H.J. Payne. Models of Freeway Traffic Control. — Math. Models of Public Systems. In *Simulation Council Proceedings*, 1971.

[5] P.I. Richards. Shockwaves on the highway. *Opns. Res.*, 4:42—51, 1956.

[6] P.I. Richards. Linear and nonlinear waves. New York, N.Y., 1974.

DISTRIBUTED TRAFFIC SIMULATION IN JAVA — FIRST FINDINGS

Stephan Schnittger
FAW — Research Institute for Applied Knowledge Processing
Information Science for Traffic Management
Helmholtzstrasse 16, D-89081 Ulm

Abstract: Java is very suitable for developing custom oriented applications like traffic simulation environments because of its conception and its foreseeable standardization. The lack of performance compared to compiled languages like C++ can be made up by distributed computing, by machine-oriented programming libraries or by presently developed Java processors. There is quite a number of items worth to improve in detail. This encompass available development environments as well as class libraries. The rapid development and the general interest in Java itself will however increasingly improve the situation.

Situation and Objectives

Traffic simulation environments are developed at the research institute for application-oriented knowledge processing (FAW) since its foundation (See e.g. Mock93, Zeller93). This simulation environments were used in the scope of the development of intelligent driver assistance systems (Mock94, Zeller96). We want thereby improve existing traffic simulation models and methods by using object-oriented programming languages and development tools with regard to configurability, modularity and reusability.

If you apply or further develop simulation environments, there is principally the problem, that the number of available models for the simulation considerably rises. It has to be assured furthermore that the know-how being reflected into this models can be used by the users in full extent. At the FAW we have dealt with the question of model administration and her realization with progressive development tools.

Likewise of interest is the fundamental suitability of the simulation environment to include traffic models which describes traffic flow especially under influence of RTI-systems (see e.g. Schnittger91) resp. to integrate approachs and tools suitable for the examination of forecast techniques in traffic information facilities (see. e.g. Leutzbach88, Jenni95).

The current developments in the scope of the programming languages and development tools towards the direct support of Client/Server systems and Intra- or Internetapplications are very interesting.

We also have dealt with the question of the suitability of Java in this area of application with this background. This paper describes our first experiences with the realization of a distributed traffic simulation environments using Java.

Specification questions

The design of the system had to contain basically the fundamental elements for the administration tasks and control structures. The following points give a summary of the requirements which were made on the concept:

- Build up a traffic infrastructure which can adequate represent both detailed driving sequences of operations of a vehicle and macroscopic traffic flows.
- It should be possible to integrate the available models with as low as possible costs in the new simulation environment.
- Control instances and -structures have to be created with which the distributed Client/Server strategy can be realized. The following boundary conditions have take into account to this:
  - Central data of a simulation run must be able to be accessed distributed. The transactions have to be synchronized.
  - There are distributed objects in the simulation run which communicate each other and exchange information and data with the central objects.
  - There are distributed clients which have various access authorizations and function-related hierarchies.

The type environment had to be conceived recently so that the administration of the simulation is specified to the new requirements and possibilities for the transcription of the intended simulation tool in the substantial. For the realization of the simulation objects and models could be fallen back upon available algorithms, so that the object trunks had to be developed in the substantial and the contents could be bound by proportionally simpler translation.

Choice of the development environment

Different development environments, partly situated only in the beta stage, were tested to the transcription of the demands and boundary conditions mentioned above. Some of the Java Integrated Develop Environments (Java-IDEs) be based on development surroundings which were designed for the development of C ++ programmes often. Whereas the development environment of Sun was written completely in Java. The IDEs dating from the conventional programming languages have the advantage of a higher operating speed, but mostly at the same time the disadvantage of their incomplete Java integration. The IDE dating from Sun has however a weak performance, an inadequate support of graphical interface tools and typical foreign aids like class browser up to the last beta version. Beyond this the operation of the development environments is unconventional.

There is besides quite a number of development environments based on extended editors which were enlarged by essential functionalities like project administration, class browser as well as partly effective debugger.

No tool could currently be found at the market, which fulfilled all of our posed requirements on a development environment. Therefore we used various tools parallelly at the development depending on focus of the implementation work.

We used the IDE-tools for building the user interfaces, for the structured development of the classes and for documentation. However for implementations of system near classes and methods as e.g. thread control, distributed objects and the imbedding's consisting programme code we took the Java-Development-Kit (JDK).

Coarse structure of the system design

Some of the concepts used subsequently shall in the following be defined for the better understanding at first.

In the Java run time environment either an application or an Applet can be executed; subsequently is with regard to simplicity only the term application used. Java knows two basic kinds of objects: 'normal' objects and Thread objects. Threads are parallel (or virtually parallel) execution paths in each of these applications. The distribution of the real simulation process is made either by the local instantiation of the simulation classes defined on distributed servers or by instantiation on the desired target computers. (Basic requirement for this is the ready-to-receive state of the desired target computer)

With these concepts the system architecture shown below schematically can be explained better. The central server application is executed on a central server, still arbitrarily further model servers can be available next to this. Clients can access these servers and provide either only front-end-functionality or also local computing performance.

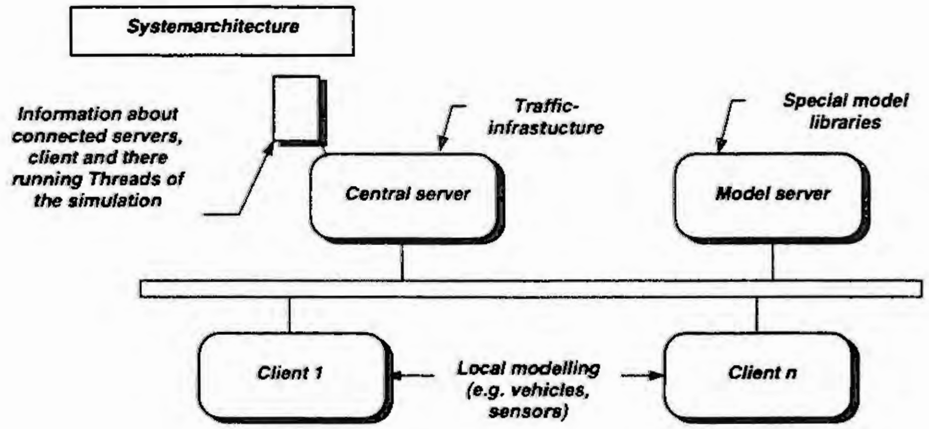
The application on the central server manages

- The simulation infrastructure
- All generic class libraries of the simulation
- Static information about which all further servers and clients connected have from each other knowledge
- Dynamic information about all Threads which locally occur in the further servers and clients involved.

Servers or clients, this one either take part in the simulation process or build up an independent parallel simulation, make use of this information and must take care that it is registered also there simultaneously. The individual computing processes of the simulation can take place only in the servers or in the servers and in the clients distributed.

A model server is built up by registering itself at the central server, obtaining required generic classes of there and executes a local modelling then. New model objects are generated by subclassing of these generic classes and following model implementation to this. For example this could be special model objects like sensors for vehicles or vehicles modelled for a particular case. This can for example also be the imbedding of special available program libraries for the numeric computing of the vehicle dynamics.

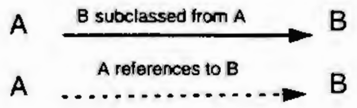
The clients register themselfs also at the central server. The clients can take on in principle pure output functions, i. e. the computing processes occur completely in the involved servers. The clients can also fetch the required classes of the servers, create local instances or derive base classes also to carry out own modellings and involve the corresponding instances in the simulation.


Systemarchitecture
Information about connected servers, client and there running Threads of the simulation
Traffic-infrastucture
Special model libraries
Central server
Model server
Client 1
Local modelling (e.g. vehicles, sensors)
Client n

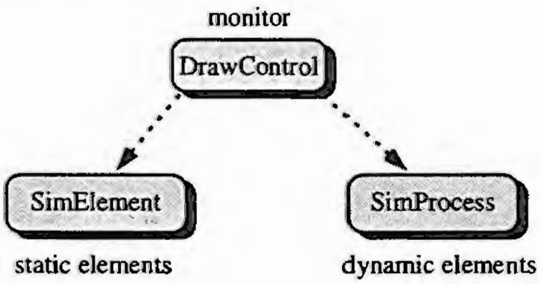
Development of the simulation system

The simulation system consists of static and dynamic objects. All of them behave during the traffic simulation as static according to the intended simulation invariable objects described, e.g. road elements. Dynamic elements are then all variable objects like vehicles. Composite elements like traffic light signals consist of static and dynamic objects. Besides these real objects according to the simulation there is a need for IO interfaces which provide for example graphical presentation or file access.

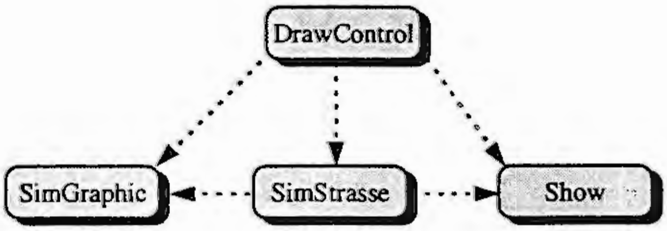
The supervision and control of all simulation objects, particularly in distributed environments, requires special methods. A monitor was developed for supervising all running Threads in the simulation environment as well as those on other engines. Generally valid notations in the following representations are:



The following figure shows the fundamental relations between the simulation elements and the monitor:



All simulation elements have a file interface and a possibility for the graphical representation and interactive manipulations. This is outlined schematically with the following figure at the example of the static simulation element "SimStrasse":



The supervision instance "DrawControl" takes a reference on a simulation object, in this case the static basic class "SimStrasse". The graphical presentation of a road object and the interactive manipulation possibility are converted in the classes "SimGraphic" and "show". "DrawControl" also keeps a reference to guarantee interactions between windows and graphical representations of the other simulation objects. This procedure can considerably simplified with the announced new "Java Beans" (JavaBeans96).

Traffic infrastructure

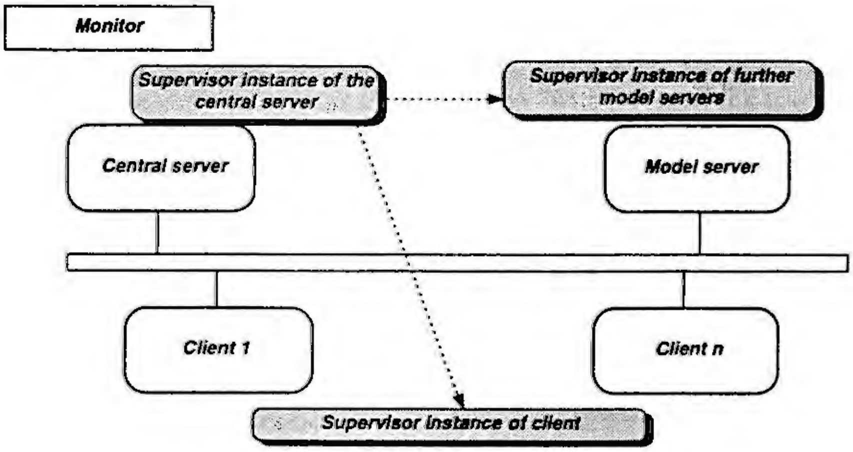
The design principle of the simulation roads is based on the primitives of the real road design. The road elements created so have a relatively high fineness of road description which also allows not only to show vehicles as mass points but with a finer movement mechanism.

On the other hand a so fine-grained representation is not necessary for the simulation of large road networks and obstructive according to the Performance. A macro language with which simulation objects can be built up about the file interface and stored therefore was integrated into the road objects so that abstracted macroscopic road sections internally consist of the above primitives.

They can be abstracted or refined at any time in the course of the simulation by this description of her structure in form of the macro language. The traffic flow is shown then macroscopically in this macroscopic track section and correspondingly microscopically on a road shown microscopically. A remodelling of macroscopic into microscopic vehicle driver elements takes place at cross-sections where macroscopically described roads meets microscopically described roads. So on the one hand a traffic network can be simulated extensively and on the other hand be viewed an individual vehicle full particulars since all vehicles situated in its nearer environments are simulated microscopically.

The supervision instance

It was pointed out to this that all simulation objects and the graphical objects are checked by a supervision instance so that the simulation particularly remains manageable in distributed environments or on multiprocessor systems. The Threads provided by Java well parallel execution paths in the process which have own data areas are starting point of the considerations. These Threads can now depending on environments occur sequentially after the priority principle on a single processor, parallel on multiprocessor systems or parallel on multiple machines.


Monitor
Supervisor instance of the central server
Supervisor instance of further model servers
Central server
Model server
Client 1
Client n
Supervisor instance of client

The supervising unit consists of the instance on the central model server and of the supervision instances on the model servers and the clients. The supervision instances on the model servers and the clients are not different, the supervision instance has however the additional task on the central server to look after these "agents". On every model servers and clients becomes a such supervision instance built up at the start of a simulation, this one holds contact with the supervision instance on the central model server.

Future Extensions

The following conceptional expansions of the simulation environment are planned:

- A version administration for simulation components is absolutely required for an industrially utilizable tool. The version administration shall be integrated by a completion of the generic classes to all simulation objects. The entries in the version database can by subclassing a model class or by modifications of so one be carried out in a simple way. The recording has however manually to be activated to distinguish between program corrections and new models.
- The large initial overhead, which is connected with the implementation of such simulation environment, lead to large costs. These costs should be distributed on the users in dependence of her demands. The underlying thought been based on the dynamically scalable and accountable use of program parts. Only the proportional claims on the relevant model parts of the servers is accounted on local computers. On the other hand the complete computing effort is also conceivable on the servers. Computing effort would in this case be accounted in addition, while clients create only local output. This expansion for a dynamic invoice of services would be a comfortable and fast alternative.

Conclusions/Summary

The experiences won with the previous implementation work confirm the selection of Java as integration language for a distributed traffic simulation environment. The development environments have to be improved strongly in our opinion provided that Java wants to get just to its claim as a design language for distributed Intranetapplications and shall not merely serve for the animation of the WWW pages to the construction of simple Applets. Performance and uniformity of the Java-interpreters also are worthy improvements — there are internally clear differences between the Java-interpreters on the various platforms. Sun has been announced to clear this inconsistencies in the Java-Interpreter with publication of the version 1.1. The question of the performance can be invalid by a quick distribution or even integration of Java processors.

Summarizing one can say that today's stand of the Java-language and hers available libraries allows th proportionally simply development of distributed applications over system platforms compared to other programming languages.

Literatur

Jenni95 Jenni+Gottardi AG: SIAM - Straßenverkehrsinformationssystem mit einem adaptiven Modell, Zürich, 1995

Leutzbach88 W.Leutzbach, S.Schnittger: Erweiterung eines bestehenden Modells um ein Modul zur Erfassung der Reisezeit, des Energieverbrauchs und der Abgasemissionen, Schriftenreihe Straßenbau und Straßenverkehrstechnik, Heft 586, Bundesministerium für Verkehr, 1988

Mock94 R. Mock-Hecker: Wissensbasierte Erkennung kritischer Verkehrssituationen - Erkennung von Plankonflikten, VDI-Verlag, Reihe 12 Verkehrstechnik/Fahrzeugtechnik, Band-Nr. 209, 1994

Schnittger91 S. Schnittger: Einfluß von Sicherheitsanforderungen auf die Leistungsfähigkeit von Schnellstraßen, Schriftenreihe des Instituts für Verkehrswesen, Universität (TH) Karlsruhe, 1991

Zeller96 M. Zeller: Planerkennung im Straßenverkehr, VDI-Verlag, Reihe 12 Verkehrstechnik/Fahrzeugtechnik, Band-Nr. 282, 1996

JavaLang95 The Java Language Specification, Version 1.0 Beta, October 1995

JavaVirt95 The Java Virtual Machine Specification, August 1995

JavaBeans96 Java Beans 1.0 API-Specification, October 1996

JavaRMI96 Java RMI Specification Overview, Rev. 0.9, May 1996

NEW WIDE-BAND PROPAGATION CHANNEL MODEL FOR THE MM-WAVE BAND

José Fernandes and José Neves

Universidade de Aveiro, Instituto de Telecomunicações, 3810 Aveiro - Portugal
Phone: +351 34 383090, Fax: +351 34 383091, Email: zf@ua.pt

Abstract. A new wide-band propagation channel model for the millimetre-wave band, based on high frequency ray theory approximation, is presented in this paper. This model consists of a set of simple equations derived from the propagation theory and is able to take into account the propagation environment characteristics as well as the location of the transceiver antennas and their radiation patterns. The results obtained with the developed model agree quite well with the results obtained with a ray tracing tool in a relatively wide range of environments. Also a comparison with some measurements have shown the validity of the proposed model.

Introduction

The Mobile Broadband System (MBS) is currently under development in Europe, aiming at offering to the mobile users an (Asynchronous Transfer Mode) ATM based radio access to the future Broadband Integrated Services Digital Network (B-ISDN)[1].

The specification of the radio interface for the MBS system represents a considerable challenge. The wide range of services to be offered, with great variety of characteristics and requirements, including service bit rates over 100 Mbit/s, are clearly beyond the capabilities of the existing radio mobile systems. This will lead to the use of the millimetre-wave frequencies for the radio link due to the high bandwidth required to transmit such high bit rates. As the propagation channel strongly influences the system performance[2], it is required to use channel models able to take into account the site-specific propagation characteristics as well as the antennas of the mobile station (MS) and of the base station (BS).

This paper presents a propagation model for indoor environments and easily extended to the outdoor environments. The model allows to estimate the channel impulse response (IR) in a given environment taking into account the geometry of the scenario, reflection properties, positions of the MS and the BS and their antennas radiation patterns. Once the IR is obtained, it is straightforward to calculate the time dispersion parameters, such as the delay spread (DS) and delay window (DW), and the normalised received power (NRP) for each MS position.

Impulse response modelling

The impulse response of a multipath propagation channel[3] is represented by multiple paths or rays having real positive gains, propagation delays and associated phase shifts. It can be written as a combination of line-of-sight (LOS) ray and multiple order reflected rays:

$$h(\tau)=\alpha_{LOS}e^{-j\varphi_{LOS}}\delta(\tau-\tau_{LOS})+\sum_{m=1}^{M}\sum_{n=1}^{Nm}\alpha_{mn}e^{-j\varphi_{mn}}\delta(\tau-\tau_{mn}) \qquad (1)$$

where $\delta(.)$ is the Dirac delta function; α_{mn} represents the normalised amplitude (normalised to the transmitted pulse amplitude) of the nth ray reaching the receiver after m reflections; τ_{mn} is the excess time delay given by r_{mn}/c, where r_{mn} is the length of nth ray reflected m times and c represents the light speed in free space; φ_{mn} accounts for the phase shift. Correspondingly, α_{LOS}, τ_{LOS} and φ_{LOS} represent the amplitude, excess delay and phase shift of the LOS-ray. M is the maximum reflection order to be considered.

The NRP, defined as the quotient of the received to the transmitted power, can be written as

$$NRP=\frac{P_r}{P_t}=\alpha^2_{LOS}+\sum_{m=1}^{M}\sum_{n=1}^{Nm}\alpha^2_{mn} \qquad (2)$$

and the DS defined in [3] can be obtained as follows:

$$DS=\sqrt{\overline{\tau^2}-(\overline{\tau})^2} \qquad (3)$$

where
$$\overline{\tau^k}=\left(\tau^k_{LOS}\alpha^2_{LOS}+\sum_{m=1}^{M}\sum_{n=1}^{Nm}\tau^k_{mn}\alpha^2_{mn}\right)/NRP,\quad k=1,2. \qquad (4)$$

As excess time delay is proportional to the ray-path length, $h(\tau)$ can be referred as $h(r)$ and can be seen as the IR in the distance domain:

$$h(r) = \alpha_{LOS} e^{-j\beta r_{LOS}} \delta(r - r_{LOS}) + \sum_{m=1}^{M} \sum_{n=1}^{Nm} \alpha_{mn} e^{-j(\varphi_{Rmn} + \beta r_{mn})} \delta(r - r_{mn}) \tag{5}$$

where α_{mn} is calculated as:

$$\alpha_{mn} = C \left| \frac{\vec{E}(\theta_{Tmn}, \phi_{Tmn})}{r_{mn}} \prod_{k=1}^{m} R(\varepsilon_{r_k}, \gamma_{ik}, \psi_k) \cdot \bar{l}_e(\theta_{Rmn}, \phi_{Rmn}) \right| \tag{6}$$

$\vec{E}(\theta,\phi)$ represents the electric field vector of the transmitting antenna at ray departure direction and $\vec{l}_e(\theta,\phi)$ the effective length of the receiving antenna; θ and ϕ are directional angles defined as in spherical co-ordinates where the subscript T stands for transmitting and R for receiving ray direction and β is the wave number. Eq. (6) can be seen as Friis formula applied to each ray including the reflection losses. Neglecting the losses introduced by the antennas, the constant C $(=\lambda/4\pi)$ can be obtained through the Friis formula for free space propagation. $R(\varepsilon_r, \gamma_i, \psi)$ represents the reflection coefficient which depends on the dielectric permitivity ε_r, the angle of incidence γ_i, and polarisation angle ψ. φ_{Rmn} accounts for the phase shift at each reflection point and the number of rays Nm, for a reflection order m, is given by $4m^2 + 2$ [4]. α_{LOS} can be obtained using (6) for $m = 0$.

Using the relation $\tau_i = r_i/c = 10r_i/3$ (ns), time domain can be changed to distance domain, thus by simple mathematical manipulation it follows from (3) that DS(ns) = 10/3 DS(m), as well as for the other time dispersion parameters. The estimation of the IR requires the calculation of each ray-path length r_{mn}, the respective departure and arriving directions (θ, ϕ) to the transceiver antennas and the reflection coefficients $R(\varepsilon_{rk}, \gamma_{ik}, \psi_k)$. These parameters calculations are based on the geometrical optics and on the image theory.

Image theory

The Fig. 1 illustrates the application of the image theory (2-D) to trace two rays of second reflection order. The thicker rectangle represents the "real" room and the others represent the image rooms to obtain the image points of the BS, P'_B. For a second order reflected ray it is necessary to calculate two images to obtain the required image point. For the ray represented in Fig. 1, reflected in the walls $y = 0$ e $y = W$, the co-ordinates of the points $P'^{(1)}_{B1}$ e $P'^{(2)}_{B1}$ are given by:

$$P'^{(1)}_{B1} = \begin{cases} x'_B = x_B \\ y'_B = W + W - y_B = 2W - y_B \end{cases} \qquad P'^{(2)}_{B1} = \begin{cases} x'_B = x_B \\ y'_B = -(W + W - y_B) = -2W + y_B \end{cases} \tag{7}$$

For the first order reflected rays, it is enough to calculate the image of the transceiver relatively to the plane where reflection occurs, and as shown in Fig. 1 and the second order reflected rays can be obtained from first ones. The successive application of this theory allows to calculate the image points of a transceiver (MS or BS) enabling the calculation of all possible reflected rays of each reflection order. For a m times reflected ray it is necessary to calculate m image points, e.g., the image point $P'^{(k)}$ is obtained as the image of $(P'^{(k-1)})$. The process is identical in 3-D and each point is then represented by the usual (x, y, z) co-ordinates.

Once the image points are known, the length, the departure and the arriving angles of each ray are obtained by simple 3-D vectorial analysis. Note that the length of the second order ray measured from P_B to P_M is equal to the distance between $P'^{(2)}_B$ and P_M, then the length of each ray emerges from the 3-D distance calculation between the MS position and the m-order image point of the BS, as shown in Fig. 1 (in 2-D), then

$$r_{mn} = \sqrt{(x'^{(m)}_B - x_M)^2 + (y'^{(m)}_B - y_M)^2 + (z'^{(m)}_B - z_M)^2} \tag{8}$$

Also using simple vectorial analysis, the departure angle of each ray from the MS in the vertical plane, θ_{MS}, is given by:

$$\theta_{MS} = \frac{\pi}{2} - arctg\left(\frac{\left| z'^{(m)}_B - z_M \right|}{\sqrt{(x'^{(m)}_B - x_M)^2 + (y'^{(m)}_B - y_M)^2}} \right) \frac{z'^{(m)}_B}{\left| z'^{(m)}_B \right|} \tag{9}$$

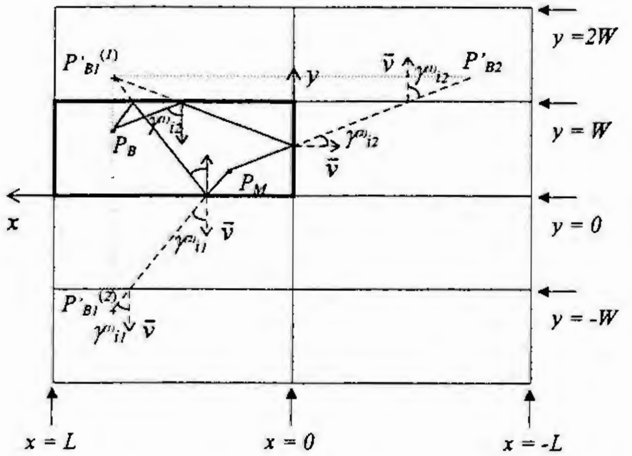


Fig. 1 Example of two second order rays in 2-D traced using the image theory, including also the incidence angles

The second term of (9) defines θ_{MS} from xy plane clock-wise, and $\pi/2$ converts θ_{MS} to the standard spherical co-ordinates. The arriving angle to the BS, θ_{BS}, is obtained from (9) by interchanging P_B with P_M. The departure angle of the MS in the horizontal plane, ϕ_{MS}, in the four quadrants, is given by (10) and (11); and similarly, the arriving angle to the BS, ϕ_{BS}, is obtained from (10) and (11) by interchanging P_B with P_M..

$$\phi_{MS} = arctg\left(\frac{\Delta y}{\Delta x}\right) + \frac{\Delta y}{|\Delta y|}\left(1 - \frac{\Delta x}{|\Delta x|}\right)\frac{\pi}{2} \qquad (10)$$

with

$$\Delta y = y'^{(m)}_B - y_M \quad \text{and} \quad \Delta x = x'^{(m)}_B - x_M \qquad (11)$$

Calculation of the image points

A recursive algorithm to generate the co-ordinates of the image points $P'_B = (x'_B, y'_B, z'_B)$ or $P'_M = (x'_M, y'_M, z'_M)$ (for simplicity the "exponential" m will not be used from now) will be established for all ray-paths up to the maximum reflection order M. Based on the image theory, Snell reflection law and using a ray-tracing tool described in [5], tailored "maps" to represent the Cartesian co-ordinates of the image points were built for each reflection order.

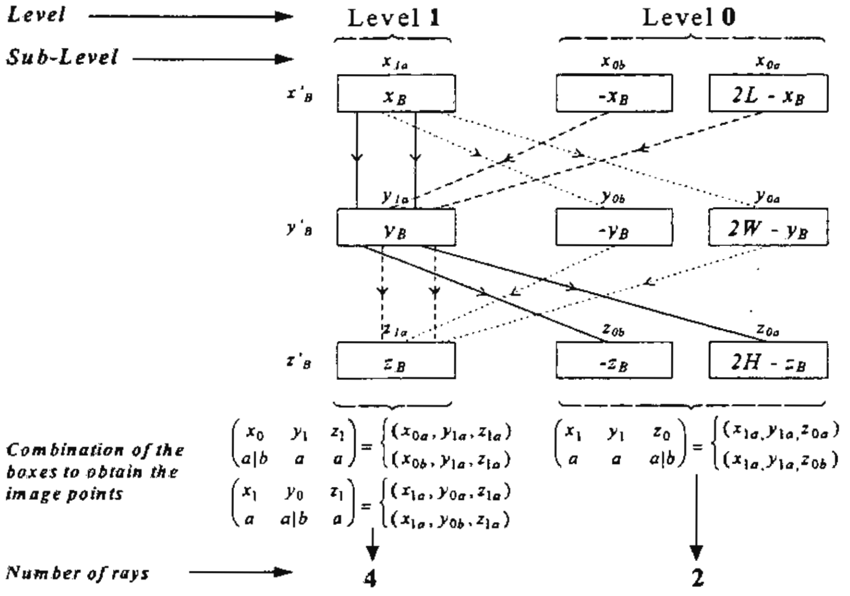
The Fig. 2 represents the six image points of the first order reflected rays, which can be drawn in Fig. 1. The level in the "map" varies from zero up to the considered reflection order (m + 1 levels). Each level has two sub-levels named a and b, with the exception of the last one (level m), that has only the level a. Each trajectory in the "map", linking three boxes (rectangles representing the co-ordinates), gives one triplet (x'_B, y'_B, z'_B) that represents an image point. The six trajectories defined by the lines, show the necessary combinations to obtain the six image points of the first order reflected rays. For example, to determine the co-ordinates of the image point relatively to the combination (x_{1a}, y_{0a}, z_{1a}), we just need follow in the "map" the trajectory starting from sub-level $1a$ of the co-ordinate x'_B ($x_{1a} = x_B$), go through level 0, sub-level a, of the co-ordinate y'_B ($y_{0a} = 2W - y_B$) and stop at z'_B ($z_{1a} = z_B$). Similarly, the Fig. 3 represents the "map" for second order reflect rays, where we skip, for legibility reasons, the trajectories corresponding to level 1 of the z co-ordinate. The thicker break line corresponds to the second order reflected mentioned above with co-ordinates $(x_B, -2W + y_B, z_B)$, which can easily be checked in the Fig. 1. The first order reflected rays are the easiest to obtain, because only one reflection point is needed and then only one image point per ray.

Analysing carefully the Figs. 2 and 3, we conclude that the "map" of the second reflection order can be obtained from the first order. Similarly, the "map" of the third order can be obtained from the second order, and so on. To construct a m-order "map", one has to add to the "map" of order m - 1 one level on the right hand side (thicker boxes in the Fig. 3), being this one the level 0 (zero). The levels of the order m - 1 are increased by one unit, being this rule valid for any order.

From the "maps" of each reflection order, we developed a recursive formula to calculate the content of each box, starting at level m up to level zero. As in each column (sub-level) the equations for x'_B, y'_B and z'_B are identical, the same equation can be used for all co-ordinates just by assigning the respective variables.

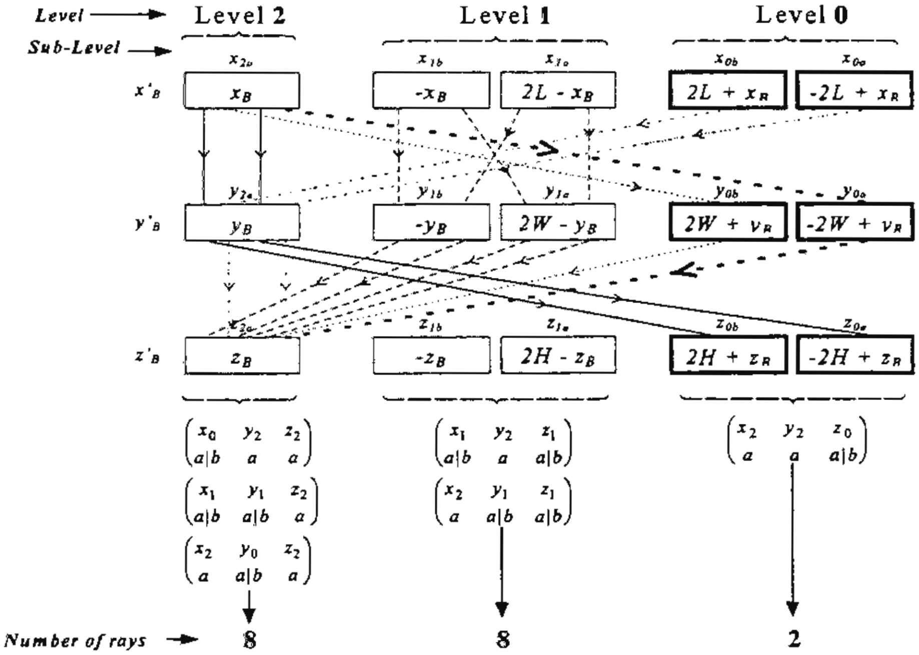
Looking to the Fig. 3, we conclude that the contents of the boxes in the column b (sub-level b) of level 1 is the symmetric of the boxes in the level 2. The contents of the boxes in the column a (sub-level a) of level 1 is obtained by adding $2L$ (in case of x co-ordinate, $2W$ for y and $2H$ for z) to the symmetric of the boxes in the level 2. Correspondingly, the contents of the boxes in the column b of level 0 is obtained by adding $2L$ (or $2W$ or $2H$) to the symmetric of column b of level 1 and the contents of column a, level 0 is the symmetric of the column a of level 1. This calculation rule is valid for any reflection order and is illustrated in Fig. 4.

Now, in order to obtain the reflection points, it is necessary to define how to combine the boxes represented in the "maps". The bottom part of Fig. 2 shows the matrix representation of the box combination for the first order reflected rays and the corresponding triplets to obtain the co-ordinates. The first row of a matrix indicates the co-ordinate x', y' and z' being the level indicated by the index of the co-ordinate. The second row, indicates the sub-levels of each co-ordinate, which can be a or $(a|b)$, where $(a|b)$ means that the co-ordinate has two sub-



Level
Sub-Level
Level 1
Level 0
x_{1a}
x_{0b}
x_{0a}
x'_B
x_B
$-x_B$
$2L - x_B$
y_{1a}
y_{0b}
y_{0a}
y'_B
y_B
$-y_B$
$2W - y_B$
z_{1a}
z_{0b}
z_{0a}
z'_B
z_B
$-z_B$
$2H - z_B$
Combination of the boxes to obtain the image points
$\begin{pmatrix} x_0 & y_1 & z_1 \\ a|b & a & a \end{pmatrix} = \begin{cases} (x_{0a}, y_{1a}, z_{1a}) \\ (x_{0b}, y_{1a}, z_{1a}) \end{cases}$
$\begin{pmatrix} x_1 & y_0 & z_1 \\ a & a|b & a \end{pmatrix} = \begin{cases} (x_{1a}, y_{0a}, z_{1a}) \\ (x_{1a}, y_{0b}, z_{1a}) \end{cases}$
$\begin{pmatrix} x_1 & y_1 & z_0 \\ a & a & a|b \end{pmatrix} = \begin{cases} (x_{1a}, y_{1a}, z_{0a}) \\ (x_{1a}, y_{1a}, z_{0b}) \end{cases}$
Number of rays
4
2

Fig. 2 "Map" for the first reflection order to obtain the image points



Level
Sub-Level
Level 2
Level 1
Level 0
x_{2a}
x_{1b}
x_{1a}
x_{0b}
x_{0a}
x'_B
x_B
$-x_B$
$2L - x_B$
$2L + x_R$
$-2L + x_R$
y_{2a}
y_{1b}
y_{1a}
y_{0b}
y_{0a}
y'_B
y_B
$-y_B$
$2W - y_B$
$2W + y_R$
$-2W + y_R$
z_{2a}
z_{1b}
z_{1a}
z_{0b}
z_{0a}
z'_B
z_B
$-z_B$
$2H - z_B$
$2H + z_R$
$-2H + z_R$
$\begin{pmatrix} x_0 & y_2 & z_2 \\ a|b & a & a \end{pmatrix}$
$\begin{pmatrix} x_1 & y_1 & z_2 \\ a|b & a|b & a \end{pmatrix}$
$\begin{pmatrix} x_2 & y_0 & z_2 \\ a & a|b & a \end{pmatrix}$
$\begin{pmatrix} x_1 & y_2 & z_1 \\ a|b & a & a|b \end{pmatrix}$
$\begin{pmatrix} x_2 & y_1 & z_1 \\ a & a|b & a|b \end{pmatrix}$
$\begin{pmatrix} x_2 & y_2 & z_0 \\ a & a & a|b \end{pmatrix}$
Number of rays
8
8
2

Fig. 3 "Map" for the second reflection order to obtain the image points. For legibility reasons, the trajectories corresponding to the *z* co-ordinate of the level *1* were not drawn.

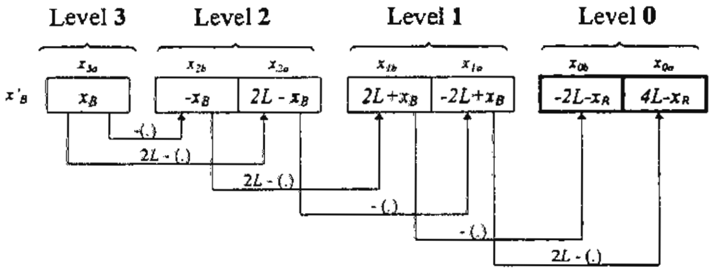


Fig. 4 Algorithm to calculate the contents of the boxes. (.) means contents of the box where the arrow starts.

levels a and b. The Fig. 3 shows the matrix representation for the second order, and the points can obtained as in the example in Fig. 2. In practice three boxes for each reflection point are obtained. Since its contents is already known, it is straightforward to calculate the image points.

Observing the "maps" in Figs. 2 and 3, the m-order box combination can be obtained from the order m - 1, as it is for the "maps" itself, and then we can derive the box combinations starting with first order shown in Fig. 2. Thus, the box combination for the level m are obtained by adding the sub-level b to the z co-ordinate of the order m - 1, e.g., changes from a to $a|b$, the levels of the combinations that come out from this operation, as well as the other ones, are increment by one unit except for the z' co-ordinate. The combinations of the former order allocated to the levels 0, .., m, are now allocated to the 0, ..., m - 1 levels of the new reflection order, being the ones relatively to the level m given by (12). This method is quite efficient because it uses the order m - 1 to obtain the order m by just using (12) to generate the level m for the order m.

$$\begin{pmatrix} x_0 & y_m & z_m \\ a|b & a & a \end{pmatrix}$$
$$\begin{pmatrix} x_m & y_0 & z_m \\ a & a|b & a \end{pmatrix} \tag{12}$$
$$\begin{pmatrix} x_p & y_{m-p} & z_m \\ a|b & a|b & a \end{pmatrix} ,p = 1,2,m-1$$

Once we know the image points, (8) can be used to obtain the ray-path length, (9) - (11) to calculate the arriving and the departure angles enabling the use of any antenna radiation pattern at the BS and the MS. Looking at eq. (6), we need now to calculate the reflection coefficient to obtain the IR written in(5).

Reflection coefficient

The reflection coefficient of a surface depends on the ε_r, $tg\delta$, γ_i, and ψ, being the polarisation described by two orthogonal components, one parallel and other perpendicular to the incidence plane. Thus, the amplitude of the reflected field can be expressed as [6]:

$$\left|\bar{E}^r(\varepsilon_r,tg\delta,\gamma_i,\psi)\right| = \sqrt{\left|\left\|\bar{E}^i\right|R_\perp(\varepsilon_r,tg\delta,\gamma_i)cos(\psi)\right|^2 + \left|\left\|\bar{E}^i\right|R_\parallel(\varepsilon_r,tg\delta,\gamma_i)sin(\psi)\right|^2} \tag{13}$$

In order to simplify the model, we skip the reflection coefficient dependence with polarisation angle ψ, taking an average value in the variation domain, being the average power reflection coefficient given by (14). Since ε_r and $tg\delta$ are the electromagnetic characteristics of a particular surface, we need to calculate the incidence angle γ_i.

$$\left|R(\varepsilon_r,tg\delta,\gamma_i)\right|^2 = \frac{1}{\pi}\int_0^\pi \left|R(\varepsilon_r,tg\delta,\gamma_i,\psi)\right|^2 d\psi = \frac{1}{2}\left[\left|R_\perp(\varepsilon_r,tg\delta,\gamma_i)\right|^2 + \left|R_\parallel(\varepsilon_r,tg\delta,\gamma_i)\right|^2\right] \tag{14}$$

Calculation of incidence angle

Using the definition of the internal product between two vectors, the incidence angle can be obtained for each ray at each reflection point as shown in Fig. 1. Defining the vector $\vec{u}$ by the points P_M and P'_B, and vector $\vec{v}$ by the unit vector normal at each reflection plane: $\vec{v}$ = (1, 0, 0) for $x = kL$ planes, $\vec{v}$ = (0, 1, 0) for $y = kW$ planes and $\vec{v}$ = (0, 0, 1) for $z = kH$ (ceiling and floor), with k being an integer number, the incidence angle is given by(15).

$$\gamma_i = arccos\left(\frac{\vec{u}.\vec{v}}{\|\vec{u}\|\|\vec{v}\|}\right) \tag{15}$$

The problem now is to identify the planes where a ray is successively reflected. From the box combination that originates an image point p'_B , we can derive those planes and then the respective unit vectors. The co-ordinates of the points represented in Fig. 1 are $P'_{B1} = (x_B, -2W + y_B, z_B)$ e $P'_{B2} = (-x_B, 2W - y_B, z_B)$. The ray associated to P'_{B1} is reflected at the planes $y = 0$ and $y = W$, being the box combination represented in Fig. 3 by the thicker lines (x_{2a}, y_{0a}, z_{2a}). This ray is reflected twice at $y = kW$, that corresponds to the difference between the reflection order and the box level of the co-ordinate y', y_{0a} belongs to level 0 and then 2 - 0 = 2. There are no reflections in the other planes. The box combination corresponding to P'_{B2} is (x_{1b}, y_{1a}, z_{2a}), which results in a reflection at $x = kL$ and other at $y = kW$. Therefore, the number of reflections per ray in one co-ordinate is equal to the reflection order minus the corresponding level of the box of that co-ordinate, but we still don't know how to distinguish a plane $x = 0$ or $x = L$. This information can be obtained from the sub-level of the box. If a ray is reflected p times in one co-ordinate, the sequence of the reflection planes starts at $C = D$ (C: co-ordinate x, y, z, D: dimension L, W, H) if the box belongs to sub-level a, interchanging successively with $C = 0$; if a box belongs to sub-level b the sequence starts at $C = 0$. As an example the ray corresponding to P'_{B2} is reflected at $x = 0$ and at $y = W$, and the other at $y = 0$ and $y = W$. Like this we can identify each plane of the scenario where a ray is reflected, enabling the use of proper reflection properties.

Results and comparison with measurements

Due to the limited space available only very few results are presented to validate the developed model. The Figs. 5 and 6 demonstrate that the results obtained with the model agree quite well with a ray tracing tool (sim. in figures legend) and with experimental measurements obtained in the rooms C and H described in [5], showing the validity of the model. The results depicted in Fig. 5 were obtained with a pair of biconic horn antennas (Bic-Bic), while for the ones in Fig. 6, the BS uses a horn antenna located in a room corner and the MS uses a biconic antenna (Horn-Bic). This shows that the model is able to take into account antennas with directivity in both planes, vertical and horizontal. The simulated and measured results depicted in Figs. 5 and 6 were obtained with 1 ns time resolution, while the ones from the model have infinite resolution, which explains some differences between them due to the fast fading.

More extensive validation of the model in [6], demonstrates that the results of the model agree quite well with a ray tracing tool in several environments considering different polarisation states and different locations of the BS as predicted by the model.

Conclusions

A wide-band propagation channel model able to take into account the geometry of the scenario, its reflection properties, the positions of MS and of the BS as well as the radiation patterns of the transceiver antennas was presented. This model was developed for indoor environments, but it is easily extended to outdoors. In order to simplify the model, an average value of the of the reflection coefficient relatively to the polarisation angle has been considered. However, the results of the model agree quite well with the results obtained with a ray tracing tool in several environments considering different polarisation states. Also a comparison with experimental results have shown the validity of the proposed model

Due to the flexibility and the implementation simplicity, this model will be of great utility in the estimation of channel IR in a given environment and its characteristic parameters, such as PRN, DS, DW, etc. These parameters are very useful in the practical implementation aspects of the system, allowing the estimation of the system transmission capacity based on time dispersion parameters [6]. On the other hand, the power delay profile of the channel is a key issue for system simulation because it allows to study the system behaviour in different scenarios and with different antennas set-up. Based on power delay profile, a tap-delay-line model can be established and an adequate statistical amplitude and phase distribution for each time bin to emulate the fast fading can be included.

In our opinion, this model is a contribution for the radio propagation channel modelling in millimetre-wave band, and it will enable the system researchers to use it to represent realistic scenarios.

References

1. L. Fernandes, "*Developing a System Concept an Technologies for Mobile Broadband Communications*", IEEE Personal Communications Magazine, February 1995.
2. J. Fernandes, J. Nascimento, A. Gusmão, R. Dinis and J. Neves, "*Performance Evaluation of Mm-wave Wide-Band Digital Radio Transmission*", IEEE 2nd Symposium on Communications and Vehicular technology in the Benelux, Nov. 1994.
3. J.D. Parsons, "*The Mobile Radio propagation Channel*", Prentech Press, London, 1992.
4. U. Dersch and E. Zollinger, "*Propagation Mechanisms in Micro Cell and Indoor Environments*", Proc. Joint COST 227/231 Workshop on Mobile Comm., University of Limerick, pp.191-213, Sept. 1993.

5. J. Fernandes, P. Smulders and J. Neves, *"Mm-wave Indoor Channel Vs. Measurements"*, Wireless Personal Communications Journal, Vol. 1, N° 3, pp. 211-219, 1995

6. J. Fernandes, " *Modelização do Canal de Propagação Rádio Móvel de Banda Larga na Faixa das Ondas Milimétricas e seu Impacto no Desempenho de Transmissão do Sistema*", PhD Thesis, Universidade de Aveiro - DETUA (in Portuguese). Available in the beginning of 1997.

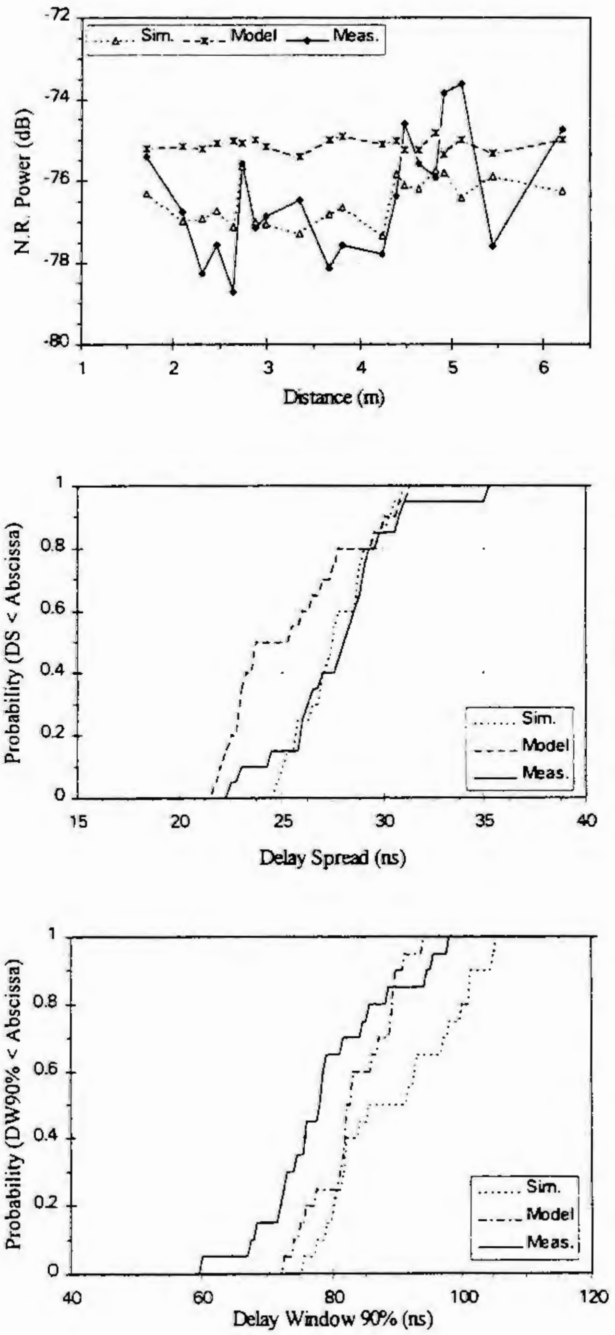


Fig. 5 PRN, DS and DW90% obtained with the ray tracing tool, with the model and experimental measurements in Room H.

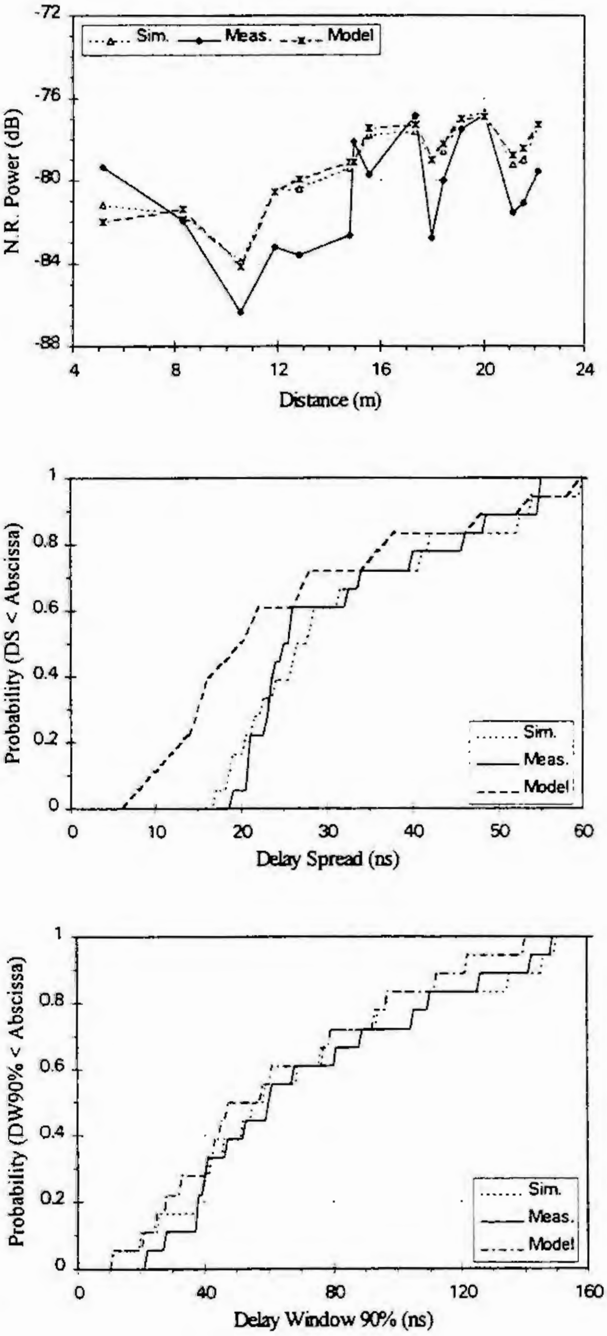


Fig. 6 PRN, DS and DW90% obtained with the ray tracing tool, with the model and experimental measurements in Room C.

Fast Collision Resolution in Wireless ATM Networks*

Dietmar Petras, Andreas Krämling

Communication Networks, Aachen Univ. of Technology
E-Mail: {petras|akr}@comnets.rwth-aachen.de
WWW: http://www.comnets.rwth-aachen.de/~petras

Abstract — **The paper models the medium access control (MAC) layer of a wireless ATM network as a distributed queueing system. A random access channel with short slots is used for the transmission of capacity requests from distributed queues in the wireless terminals to the central scheduler in the base station. The paper describes the mathematical analysis of a fast collision resolution algorithm which is based on conventional splitting algorithms but employs identifiers of terminals to choose a subset after a collision. Based on the analytical results, a new medium access control protocol for the random access channel is defined which is called *probing algorithm*. Its performance is evaluated by stochastic simulations.**

1 Introduction

Future broadband multimedia telecommunication networks according to the I-300-series of the ITU-T recommendations are based on a packet switching technique established in 1990/91, the so-called asynchronous transfer mode (ATM).

In this paper we analyse a model of the medium access control (MAC) layer of a wireless (W) ATM network. The MAC layer is characterized by the realization of a distributed queueing system in Fig. 1 as described in [4, 5]. The scheduler of the distributed queueing system is located in the central base station. The buffers with packets (so-called ATM cells) waiting for transmission over the radio link from the wireless terminals to the base station are located in the terminals.

A difficult task of the MAC protocol is the transmission of the queue status (so-called capacity request) from the terminals to the scheduler in the base station, where it is required for the correct execution of the serving strategy of the scheduler.

Usual MAC protocols for W-ATM networks are using frames of variable length (so-called periods) with slots for the transmission of ATM cells and shorter slots for the transmission of capacity requests. At the beginning of each period the assignment of slots of the period to terminals is broadcasted by the base station. The number of short slots in a period can be chosen from 0 to n with realistic $n < 50$. The sequence of short slots is called random access channel (RACH). Polling means, that the base station invites a specific terminal to transmit in a reserved slot. Random access happens, if a group of or all terminals are allowed to transmit in a slot. The result of a random access (ternary feedback: free, successful, collision) is broadcasted by the base station at the end of each period over a feedback channel. An error-free feedback is assumed. In the conclusions of the paper we discuss the effect of faulty feedbacks. If a collision occurs, all collided packets are lost and have to be retransmitted[1]. A collision resolution algorithm is necessary to guaranty stability and limited delays [1].

The literature describes splitting algorithms as the collision resolution algorithms with highest throughput [2, 3]. In this family of algorithms terminals are grouped to sets. All terminals of a set are allowed to transmit in a specific slot. A transmission will only be successful, if a set contains exact one terminal. After a collision the set is split into several subsets according to the order of the collision resolution algorithm (two subsets with binary algorithms, three subsets with ternary algorithms, etc). A collided terminal chooses its followup subset by using a certain strat-

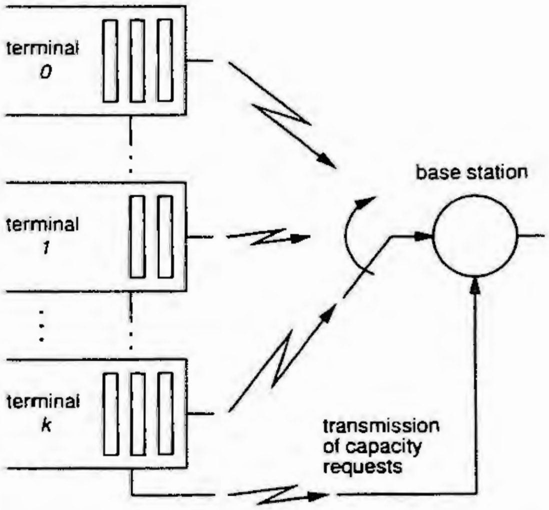


Figure 1: Modelling the MAC layer of a W-ATM network as a distributed queueing system

*Supported by: The German Federal Ministry of Education, Science, Research and Technology

[1] Capture may enable the reception of the packet with the highest signal strength even if a collision occurred. This effect is neglected.

egy (eg. by an unbiased random experiment, so-called coin flipping). If no collision occurs in a subset, the collision is resolved, otherwise the subset is split again. In blocking algorithms, a collision resolution phase is started with a start set. Terminals with new arrivals are not allowed to access the subsets of an ongoing phase. They have to wait for the next start set founding a new phase. Unblocked algorithms allow new arrivals to enter the current phase directly. It has been shown that this decreases the performance slightly but reduces implementation effort [3]. The performance of collision resolution algorithms is given by throughput (radio of slots with successful transmission to all slots) and delays.

Due to the support of realtime oriented multimedia services in ATM networks, the collision resolution algorithm in a W-ATM network is less to be optimized to throughput but to short delays. Furthermore, the maximum delay is an important performance parameter.

2 Model of the RACH

We introduce the following model of the RACH: We assume that a period of duration τ_P is able to offer any number of RACH slots. In a terminal the need for transmitting capacity requests is modelled by the arrival of a packet. We assume the probability p of at least one new arrival in a terminal during one period τ_P. New arrivals can only occur at terminals with no waiting packet. At the start of each period the base station determines the assignment of slots to groups of terminals. At the end of each period the error-free feedbacks are broadcasted.

This model differs from that of other systems in following items:

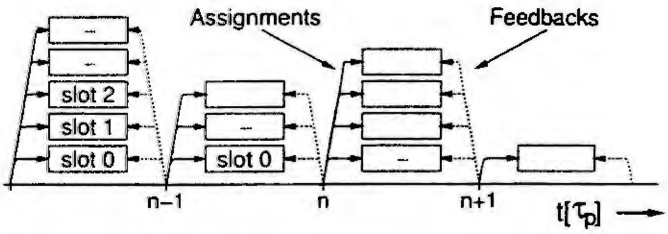


Figure 2: Model of the RACH for transmission of capacity requests

- Limited and known number of terminals, since terminals have to register before transmitting capacity requests. The terminals are numbered by consecutive identifiers from a identifier space $[0, \ldots, o^n - 1]$ with o being the order of the space and n being its dimension.
- Unlimited number of simultaneous slots per period
- Delayed feedback at the end of each period
- Delays are measures as multiples of the period duration τ_P.

3 Analysis of the Identifier Splitting Algorithm

Since the number of terminals is limited and known, it is useful to distribute collided terminals on the followup subsets according to their identifiers leading to the *identifier splitting algorithm*. With each splitting step the dimension n of the remaining identifier space decreases by one.

For the binary ($o = 2$) identifier splitting algorithm the number of terminals in the resulting subsets is a hypergeometric random variable. In case of a collision ($k \geq 2$) the probability of k_l terminals choosing the left subset and the remaining $k_r = k - k_l$ terminals choosing the right subset is:

$$P_{n,k}(k_l) = \frac{\binom{2^{n-1}}{k_l}\binom{2^{n-1}}{k-k_l}}{\binom{2^n}{k}} \tag{1}$$

We analyse the throughput of the identifier splitting algorithm by determining the number of slots $N_{o,n}(k)$ for the resolution of a start set of k terminals (splitting order o, dimension n of identifier space). The recursions (2) for the binary and (3) for the ternary algorithm are used with the starting condition $N_{o,n}(0) = N_{o,n}(1) = 1$.

$$\begin{aligned} N_{2,n}(k) &= 1 + \sum_{i=\max(0,k-2^{n-1})}^{\min(k,2^{n-1})} \frac{\binom{2^{n-1}}{i}\binom{2^{n-1}}{k-i}}{\binom{2^n}{k}} \Big(N_{2,n-1}(i) + N_{2,n-1}(k-i) \Big) \\ &= 1 + \frac{2}{\binom{2^n}{k}} \sum_{i=\max(0,k-2^{n-1})}^{\min(k,2^{n-1})} \binom{2^{n-1}}{i} \binom{2^{n-1}}{k-i} N_{2,n-1}(i) \end{aligned} \tag{2}$$

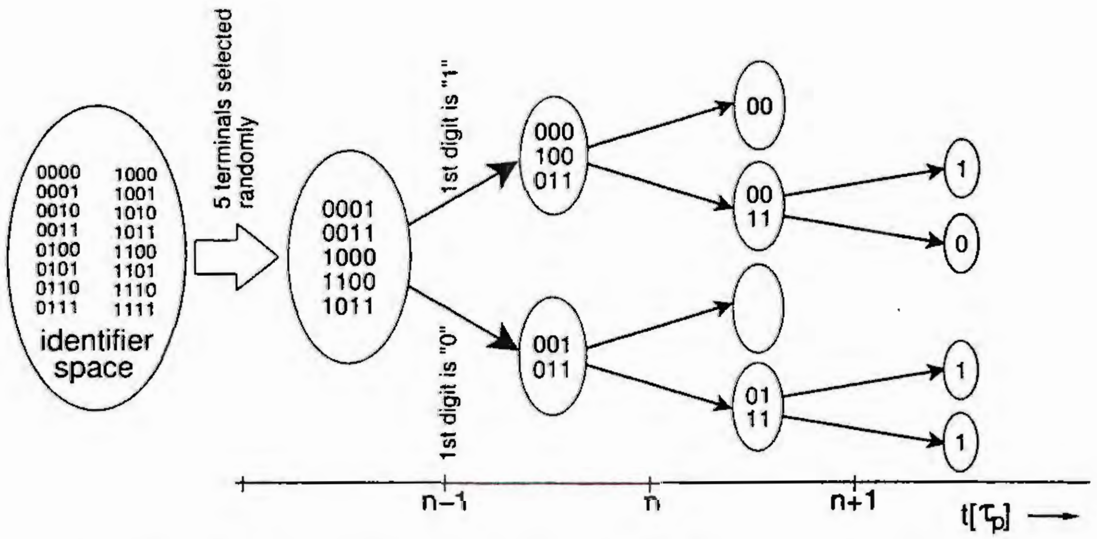


Figure 3: Example of binary ($o = 2$) identifier splitting with identifier space of dimension $n = 4$

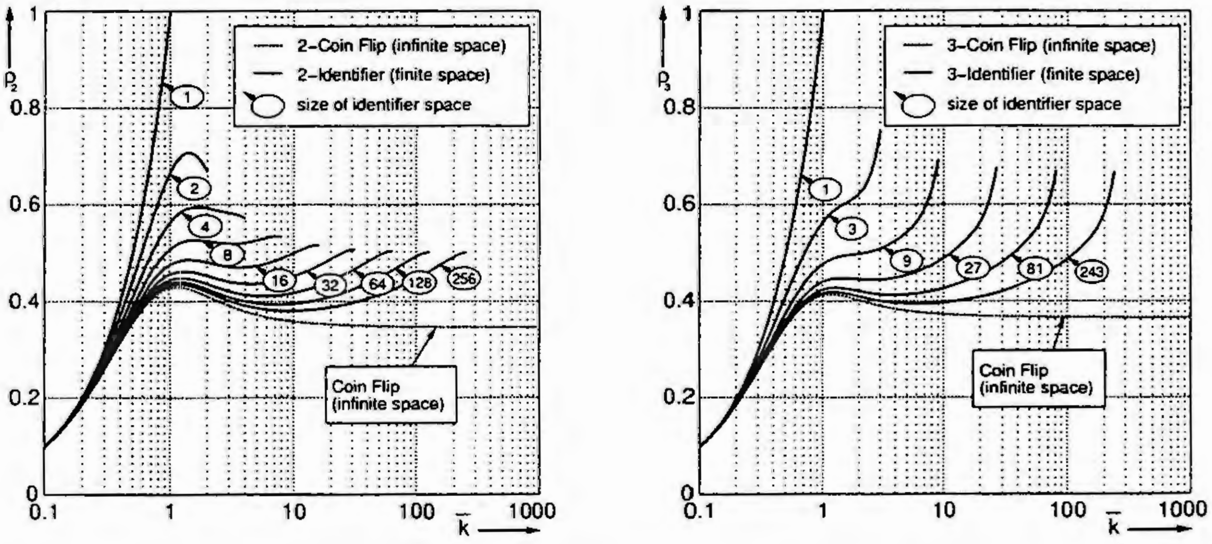


Figure 4: Throughput $\rho_{o,n}(\overline{k})$ of binary ($o = 2$) and ternary ($o = 3$) identifier splitting algorithm with binomial-distributed number of terminals in a start set

$$N_{3,n}(k) = 1 + \sum_{i=0}^{\min(k,3^{n-1})} \sum_{j=\max(0,k-i-3^{n-1})}^{\min(k-i,3^{n-1})} \frac{\binom{3^{n-1}}{i}\binom{3^{n-1}}{j}\binom{3^{n-1}}{k-i-j}}{\binom{3^n}{k}} \Big(\cdots$$
$$\cdots N_{3,n-1}(i) + N_{3,n-1}(j) + N_{3,n-1}(k-i-j)\Big) \quad (3)$$

The size k of a start set of a collision resolution phase is a binomial random variable with $0 \leq k \leq o^n$ and mean $\overline{k}$. Thus, the throughput $\rho_{o,n}(\overline{k})$ is calculated by (4).

$$\rho_{o,n}(\overline{k}) = \frac{\overline{k}}{\sum_{k=0}^{o^n} N_{o,n}(k) \cdot \binom{o^n}{k} \left(\frac{\overline{k}}{o^n}\right)^k \left(1 - \frac{\overline{k}}{o^n}\right)^{o^n - k}} \quad (4)$$

The curves $\rho_{o,n}(\overline{k})$ in Fig. 4 for the binary and ternary algorithm have been calculated numerically. For comparison, the throughput of the coin flip splitting algorithm with a Poisson distributed size of a start set is given.

The distribution of delays is also calculated by a recursion. We define the probability $p_{o,n,k}(l, m)$ of m mobiles still being involved in a collision of a start set of k terminals after l splitting steps.

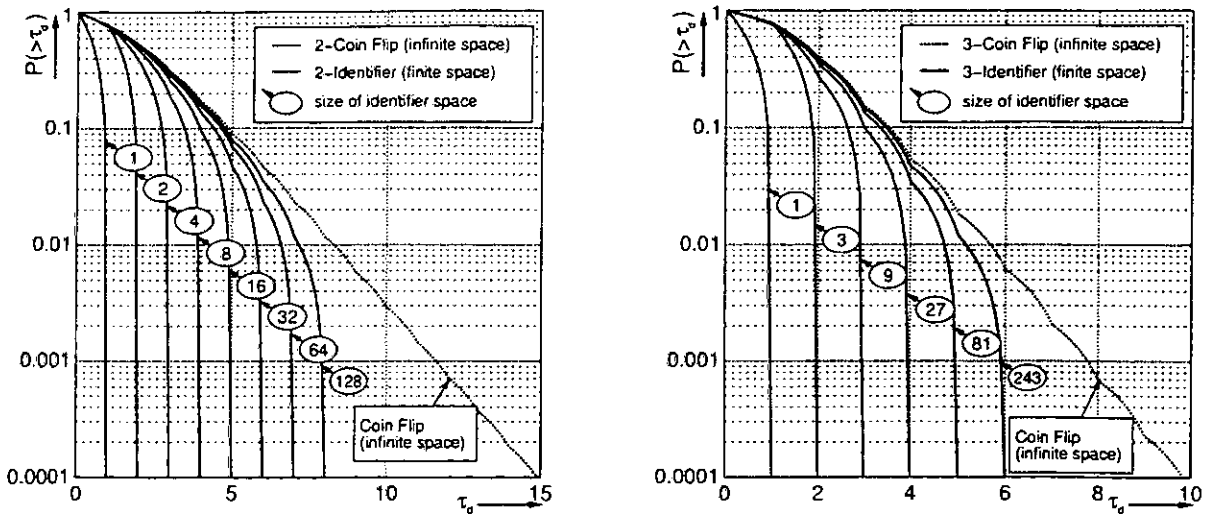


Figure 5: Complementary distribution of delays $P_{o,n,\overline{k}}(\tau_d > t)$ of binary ($o = 2$) and ternary ($o = 3$) identifier splitting algorithm with binomial distributed number of terminals in a start set (operating point $\overline{k} = 1.5$)

$$p_{2,n,k}(l,m) = \begin{cases} 1 & \mathit{for} \quad m = k, l = 0 \\ 1 & \mathit{for} \quad m = 0, k \leq 1, l > 0 \\ \displaystyle\sum_{i=\max(0,k-2^{n-1})}^{\min(k,2^{n-1})} \frac{\binom{2^{n-1}}{i}\binom{2^{n-1}}{k-i}}{\binom{2^n}{k}} \sum_{j=\max(0,m-(k-i))}^{\min(m,i)} p_{2,n-1,i}(l-1,j) \cdot p_{2,n-1,k-i}(l-1,m-j) & \\ & \mathit{for} \quad k > 1, l > 0 \\ 0 & \mathit{else} \end{cases} \tag{5}$$

$$p_{3,n,k}(l,m) = \begin{cases} 1 & \mathit{for} \quad m = k, l = 0 \\ 1 & \mathit{for} \quad m = 0, k \leq 1, l > 0 \\ \displaystyle\sum_{i=0}^{\min(k,3^{n-1})} \sum_{j=\max(0,k-i-3^{n-1})}^{\min(k-i,3^{n-1})} \frac{\binom{3^{n-1}}{i}\binom{3^{n-1}}{j}\binom{3^{n-1}}{k-i-j}}{\binom{3^n}{k}} \sum_{r=0}^{\min(m,i)} \sum_{s=\max(0,m-r-(k-i-j))}^{\min(m-r,j)} \cdots & \\ \quad \cdots p_{3,n-1,i}(l-1,r) \cdot p_{3,n-1,j}(l-1,s) \cdot p_{3,n-1,k-i-j}(l-1,m-r-s) & \\ & \mathit{for} \quad k > 1, l > 0 \\ 0 & \mathit{else} \end{cases} \tag{6}$$

The complementary distribution of the delays τ_d results in:

$$P_{o,n,k}(\tau_d > t) = \frac{1}{k} \sum_{m=0}^{k} m \cdot p_{o,n,k}(\lfloor t/\tau_P \rfloor, m) \quad , \quad k > 0 \tag{7}$$

Taking into account the binomial distributed size $\overline{k}$ of a start set, we get the complementary distribution of delays $P_{o,n,\overline{k}}(\tau_d > t)$ in (8). Fig. 5 shows the curves with the numerically calculated values.

$$P_{o,n,\overline{k}}(\tau_d > t) = \frac{1}{\overline{k}} \sum_{k=0}^{o^n} \binom{o^n}{k} \left(\frac{\overline{k}}{o^n}\right)^k \left(1 - \frac{\overline{k}}{o^n}\right)^{o^n - k} \cdot k \cdot P_{o,n,k}(\tau_d > t) \quad , \quad k > 0 \tag{8}$$

The calculation of the average delay $\overline{\tau}_d$ is based on the probability function of the number of periods required for the successful transmission of a packet in eq. (9) and (10).

$$p_{2,n,k}(l) = \begin{cases} 1 & \mathit{for} \quad k = 1, l = 1 \\ \displaystyle\frac{1}{k} \sum_{i=\max(0,k-2^{n-1})}^{\min(k,2^{n-1})} \frac{\binom{2^{n-1}}{i}\binom{2^{n-1}}{k-i}}{\binom{2^n}{k}} (i \cdot p_{2,n-1,i}(l-1) + (k-i) \cdot p_{2,n-1,k-i}(l-1)) & \\ & \mathit{for} \quad k > 1, l > 1 \\ 0 & \mathit{else} \end{cases} \tag{9}$$

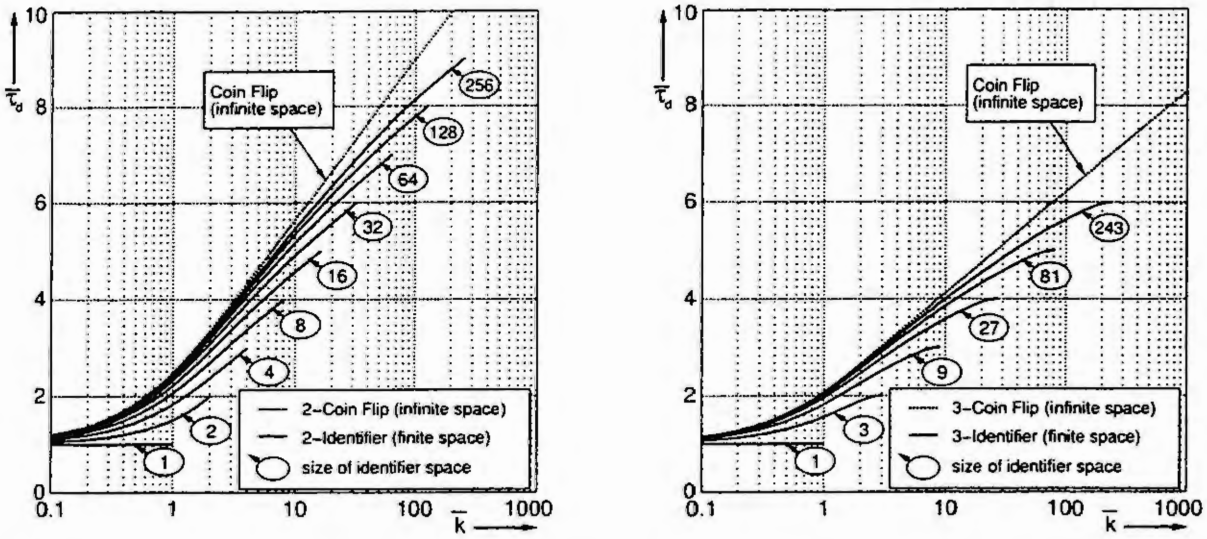


Figure 6: Average delay $\overline{\tau}_{d_{o,n}}(\overline{k})$ of binary ($o = 2$) and ternary ($o = 3$) identifier splitting algorithm with binomial distributed number of terminals in a start set

$$p_{3,n,k}(l) = \begin{cases} 1 & \textit{for} \quad k = 1, l = 1 \\ \frac{1}{k} \sum\limits_{i=0}^{\min(k,3^{n-1})} \sum\limits_{j=\max(0,k-i-3^{n-1})}^{\min(k-i,3^{n-1})} \frac{\binom{3^{n-1}}{i}\binom{3^{n-1}}{j}\binom{3^{n-1}}{k-i-j}}{\binom{3^n}{k}} \Big(\cdots & \\ \cdots i \cdot p_{3,n-1,i}(l-1) + j \cdot p_{3,n-1,j}(l-1) + (k-i-j) \cdot p_{3,n-1,k-i-j}(l-1) \Big) & \\ & \textit{for} \quad k > 1, l > 1 \\ 0 & \textit{else} \end{cases} \tag{10}$$

Using these equations, the average delay $\overline{\tau}_{d_{o,n}}(\overline{k})$ and its variance σ^2 can be calculated. The curves $\overline{\tau}_{d_{o,n}}(\overline{k})$ are shown in Fig. 6.

$$\overline{\tau}_{d_{o,n}}(\overline{k}) = \tau_P \cdot \frac{1}{\overline{k}} \sum_{k=0}^{o^n} \left(\binom{o^n}{k} \left(\frac{\overline{k}}{o^n}\right)^k \left(1 - \frac{\overline{k}}{o^n}\right)^{o^n - k} \cdot k \sum_{l=0}^{n+1} l \cdot p_{o,n,k}(l) \right) \quad , \quad \overline{k} > 0 \tag{11}$$

$$\sigma^2\left(\overline{\tau}_{d,o}(\overline{k})\right) = \frac{1}{\overline{k}} \sum_{k=0}^{o^n} \left(\binom{o^n}{k} \left(\frac{\overline{k}}{o^n}\right)^k \left(1 - \frac{\overline{k}}{o^n}\right)^{o^n - k} \cdot k \sum_{l=0}^{n+1} (l \cdot \tau_P - \overline{\tau}_{d_{o,n}}(\overline{k}))^2 \cdot p_{o,n,k}(l) \right) \quad , \quad \overline{k} > 0 \tag{12}$$

4 Derivation of the Adaptive Identifier Splitting Algorithm

If the size of a start set is large, the throughput can be increased and delays reduced, if the first splitting steps are skipped. This is equivalent to a dynamically selected splitting order o^{n_1} of the first splitting step. That n_1 is chosen that maximizes throughput:

$$\rho_{opt_{o,n}}(\overline{k}) = \max\left(\rho_{o,0}\left(\frac{\overline{k}}{o^n}\right), \cdots, \rho_{o,n-n_1}\left(\frac{\overline{k}}{o^{n_1}}\right), \cdots, \rho_{o,n}\left(\overline{k}\right) \right) \tag{13}$$

The resulting curves of $\rho_{opt_{o,n}}(\overline{k})$ are shown in Fig. 7. The curves result from a piecewise composition of segments of the curves in Fig. 4. The same applies for the average delay $\overline{\tau}_{d_{opt\,o,n}}(\overline{k})$ in Fig. 8.

The exact ordinate values $\overline{k}$ of the transitions between segments can be calculated by (14).

$$\rho_{o,n-n_1-1}\left(\frac{\overline{k}}{o^{n_1+1}}\right) = \rho_{o,n-n_1}\left(\frac{\overline{k}}{o^{n_1}}\right) \tag{14}$$

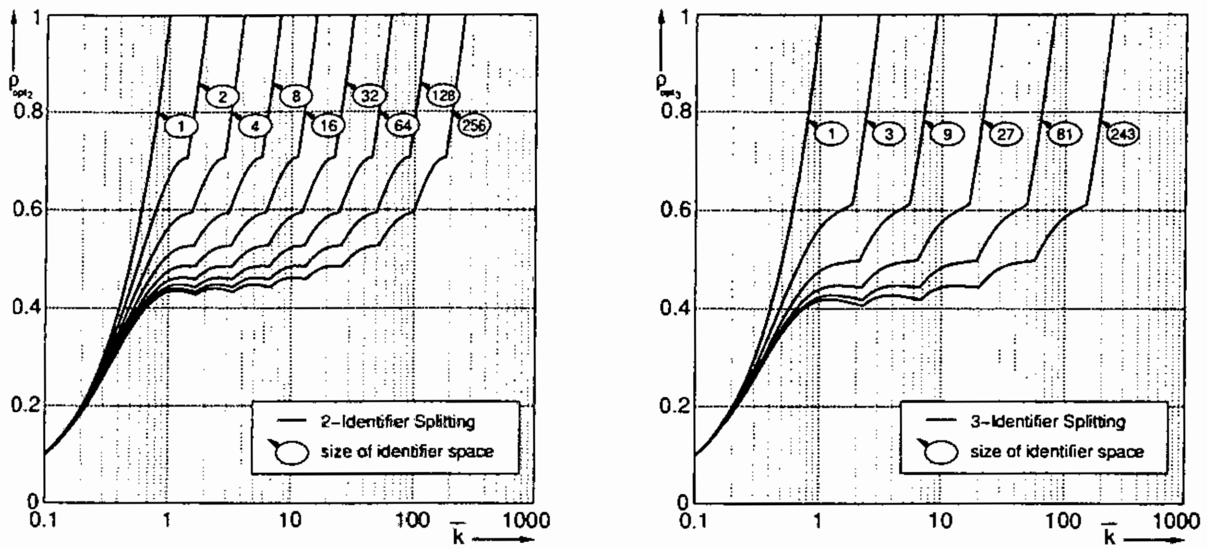


Figure 7: Optimal throughput $\rho_{opt_{o,n}}(\overline{k})$ of identifier splitting algorithm with adaptive order of first splitting step

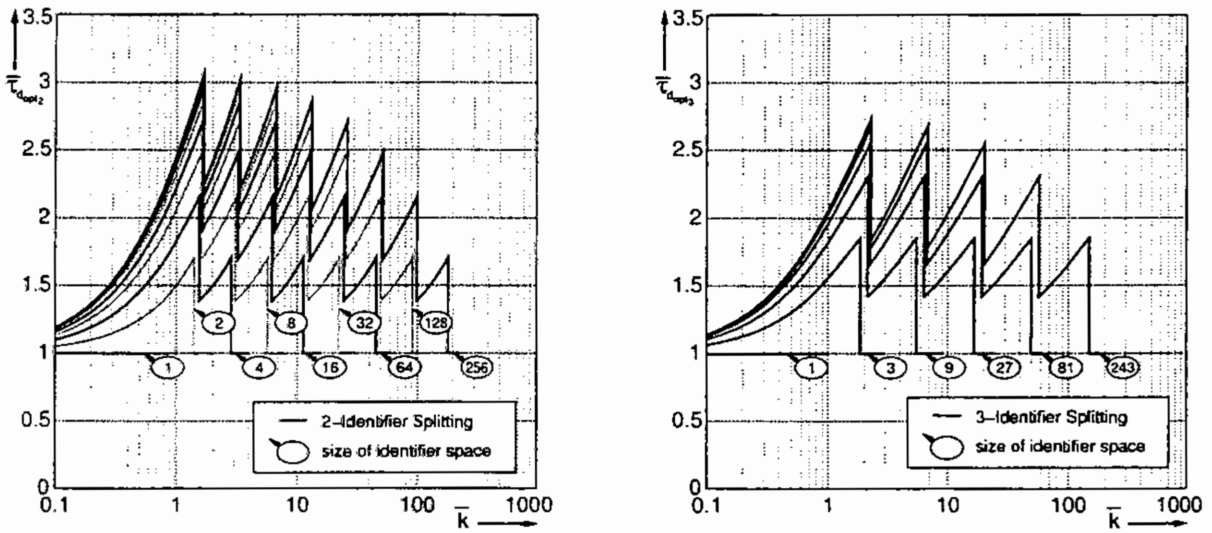


Figure 8: Optimal average delay $\overline{\tau}_{d_{opt_{o,n}}}(\overline{k})$ of identifier splitting algorithm with adaptive order of first splitting step

Table 1 summarizes the ordinate values for the binary and ternary algorithm. Dependent on the dimension of the identifier space and the known or estimated size $\overline{k}$ of a start set, the optimal number of initial slots can be determined.

The comparison of Fig. 7 and Fig. 8 of the adaptive identifier splitting algorithm with the corresponding Figures 4 and 6 of the original identifier splitting algorithm demonstrated the dramatical improvement of performance that can be realized, if the size of a start set is known or can at least be estimated.

5 Simulation of Medium Access Control Protocol with Probing Algorithm

The analysis of the identifier splitting algorithm with an adaptive number of initial slots has shown, that the optimal size of a start set is approximately 1.5 for binary splitting and 2 for ternary splitting with some deviations depending on the dimension of the identifier space. Now we return to the model of the RACH. We can estimate the probability $p_{\geq 1}$ of at least one arrival at terminal i during the interval $n_{idle,i} \cdot \tau_P$ since its last transmission of a packet:

Size of identifier space										
1	2	4	8	16	32	64	128	256		Number of initial slots
> 0	> 0	> 0	> 0	> 0	> 0	> 0	> 0	> 0	1	
	> 1.4142	> 1.5429	> 1.6091	> 1.6432	> 1.6607	> 1.6695	> 1.6739	> 1.6761	2	
		> 2.8284	> 3.0858	> 3.2181	> 3.2865	> 3.3214	> 3.3390	> 3.3478	4	
			> 5.6569	> 6.1716	> 6.4362	> 6.5730	> 6.6427	> 6.6780	8	
				> 11.314	> 12.343	> 12.872	> 13.146	> 13.285	16	
					> 22.628	> 24.687	> 25.745	> 26.292	32	
						> 45.255	> 49.373	> 51.490	64	
							> 90.510	> 98.746	128	
								> 181.02	256	

Size of identifier space							
1	3	9	27	81	243		Number of initial slots
> 0	> 0	> 0	> 0	> 0	> 0	1	
	> 1.8391	> 2.1306	> 2.2352	> 2.2711	> 2.2832	3	
		> 5.5173	> 6.3919	> 6.7057	> 6.8134	9	
			> 16.5520	> 19.1756	> 20.1171	27	
				> 49.6560	> 57.5269	81	
					> 148.9689	243	

Table 1: Optimal number of initial slots dependent on the dimension n of the identifier space and the known or estimated size $\overline{k}$ of a start set for binary and ternary identifier splitting algorithm

$$p_{\geq 1,i} = 1 - C_i \cdot (1-p)^{n_{idle,i}} \tag{15}$$

The parameter C_i is set to 1 and will be explained later.

Our new medium access control protocol for the RACH can be considered as an unblocking adaptive identifier splitting algorithm. We call it *probing algorithm.* At the beginning of each period it divides the identifier space in a variable number t of consecutive intervals and assigns one slot to each interval. The l-th interval is starting with terminal i_l and ending with terminal $i_{l+1} - 1$, with $i_1 = 0$ and $i_t = o^n - 1$. It contains $K_l = i_{l+1} - i_l$ terminals. K_l has to be maximized under the constrain (16).

$$N_l = \sum_{i=i_l}^{i_{l+1}-1} p_{\geq 1,i} < W \tag{16}$$

With the parameter W the probability of a successful transmission can be adjusted. At the end of a period the results of accesses can be used to correct the estimation of $p_{\geq 1,i}$. If no or one transmission happened in a slot, $n_{idle,i}$ is reset to zero and C_i to 1 for all involved station. If a collision occurred in the slot belonging to the l-th interval, the number $N_{coll,l}$ of involved terminals is estimated by (17).

$$N_{coll,l} = N_l \frac{1 - \left(1 - \frac{N_l}{K_l}\right)^{K_l - 1}}{1 - \left(1 - \frac{N_l}{K_l}\right)^{K_l} - N_l \left(1 - \frac{N_l}{K_l}\right)^{K_l - 1}} \tag{17}$$

This estimation is based on the assumption of a binomial distribution of N_l. This is no exact model but a sufficient approximation. We correct the estimation of $p_{\geq 1,i}$ by adjusting C_i:.

$$C = \frac{K_l - N_{coll,l}}{K_l - N_l} \tag{18}$$

$$C_{i,new} = C \cdot C_{i,old} \tag{19}$$

After a successful or no transmission on a slot, C_i of the terminals in the belonging interval is reset to 1.

The approximation made above requires a special treatment of terminals with high $p_{\geq 1,i}$. To avoid high delays, terminals with $p_{\geq 1,i} > W/2$ are polled in specific slots. The same happen with terminals, that have been involved in more than n consecutive collisions with n being the dimension of the identifier space.

The performance of the protocol has been evaluated by stochastic simulations. The number k of terminals, the arrival probability p and the parameter W have been varied. The average and maximum delay as well as the throughput ρ over p for $k = 5$ and $k = 20$ terminals is shown in the diagrams in Fig. 9 (with a relative error $\ll 0.01$). W has been chosen to 1.0 and 1.4. It can be seen, that W has the same effect like the order of a splitting algorithm. Lower values of W lead to shorter delays but reduces the throughput.

The determination of the optimal value of W requires a more precise model of the MAC protocol and the surrounding system. But our results may be a guideline for finding optimal parameter settings of a real MAC protocol.

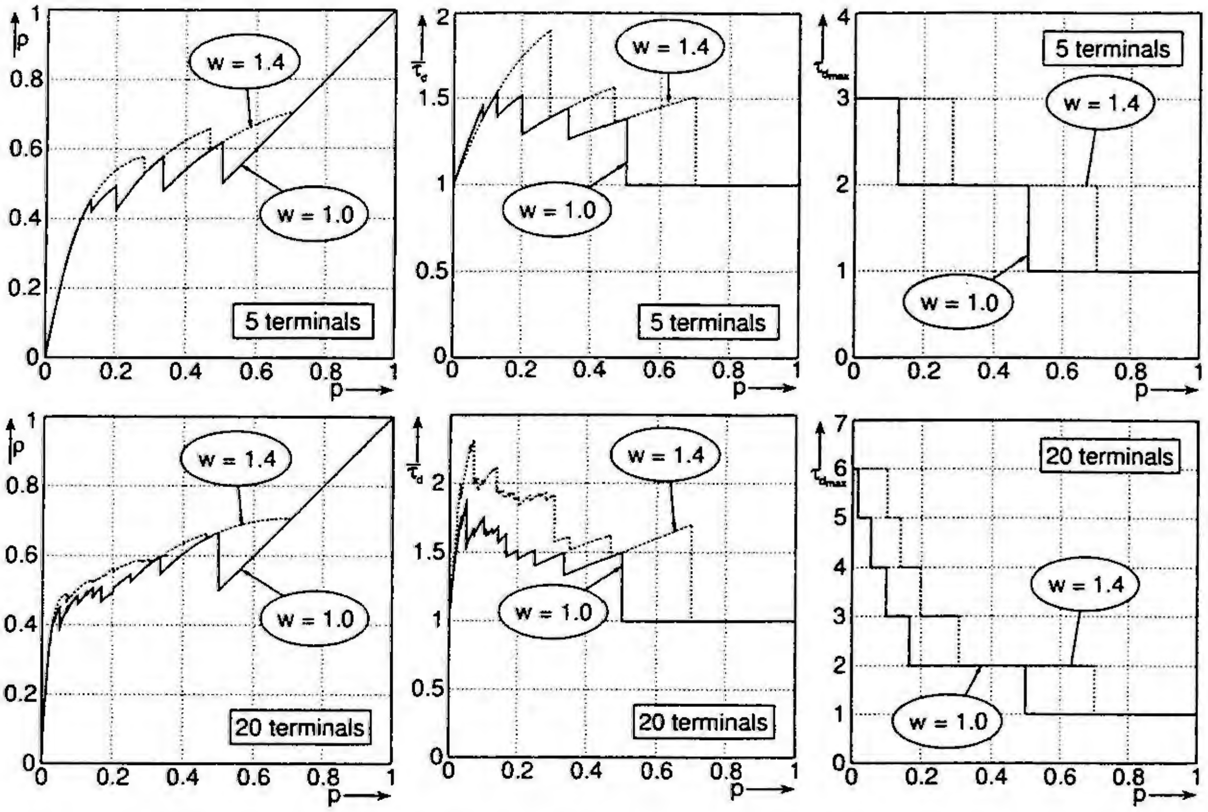


Figure 9: Throughput ρ, average delay $\bar{\tau}_d$ and maximum delay $\tau_{d_{max}}$ of probing algorithm with $k = 5$ and $k = 20$ terminals

6 Conclusions

Our new medium access control protocol has been developed taking into account the results of the analysis of the identifier splitting algorithm. The protocol offers a good performance by combining the advantages of the identifier splitting algorithm and pure polling. We assumed an error-free feedback. This is no realistic model for a radio channel with noise and interference. We intent to modify our algorithms in order to use a soft decision feedback. Based on the accuracy of this feedback, the grade of correcting C_l can be adjusted.

7 References

[1] D. Bertsekas, R. Gallager. *Data Networks.* Prentice-Hall, Englewood Cliffs, NJ, 1987.

[2] J. I. Capetanakis. *Tree Algorithms for Packet Broadcast Channels.* IEEE Trans. Inform. Theory, Vol. 25, No. 5, pp. 319–329, 1979.

[3] P. Mathys, P. Flajolet. *Q-ary Collision Resolution Algorithm in Random Access Systems with Free or Blocked Channel Access.* IEEE Trans. Inform. Theory, Vol. 31, pp. 217–243, 1985.

[4] D. Petras, A. Krämling, A. Hettich. *MAC protocol for Wireless ATM: contention free versus contention based transmission of reservation requests.* In *PIMRC'96*, Taipei, Taiwan, October 1996.

[5] ETSI RES10. *HIgh PErformance Radio Local Area Network (HIPERLAN), Requirements and Architectures.* Draft ETR, Sophia Antipolis, France, 1996.

BOTTLENECK ANALYSIS FOR COMPUTER AND COMMUNICATION SYSTEMS WITH WORKLOAD VARIABILITIES & UNCERTAINTIES*

J. Lüthi and G. Haring
Institut für Angewandte Informatik und Informationssysteme, Universität Wien
Lenaugasse 2/8, A-1080 Wien, Austria
e-mail: {luethi,haring}@ani.univie.ac.at

Abstract. Bottleneck analysis using queueing network models is an important technique for the performance analysis and capacity planning of computer and communication systems. Conventional single class as well as multiclass queueing network models use single mean values as input parameters. However, uncertainties and variabilities in service demands may exist in many types of systems. Using models with a single aggregate mean value for each parameter for such systems can lead to inaccurate or even incorrect results. This paper proposes to use histograms for characterizing model parameters that are associated with workload uncertainty and/or variability. Methods to identify system bottlenecks as well as first cut approximation tools for the potential effects of service demand modifications for queueing networks with histogram-based input parameters are presented in the paper.

1 Introduction

Computer and communication systems require effective tools for predicting their performance and for analyzing their behavior. Analytic models such as single class queueing networks can be used for performance estimation of such systems [14]. These techniques are popular because of their relatively low cost in comparison with simulations and benchmarks. A conventional analytic performance model accepts a set of single valued parameters (such as service demands for different devices) and produces a single point measure for each performance index of interest (such as the mean response time or mean processor utilization). However, the exact value of every parameter for the system may not always be known to the performance analyst leading to uncertainties in workload characterization (WLC). Furthermore, systems are often subject to variabilities in the workload [2]. Different phases in the operation of the system under study may lead to a different set of parameters that characterize each phase.

Using queueing networks for modeling system performance is a well-known and popular technique (see e.g. [9] for an introduction to the analysis of computer systems using queueing network models). In this paper, we consider analysis of single class closed queueing network models (QNM). Besides exact solution techniques such as the mean value analysis (MVA) algorithm [15] that provides a single mean value for performance measures, analysis of system bottlenecks (BN) is a popular technique especially for the capacity planning of large systems [14, 16]. It requires only very little computation and is thus often preferred as a first cut modeling tool. However, existing BN and corresponding modification analysis techniques are inadequate for handling parameters characterized by uncertainties or variabilities. Using a single aggregated mean for each input parameter can lead to an incorrect identification of BNs.

Appropriate characterization of the workload is required to capture any uncertainty or variability associated with it. We propose to characterize the mean service demand d_k of each device k in the system that exhibits variability and/or uncertainty by a histogram of mean service demands $H(d_k)$. The histogram consists of a number of intervals and associated probabilities of occurrence. The histogram-based performance analysis technique is applicable in single class systems in which various classes of demands are observed for a given device. Multiclass queueing networks are quite popular in a different situation when various classes of behavior are observed among the customers. Each of these systems is concerned with the variability in a particular aspect of the system model: temporal in the first and spatial in the second. Variabilities and uncertainties in workload may lead to the existence of multiple devices which may be the system BNs with a certain probability.

Association of a single interval or range of values with model parameters and performance measures is described in [10] and [13]. The adaptation of the MVA algorithm for product form closed single class QNMs to handle models with variabilities and uncertainties in workload is considered in [11]. Also existing bounding techniques for single class QNMs have been adapted to handle histogram-based WLC [12]. Large inaccuracies have been reported as an effect of ignoring such variabilities. Existence of multiple

*This work was supported by the Austrian Science Foundation (FWF research grant No. S05303-MAT)

BNs for systems modeled with multiclass closed queueing networks is studied by Balbo and Serazzi [1]. Uncertainty analysis in the context of performability modeling of computer systems using Markov reward models is considered by Haverkort and Meeuwissen [7].

This paper is organized as follows: in Section 2, WLC for systems with variabilities and uncertainties in workload is discussed. Mathematical results which are needed for the modification analysis techniques presented in this work, are presented in Section 3. In Section 4, the generalization of conventional BN and modification analysis techniques to handle models with histogram-based input parameters is proposed and demonstrated along the lines of a small but illustrative example QNM. Section 5 presents a summary of the results and our conclusions.

2 Workload Characterization

A parameter X may be specified through a histogram $H(X)$ as follows:

$$X_1 = [\underline{x}_1,\ \overline{x}_1] : p_1, \quad X_2 = [\underline{x}_2,\ \overline{x}_2] : p_2, \quad \ldots \quad , X_m = [\underline{x}_m,\ \overline{x}_m] : p_m$$

with $\sum_{i=1}^{m} p_i = 1$. Each entry in the definition of X provided above is a two-tuple, an interval $[\underline{x}_i,\ \overline{x}_i]$ and an associated probability p_i. That is, with probability p_i, $\underline{x}_i \leq X \leq \overline{x}_i$. This general model can be used to represent uncertainties and/or variabilities. Uncertainties are characterized by the length of the interval, and an interval of width zero represents no uncertainty. Variability is described by the distribution of the probabilities p_i, with $m = 1$ corresponding to a workload with no variability. In the following subsections, we describe each possible type of model with the exception of the single value (SV) case which is already well-known and we give some examples.

Uncertainties

Associating intervals with input parameters of interest is useful when uncertainties are associated with parameter values. The probability of occurrence of any value within an interval can follow any given arbitrary distribution. However, in this work we assume uniform distribution of parameter values within the intervals. Consider for example software performance engineering that integrates performance modeling with the various phases of software design and implementation [17]. Uncertainties may be associated with model parameters for various reasons. For example, exact values of system parameters are often unknown in early stages of system design. Although uncertainties may be associated with one or more system parameters, the designer may have a good idea about the range of values associated with these parameters from previous experience with similar systems. A single interval with $p_1 = 1$ may be used to describe the range of values associated with each such parameter. Also, within hierarchical performance modeling (see for example [3]), bounding techniques used in one layer of the modeling hierarchy may lead to input intervals on the next layer. For example, Hartleb and Mertsiotakis propose bounding techniques for the runtime of parallel programs [6], which are integrated in the modeling tool PEPP [5]. Bounds for the mean runtime of parallel programs are also provided by the *serialization analysis* technique used in the PAMELA approach [18].

In this work we assume that the intervals for input parameters are specified, no matter how such parameters are obtained which is beyond the scope of this paper.

Variabilities

Variabilities in workload can occur in systems which are characterized by different phases of operation. As an example consider a client-server system, where different mean demands at a given device may occur during various time periods. Such a variability may be exhibited by a point-of-sale system where different amount of work per transaction occurs during different periods of the day. Variabilities in service demands can also occur implicitly in systems. For example, different service demands have been observed in a database system described in [2]: during periods of time when less memory was available for transaction processing a larger number of I/O operations was observed. Even though the system is in steady state during each phase of operation, neither a conventional single class nor a multiclass queueing network is adequate for the computation of system performance. Using single mean values for the service demands often leads to inaccurate results for these systems. The technique proposed in this paper is apt for the analysis of such systems.

Consider a QNM with K queueing devices. In the sequel we will treat a variability model (with or without uncertainties) as a list of variability combinations with corresponding probabilities of occurrence.

Each of the $i = 1, \ldots, I$ elements of the list is of the form: $(D_{1,i}, \ldots, D_{K,i}) : p_i$. In this list, $D_{k,i} = [\underline{d}_{k,i}, \overline{d}_{k,i}]$ represents the service demand interval for the k^{th} device within the i^{th} variability component which has probability of occurrence p_i. If the system is modeled without uncertainties, intervals $D_{k,i}$ are replaced by the respective single values $d_{k,i}$. Note that these lists can either be derived from multiple existing histograms (see [11] for the transformation of multiple parameter histograms to variability lists) or they can directly be specified. Such variability lists can also be used to approximate workloads specified as probability distributions. In the example presented in Section 4, we use directly specified variability lists. We denote uncertainty models without variability by UN, WLC with variability without uncertainties by VA, and we denote the combination of variability and uncertainty by VU.

3 Mathematical Preliminaries

In this section, expressions which are needed to compute the matrices defined in Section 4 are derived. Consider n stochastically independent random variables $X_1, \ldots, X_n$, uniformly distributed in the interval $[a, b]$, $a > 0$. Let $X_{(j)}$ denote the corresponding j^{th} order statistic (see for example [4]). In general, the probability density function (pdf) of the j^{th} order statistic from a continuous population with cumulative distribution function (cdf) $F_X(x)$ and pdf $f_X(x)$ is [4]:

$$f_{X_{(j)}}(x) = \frac{n!}{(j-1)!(n-j)!} f_X(x)[F_X(x)]^{j-1}[1 - F_X(x)]^{n-j}.$$

In the case of uniform distributions in the interval $[a, b]$ we get:

$$f_{X(j)}(x) = \frac{n!}{(j-1)!(n-j)!(b-a)^n}(x-a)^{j-1}(b-x)^{n-j}. \tag{1}$$

That is, $X_{(j)}$ is $\beta(j, n-j+1)$-distributed [8] and its expected value is well-known [4, 8]:

$$E(X_{(j)}) = a + \frac{(b-a)j}{n+1}. \tag{2}$$

In Section 4, we also use expected values of reciprocals of order statistics of uniformly distributed random variables. Consider the random variables $Y_{(j)} = 1/X_{(j)}$. For the derivation of $E(Y_{(j)})$ presented in Theorem 1, we need the following considerations:

Lemma 1 *Let $u, v \in \mathbb{N}_0$, $a, b \in \mathbb{R}$.*

$$\Rightarrow \quad \int_a^b (x-a)^u (b-x)^v \, dx = \frac{(b-a)^{u+v+1} u! v!}{(u+v+1)!}.$$

Proof: Let $c = (b - a)$. Parameter transformation and integration by parts yields:

$$\begin{aligned} \int_a^b (x-a)^u (b-x)^v \, dx &= \int_0^c x^u (c-x)^v \, dx = \underbrace{\frac{x^{u+1}}{u+1}(c-x)^v \Big|_0^c}_{=0} + \frac{v}{u+1} \int_0^c x^{u+1}(c-x)^{v-1} \, dx \\ &= \frac{v}{u+1} \int_0^c x^{u+1}(c-x)^{v-1} \, dx. \end{aligned}$$

After repeated integration by parts (v times) we get:

$$\int_0^c x^u (c-x)^v \, dx = \frac{v(v-1)\cdots 2 \cdot 1}{(u+1)(u+2)\cdots(u+v)} \int_0^c x^{u+v} \, dx = \frac{u! v!}{(u+v+1)!} c^{u+v+1},$$

which completes the proof of the lemma. ■

Lemma 2 *Let $n \in \mathbb{N}_0$, $\alpha, \beta, \xi \in \mathbb{R}$ such that $\alpha\xi + \beta \neq 0$.*

$$\Rightarrow \quad \frac{\xi^n}{\alpha\xi + \beta} = \frac{(-\beta/\alpha)^n}{\alpha\xi + \beta} - \frac{1}{\beta} \sum_{i=0}^{n-1} (-\beta/\alpha)^{n-i} \xi^i. \tag{3}$$

Proof: We proof the lemma by multiplication of the right-hand side and the denominator of the left-hand side of (3):

$$\left(\frac{(-\beta/\alpha)^n}{\alpha\xi+\beta}-\frac{1}{\beta}\sum_{i=0}^{n-1}(-\beta/\alpha)^{n-i}\xi^i\right)\cdot(\alpha\xi+\beta)=(-\beta/\alpha)^n+\sum_{i=0}^{n-1}(-\beta/\alpha)^{n-(i+1)}\xi^{i+1}-\sum_{i=0}^{n-1}(-\beta/\alpha)^{n-i}\xi^i$$

$$= \quad (-\beta/\alpha)^n+\xi^n+\sum_{i=1}^{n-1}(-\beta/\alpha)^{n-i}\xi^i-\sum_{i=1}^{n-1}(-\beta/\alpha)^{n-i}\xi^i-(-\beta/\alpha)^n=\xi^n.$$

■

Corollary 1 *Let $n \in \mathbb{N}_0$, $\alpha, \beta, \xi \in \mathbb{R}$ such that $\alpha\xi + \beta \neq 0$.*

$$\Rightarrow \quad \int \frac{\xi^n}{\alpha\xi+\beta}d\xi \;=\; \frac{(-\beta/\alpha)^n}{\alpha}\log|\alpha\xi+\beta|-\frac{1}{\beta}\sum_{i=1}^{n}\frac{(-\beta/\alpha)^{n-(i-1)}}{i}\xi^i.$$

Proof: The corollary follows from Lemma 2 through term-by-term integration. ■

Lemma 3 *Let $u, v \in \mathbb{N}_0$, $a, b \in \mathbb{R}$.*

$$\Rightarrow \quad \int_a^b \frac{1}{x}(x-a)^u(b-x)^v dx$$

$$= \quad (-a)^u b^v \log|b/a| - (-a)^u\sum_{i=1}^{v}\frac{b^{v-i}(b-a)^i}{i}+\sum_{i=1}^{u}\frac{(-a)^{u-i}(b-a)^{v+i}(i-1)!v!}{(v+i)!}.$$

Proof: Again, we denote the length of the interval $[a,b]$ by $c = b - a$. Application of Lemma 2, Lemma 1, and Corollary 1 yields:

$$\int_a^b \frac{1}{x}(x-a)^u(b-x)^v dx = \int_0^c \frac{x^u}{x+a}(c-x)^v dx$$

$$\overset{\text{Lemma 2}}{=} \quad \int_0^c \left[\frac{(-a)^u}{x+a}-\frac{1}{a}\sum_{i=0}^{u-1}(-a)^{u-i}x^i\right]\cdot(c-x)^v dx$$

$$= \quad (-a)^u\int_0^c\frac{(c-x)^v}{x+a}dx+\sum_{i=0}^{u-1}\left[(-a)^{u-(i+1)}\int_0^c x^i(c-x)^v dx\right]$$

$$\overset{\text{Lemma 1}}{=} \quad (-a)^u\int_0^c\frac{x^v}{b-x}dx+\sum_{i=0}^{u-1}\frac{(-a)^{u-(i+1)}c^{v+i+1}i!v!}{(i+v+1)!}$$

$$\overset{\text{Corollary 1}}{=} \quad (-a)^u\left(-b^v\log|b-x|-\sum_{i=1}^{v}\frac{b^{v-i}}{i}x^i\right)\Bigg|_0^c+\sum_{i=1}^{u}\frac{(-a)^{u-i}c^{v+i}(i-1)!v!}{(v+i)!}$$

$$= \quad (-a)^u b^v\log|b/a|-(-a)^u\sum_{i=1}^{v}\frac{b^{v-i}c^i}{i}+\sum_{i=1}^{u}\frac{(-a)^{u-i}c^{v+i}(i-1)!v!}{(v+i)!}.$$

■

Theorem 1 *Let $X_1, \ldots, X_n$ be stochastically independent random variables, uniformly distributed in $[a,b]$, where $a, b \in \mathbb{R}$, $a > 0$. Let the corresponding j^{th} order statistic be denoted by $X_{(j)}$. The expected value of the reciprocal $Y_{(j)} = 1/X_{(j)}$ of the j^{th} order statistic is given by:*

$$\Rightarrow \quad E(Y_{(j)}) \;=\; \frac{n!}{(j-1)!(n-j)!(b-a)^n}\left((-a)^{j-1}b^{n-j}\log|b/a|\right)$$

$$- \quad \frac{n!}{(j-1)!(n-j)!}\left((-a)^{j-1}\sum_{i=1}^{n-j}\frac{b^{n-j-i}(b-a)^{i-n}}{i}\right)$$

$$+ \quad \frac{n!}{(j-1)!}\left(\sum_{i=1}^{j-1}\frac{(-a)^{j-i-1}(b-a)^{i-j}(i-1)!}{(n-j+i)!}\right).$$

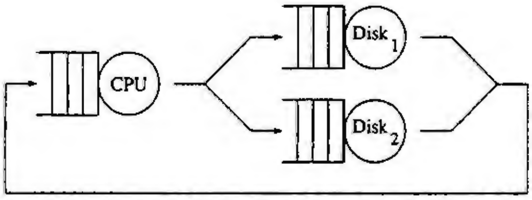


Figure 1. QNM of a computer system with a CPU and two disks.

Proof: Using the pdf denoted in (1), we have:

$$E(Y_{(j)}) \quad = \quad \int_a^b \frac{1}{y} f_{X_{(j)}}(y) dy = \frac{n!}{(j-1)!(n-j)!(b-a)^n} \int_a^b \frac{1}{y} (y-a)^{j-1} (b-y)^{n-j} dy. \tag{4}$$

Application of Lemma 3 to (4) with $u = j - 1$ and $v = n - j$ yields the proof of the theorem. ■

4 Bottleneck Analysis

In a single class QNM with conventional (SV) parameter specification, the device with highest service demand d_{max} restricts the potential performance of the modeled system (see [9]). In particular, for any number of jobs N, the system throughput $x(N)$ is limited by $x(N) < 1/d_{max}$. Moreover, for the asymptotic behavior of the model outputs we know: $x(N) \rightarrow 1/d_{max}$ and $r(N)/N \rightarrow d_{max}$ for $N \rightarrow \infty$ (for a formal proof see [10]) where $r(N)$ denotes the system response time.

Considering the upper bound of system throughput $1/d_{max}$ given the maximum service demand d_{max} in a SV parametrized model, we call the device b_1 with highest service demand the *primary BN* of the system. Usually, *secondary*, *tertiary*, etc. BNs are also considered. This approach can be generalized to the concept of *j-ary BNs*. The j-ary BN (or *BN of degree* j) of the network is the device b_j with j^{th} highest service demand.

Goals of BN analysis are the identification of BNs as well as the analysis of the expected benefit of removing a BN regarding response time and throughput of the system under investigation. For SV models with a large number N of jobs in the system, the potential response time benefit associated with the j-ary BN b_j can be approximated by N times the difference between the service demand d_{b_j} of the j-ary BN and the service demand $d_{b_{j+1}}$ of the $j+1$-ary BN. Analogously, the throughput increase potential of the j-ary BN is $1/d_{b_{j+1}} - 1/d_{b_j}$. Note that the discussed improvement caused by the elimination of the j-ary BN is only to be expected if all k-ary BNs with $k < j$ are already eliminated. These definitions are not sufficient for the investigation of workload models for systems with variabilities and uncertainties. In different service demand combinations of a model for a system with variabilities, we may identify different stations to be the j-ary BN. Additionally, parameter intervals in models with uncertainties may be overlapping hindering the identification of a unique BN device for every BN degree. An approach to generalize the identification of BNs to systems with uncertainties is the so-called *set of potential BNs* proposed in [10]. In this work, we generalize this concept to histogram-based workload models and define for a closed single class QNM:

- The *Bottleneck Probability Matrix* (BNPM),
- The *Throughput Improvement Potential Matrix* (TIPM),
- The *Response Time Improvement Potential Matrix* (RIPM).

The following subsections deal with these matrices and their computation in more detail. We demonstrate the proposed BN analysis techniques with a small but illustrative example of a closed single class QNM with three queueing devices depicted in Figure 1, modeling the CPU and two disks of a computer system. Four different workload models are considered. The corresponding parameter values and intervals are listed in Table 1. Note that the service demands of the variability and uncertainty models of this example have got the same overall mean as the mean service demands of the SV model. This means that if the variability and uncertainty models (UN, VA, and VU) would be reduced to a conventional single valued workload model, this would have the same parameter set as the SV model of our example.

In the SV model, disk 1 is identified as the primary BN. The variability model without uncertainty however, describes a situation, where with probability $p_1 = 0.7$, disk 2 is the primary BN, whereas with

SV:		UN:	
$d_{cpu} = 125\,ms$		$D_{cpu} = [\;50\,ms, 200\,ms]$	
$d_{disk_1} = 220\,ms$		$D_{disk_1} = [180\,ms, 260\,ms]$	
$d_{disk_2} = 190\,ms$		$D_{disk_2} = [180\,ms, 200\,ms]$	
VA:		VU:	
$p_1 = 0.7$	$p_2 = 0.3$	$p_1 = 0.7$	$p_2 = 0.3$
$d_{cpu,1} = 50\,ms$	$d_{cpu,2} = 300\,ms$	$D_{cpu,1} = [\;40\,ms,\;60\,ms]$	$D_{cpu,2} = [280\,ms, 320\,ms]$
$d_{disk_1,1} = 220\,ms$	$d_{disk_1,2} = 220\,ms$	$D_{disk_1,1} = [200\,ms, 240\,ms]$	$D_{disk_1,2} = [200\,ms, 240\,ms]$
$d_{disk_2,1} = 250\,ms$	$d_{disk_2,2} = 50\,ms$	$D_{disk_2,1} = [210\,ms, 290\,ms]$	$D_{disk_2,2} = [\;40\,ms,\;60\,ms]$

Table 1. Four different workload models for the example QNM.

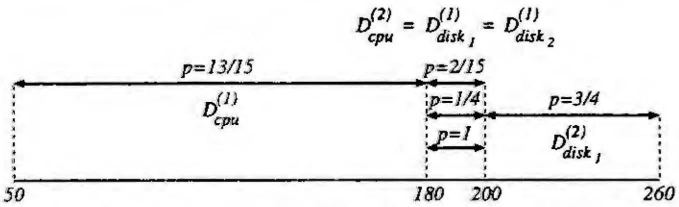


Figure 2. Transformation of UN parameters into subintervals

probability $p_2 = 0.3$ the CPU is the primary BN. In case of uncertainties (UN and VU), the detection of the BN devices and computation of associated BN probabilities is not straight forward, because of overlapping service demand intervals.

Bottleneck Probability Matrix (BNPM)

To provide a representation of multiple potential BNs of all degrees and the corresponding probabilities of occurrence, we propose the generalization of conventional BN identification to a *bottleneck probability matrix*. The BNPM for a QNM consisting of K queueing devices is a $K \times K$ matrix $B = (b_{i,j})$, representing the probabilities $b_{i,j}$ with which station i is a j-ary BN of the system under investigation. Note that the i^{th} row of B represents the distribution of possible BN locations for the i^{th} device, whereas the j^{th} column of B represents the distribution of all stations for the j^{th} BN degree. Thus, for the rows and columns of B we have:

$$\sum_{l=1}^{K} b_{i,l} \;=\; \sum_{l=1}^{K} b_{l,j} \;=\; 1, \quad \forall i,j = 1,\dots,K. \tag{5}$$

For SV models where all service demands d_k are different, $b_{i,j}$ is 1 if device i is *the* j-ary BN and 0 otherwise, since every BN degree is unique. If there are several devices with the same service demand, say m devices $k_1,\dots,k_m$ with service demand d_m which are of BN degree j, we define $b_{k_i,j+l} = 1/m$, $\forall i = 1,\dots,m$, $\forall l = 0,\dots,m-1$. This is motivated by the conservation of Equation (5) as well as by interpreting the identical service demands as identical intervals $[d_m - \epsilon, d_m + \epsilon]$ with $\epsilon \to 0$ (see construction of the BNPM for UN models). For the SV model described in Table 1, the corresponding BNPM is:

$$B_{SV} \;=\; \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

For UN models, lacking more specific information, we assume uniform distribution of parameters within intervals. To compute the respective BN probabilities, the interval parameters $(D_1,\dots,D_K)$ are transformed into subintervals $(D_1^{(1)},\dots,D_1^{(m_1)};\dots\dots;D_K^{(1)},\dots,D_K^{(m_K)})$, such that for all $D_k^{(i)}$, $D_l^{(j)}$ it holds that either $D_k^{(i)} = D_l^{(j)}$ or $D_k^{(i)} \cap D_l^{(j)} = \emptyset$ (see Figure 2). Note that this allows for a total ordering of the subintervals. To these subintervals, corresponding probabilities $p_k^{(i)} = |D_k^{(i)}|/|D_k|$ are assigned, representing the share of subinterval $D_k^{(i)}$ of its original interval D_k. Next, all possible combinations of subintervals together with their probabilities of occurrence are analyzed. Intermediate BNPMs for each of the combinations of subintervals are computed in analogy to the computation of BNPMs for SV models,

Combination	Probability	Primary BN	Secondary BN	Tertiary BN
$(D_{cpu}^{(1)}, D_{disk_1}^{(1)}, D_{disk_2}^{(1)})$	$p = \frac{13}{15} \cdot \frac{1}{4} = \frac{13}{60}$	2,3	2,3	1
$(D_{cpu}^{(1)}, D_{disk_1}^{(2)}, D_{disk_2}^{(1)})$	$p = \frac{13}{15} \cdot \frac{3}{4} = \frac{13}{20}$	2	3	1
$(D_{cpu}^{(2)}, D_{disk_1}^{(1)}, D_{disk_2}^{(1)})$	$p = \frac{2}{15} \cdot \frac{1}{4} = \frac{1}{30}$	1,2,3	1,2,3	1,2,3
$(D_{cpu}^{(2)}, D_{disk_1}^{(2)}, D_{disk_2}^{(1)})$	$p = \frac{2}{15} \cdot \frac{3}{4} = \frac{1}{10}$	2	1,3	1,3

Table 2. Subinterval combinations and corresponding BN orders for the UN model of the example from Table 1.

using the total ordering of the subintervals. The BNPM for the UN model is computed as the weighted sum of the intermediate BNPMs. We demonstrate this technique for the UN workload model of our example: The original parameter intervals $(D_{cpu}, D_{disk_1}, D_{disk_2})$ are transformed in the way described above to (see Figure 2):

$$\begin{aligned} D_{cpu}^{(1)} &= [\ 50\,ms, 180\,ms],\ p = 13/15, & D_{cpu}^{(2)} &= [180\,ms, 200\,ms],\ p = 2/15, \\ D_{disk_1}^{(1)} &= [180\,ms, 200\,ms],\ p = 1/4, & D_{disk_1}^{(2)} &= [200\,ms, 260\,ms],\ p = 3/4, \\ D_{disk_2}^{(1)} &= [180\,ms, 200\,ms],\ p = 1. \end{aligned}$$

All possible subinterval combinations together with the corresponding BN orders are listed in Table 2. Thus, the BNPM for the UN model is:

$$\begin{aligned} B_{UN} &= \frac{13}{60}\begin{pmatrix} 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \end{pmatrix} + \frac{13}{20}\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} + \frac{1}{30}\begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} + \frac{1}{10}\begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \end{pmatrix} \\ &= \begin{pmatrix} 0.011 & 0.061 & 0.928 \\ 0.869 & 0.119 & 0.011 \\ 0.119 & 0.819 & 0.061 \end{pmatrix}. \end{aligned}$$

For this workload model of our example QNM, we obtain the result that disk 1 is no longer the unique primary BN, but it is the primary BN with probability $p = 0.869$, whereas with probability $p = 0.119$, disk 2 is the primary BN, and with probability $p = 0.011$, the CPU is the primary BN. Analogous results can be observed for the secondary and tertiary BNs.

Both, variability models with uncertainties as well as variability models without uncertainties can be analyzed by computing the respective BNPM B_i for each variability combination $i = 1, \ldots, I$. The weighted sum $B = \sum_i p_i B_i$ of these intermediate BNPMs yields the aggregated BNPM for the variability model. In the VA case of our example, there are two variability combinations. In the first phase, which occurs with probability $p_1 = 0.7$, disk 2 is the primary BN, and disk 1 is the secondary BN. In the second phase, which has probability of occurrence $p_2 = 0.3$, the CPU is the primary BN, and again disk 1 is the secondary BN. This yields the following BNPM:

$$B_{VA} = 0.7\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} + 0.3\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.3 & 0 & 0.7 \\ 0 & 1.0 & 0 \\ 0.7 & 0 & 0.3 \end{pmatrix}. \qquad (6)$$

In the variability example with uncertainties (VU), the two interval parameter combinations are analyzed using the technique for UN models described above. Only analysis of the first variability combination requires the transformation to disjoint intervals, the second parameter interval combination is already of the desired form. The BNPM for the VU model is:

$$B_{VU} = 0.7\begin{pmatrix} 0 & 0 & 1.0 \\ 0.14 & 0.86 & 0 \\ 0.86 & 0.14 & 0 \end{pmatrix} + 0.3\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.3 & 0 & 0.7 \\ 0.1 & 0.9 & 0 \\ 0.6 & 0.1 & 0.3 \end{pmatrix}. \qquad (7)$$

Note that for both variability models the analysis shows much higher probabilities for the CPU and disk 2 to be the primary BN, whereas a conventional SV analysis for this example using aggregated mean values for the device service demands identifies disk 1 as the unique primary BN.

Throughput Improvement Potential Matrix (TIPM)

To analyze the potential benefit of removing a BN device, it is interesting to study the expected value of *throughput improvement* potential between succeeding degrees of BNs. We consider the asymptotic throughput bound $1/d_{max}$, and how this bound is improved by removing a BN device. The TIPM is defined as a $K \times (K-1)$ matrix $\Delta^X = (\delta^X_{i,j})$ such that $\delta^X_{i,j}$ represents the expected value of throughput improvement (compared to the BN of next higber degree) if device i is eliminated as the j-ary BN. Note that a throughput improvement obtained by removing the j-ary BN is only to be expected if all k-ary BNs with $k < j$ are already eliminated.

We first consider SV workload models. Let d_{b_j} denote the service demand of the j-ary BN. If all service demands are unique, an entry $\delta^X_{i,j}$ of the TIPM is zero if device i is not the j-ary BN, and $\delta^X_{i,j} = 1/d_{b_{j+1}} - 1/d_{b_j}$ if device i is the j-ary BN. For the SV model of our example this yields:

$$\Delta^X_{SV} = \begin{pmatrix} 0 & 0 \\ \frac{1}{0.19} - \frac{1}{0.22} & 0 \\ 0 & \frac{1}{0.125} - \frac{1}{0.19} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0.718 & 0 \\ 0 & 2.737 \end{pmatrix}.$$

From this matrix it can be seen that decreasing the service demand of the second device (the primary BN) can at most improve the upper bound for the system throughput by 0.718 jobs/second.

For the analysis of UN models, again the parameter intervals are transformed and combined as proposed in the discussion of the computation of the BNPM for UN models. That is we only have to consider parameter intervals which are either equal or not overlapping. All combinations resulting from the transformation into subintervals are considered independently and the corresponding results are summed up using the respective probabilities of occurrence as weights. Assuming uniform distribution within service demand intervals, the service demand of device k with service demand interval D_k can be considered as a uniformly distributed random variable X^{D_k}. If within a parameter interval combination μ_k devices have got the equal service demand interval D_k, we have to study the corresponding order statistics $X^{D_k}_{(j)}$, $j = 1, \ldots, \mu_k$.

In any of the transformed parameter interval combinations, two devices of succeeding BN degree are either parametrized with the same service demand interval, or they may be parametrized with disjoint intervals. Consider two devices with succeeding degrees of BNs b_j and b_{j+1} with disjoint service demand intervals D_{b_j} and $D_{b_{j+1}}$. Suppose that D_{b_j} appears with multiplicity μ_{b_j} and $D_{b_{j+1}}$ appears with multiplicity $\mu_{b_{j+1}}$. To be of succeeding BN degree, device b_j must be the one with lowest service demand of all μ_{b_j} devices with parameter interval D_{b_j} and device b_{j+1} must be the one with highest service demand of the $\mu_{b_{j+1}}$ devices with parameter interval $D_{b_{j+1}}$. Thus, we have to take into consideration the corresponding order statistics $X^{D_{b_j}}_{(\mu_{b_j})}$ and $X^{D_{b_{j+1}}}_{(1)}$. The expected throughput improvement potential between these two devices, given BN degrees j and $j+1$ is:

$$\delta^X_{b_j,j} = E\left(1/X^{D_{b_{j+1}}}_{(1)}\right) - E\left(1/X^{D_{b_j}}_{(\mu_{b_j})}\right).$$

If two devices of succeeding BN degree have got the same service demand interval D_{b_j} and there are μ_{b_j} devices with this parameter interval and κ devices with a higher parameter interval, the corresponding expected value for the throughput improvement potential is:

$$\delta^X_{b_j,j} = E\left(1/X^{D_{b_j}}_{(j+1-\kappa)}\right) - E\left(1/X^{D_{b_j}}_{(j-\kappa)}\right).$$

Thus, for the computation of the expected values for throughput improvement potential we have to compute expected values of the reciprocals of order statistics of uniformly distributed random variables. These expected values are given in Theorem 1.

The results for the UN workload model of our running example are (listed as the sum of the four possible subinterval combinations):

$$\begin{aligned} \Delta^X_{UN} &= \frac{13}{60}\begin{pmatrix} 0 & 0 \\ 0.093 & 4.488 \\ 0.093 & 4.488 \end{pmatrix} + \frac{39}{60}\begin{pmatrix} 0 & 0 \\ 0.895 & 0 \\ 0 & 4.585 \end{pmatrix} \\ &\quad + \frac{1}{30}\begin{pmatrix} 0.045 & 0.094 \\ 0.045 & 0.094 \\ 0.045 & 0.094 \end{pmatrix} + \frac{1}{10}\begin{pmatrix} 0 & 0.093 \\ 0.803 & 0 \\ 0 & 0.093 \end{pmatrix} = \begin{pmatrix} 0.002 & 0.012 \\ 0.684 & 0.976 \\ 0.022 & 3.965 \end{pmatrix}. \end{aligned}$$

With the workload characterization used in the UN model, similar to the SV model, service improvement at disk 1 promises the highest increase of system throughput. However, also the CPU as well as disk 2 are identified to promise a certain potential of throughput improvement.

The results for models with variabilities are obtained by computing the respective matrices Δ_i^X, $i = i, \ldots, I$, for all variability combinations and computing the weighted sum $\Delta^X = \sum_i p_i \Delta_i^X$. Considering the VA and VU models from our example, the expected values of throughput improvement are:

$$\Delta_{VA}^X = 0.7 \begin{pmatrix} 0 & 0 \\ 0 & 15.46 \\ 0.546 & 0 \end{pmatrix} + 0.3 \begin{pmatrix} 1.212 & 0 \\ 0 & 15.46 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0.364 & 0 \\ 0 & 15.455 \\ 0.382. & 0 \end{pmatrix},$$

$$\Delta_{VU}^X = 0.7 \begin{pmatrix} 0 & 0 \\ 0.028 & 15.69 \\ 0.555 & 4.422 \end{pmatrix} + 0.3 \begin{pmatrix} 1.220 & 0 \\ 0 & 15.72 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0.366 & 0 \\ 0.020 & 15.70 \\ 0.389 & 3.096 \end{pmatrix}.$$

Using VA and VU workload models, the CPU and disk 2 have got the highest probabilities to be the primary BN (see Equations (6) and (7)). Thus, it is not surprising that the highest throughput improvement potential due to elimination of the primary BN (see the first column of the matrix) is expected for these devices, whereas a decrease of the service demand at disk 1 does not produce a significant improvement in system throughput.

Response Time Improvement Matrix (RIPM)

In analogy to the TIPM, the potential improvement of system response time can be studied by computing a matrix listing the expected values of response time improvement between succeeding degrees of BNs. Asymptotically, the normalized system response time $r(N)/N$ is approximated by d_{max} (see [10]). Thus, in analogy to the TIPM, the entries of the SV RIPM Δ_{SV}^R are defined as $\delta_{i,j}^R = d_{b_j} - d_{b_{j+1}}$. The construction of the respective RIPMs for uncertainty models is the same as for the TIPMs. However, in the case of RIPMs, instead of the expected values of reciprocals of order statistics, the expected values of the order statistics, as described in Equation (2), have to be used.

5 Conclusions

Conventional analytic models for performance analysis of computer and communication systems accept single mean value parameters as input. However, uncertainties in parameter values and variabilities in workloads can make these techniques ineffective. Exact values for all the parameters are often unknown at early stages of system design but ranges of values that can be taken by these uncertain parameters may be available. Variabilities in workload may give rise to different mean service demands at different devices. For example, different mean service demands at a device may be observed during different periods of the day. Aggregating the workload and using a model characterized by a single mean demand for every device often leads to incorrect results. As proposed in this paper, characterization of the device demands of single class queueing network models by histograms instead of single mean values is appropriate in these situations. A histogram is a set of intervals and associated probabilities of occurrence. Histogram-based techniques are useful in a number of different situations that include the performance evaluation of conventional multiprogrammed systems and distributed systems characterized by variable workloads as well as in software performance engineering in which uncertainties are often associated with parameter values.

Bottleneck analysis is often used as a first cut analysis technique due to its low computational cost. Furthermore, identification of system BNs is especially important for capacity planning studies. Generalization of existing BN analysis techniques for single class queueing networks to handle input parameters characterized by histograms is presented in this paper. It is shown that using aggregated mean values as input parameters for systems with variabilities and/or uncertainties may lead to incorrect identification of BNs. Additionally, throughput and response time improvement potential matrices are proposed as a first cut means for modification analysis.

Consideration of more sophisticated modification analysis techniques is a subject of future work. Also BN analysis of models with uncertain parameter values without the simplifying uniformity assumption which may lead to corresponding interval matrices is worth being taken into consideration. Only single class queueing network models are considered in this paper. Adaptation of BN analysis techniques for multiclass QNMs to histogram-based WLC requires further investigation.

References

[1] Balbo, G. and Serazzi, G., Asymptotic Analysis of Multiclass Closed Queueing Networks: Multiple Bottlenecks. *Performance Evaluation*, 1997, in print.

[2] Buzen, J.P., A Modeler's View of Workload Characterization. In: Serazzi, G., ed., *Workload Characterization of Computer Systems and Computer Networks*, North-Holland, 1986, 67–72.

[3] Calzarossa, M., Merlo, A., Tessera, D., Haring, G., and Kotsis, G., A Hierarchical Approach to Workload Characterization for Parallel Systems. In: Hertzberger, B. and Serazzi G., eds., *Proc. High-Performance Computing and Networking, LNCS 919*, Springer Verlag, 1995, 102–109.

[4] Casella, G. and Berger, R.L., *Statistical Inference.* Duxbury Press, Belmont, California, 1990.

[5] Dauphin, P., Hartleb, F., Kienow, M., Mertsiotakis, V., and Quick, A., PEPP: Performance Evaluation of Parallel Programs, User's Guide – Version 3.3. Technical Report 17/93, IMMD VII, Univ. of Erlangen-Nürnberg, Germany, 1993.

[6] Hartleb, F. and Mertsiotakis, V., Bounds for the Mean Runtime of Parallel Programs. In: Pooley, R. and Hillston, J., eds., *Proc. Sixth Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation*, 1992, 197–210.

[7] Haverkort, B. and Meeuwissen, A.M.H., Sensitivity & Uncertainty Analysis of Markov-Reward Models. *IEEE Transactions on Reliability*, 44 (1995), 147–154.

[8] Johnson, N.L., Kotz, S., and Balakrishnan, N., *Continuous Univariate Distributions, Vol. 2.* John Wiley & Sons, New York e.a., 2nd edition, 1995.

[9] Lazowska, E.D., Zahorjan, J., Graham, G.S., and Sevcik, K.C., *Quantitative System Performance – Computer System Analysis Using Queueing Network Models.* Prentice-Hall, Englewood Cliffs, NJ, 1984.

[10] Lüthi, J. and Haring, G., Mean Value Analysis for Queueing Network Models with Intervals as Input Parameters. Tech. Rep. TR-950101, ANIIS, Univ. Wien, Austria, 1995.

[11] Lüthi, J., Majumdar, S., and Haring, G., Mean Value Analysis for Computer Systems with Variabilities in Workload. In: *Proc. IEEE Int. Computer Performance & Dependability Symposium, Urbana-Champaign, IL, USA, September 6-8, 1996*, IEEE Computer Society Press, 1996, 32–41.

[12] Lüthi, J., Majumdar, S., Kotsis, G., and Haring, G., Performance Bounds for Distributed Systems with Workload Variabilities & Uncertainties. *Parallel Computing, Special Issue "Distributed and Parallel Systems – Environments and Tools"*, 1997, in print.

[13] Majumdar, S. and Ramadoss, R., Interval-Based Performance Analysis of Computing Systems. In: Dowd, P. and Gelenbe, E., eds., *Proc. Third Int. Workshop on Modeling Analysis and Simulation of Computer and Telecommunication Systems*, IEEE Computer Society Press, 1995, 345–351.

[14] Menasce, D.A., Almeida, V.A.F., and Dowdy, L.W., *Capacity planning and performance modeling: from mainframes to client-server systems.* Prentice-Hall, Englewood Cliffs, NJ, 1994.

[15] Reiser, M. and Lavenberg, S.S., Mean-Value Analysis of Closed Multichain Queueing Networks. *Journal of the ACM*, 27 (1980), 313–322.

[16] Schweitzer, P.J., Serazzi, G., and Broglia, M., A Survey of Bottleneck Analysis in Closed Networks of Queues. In: Donatiello, L. and Nelson, R., eds., *Performance Evaluation of Computer and Commmunication Systems, LNCS 729*, Springer-Verlag, 1993, 491–508.

[17] Smith, C.U., *Performance Engineering of Software Systems.* Addison-Wesley, Reading, MA., 1990.

[18] Van Gemund, A.J.C., Compile-time Performance Prediction with PAMELA. In: *Proc. 4th Int. Workshop on Compilers for Parallel Computers, Delft*, 1993, 428–435.

MODELLING TRANSMISSION LINES EFFECTS IN INTEGRATED CIRCUITS BY A MIXED SYSTEM OF DAEs AND PDEs

M. Günther
Technische Hoschule Darmstadt, Fachbereich Mathematik
Schlossgartenstr. 7, D-64289 Darmstadt
e-mail: `guenther@mathematik.th-darmstadt.de`

Abstract. To model transmission lines effects in integrated circuits, we couple the network equations for the circuits with the telegrapher's equations for the transmission lines; this results in an initial/boundary value problem for a mixed system of DAEs and hyperbolic PDEs. By semidiscretization the system is transformed into differential-algebraic equations in time only. We apply this modeling approach to a CMOS ring oscillator, an oscillatory circuit with transmission lines as coupling units, and discuss the simulation results.

1 Introduction

Modeling and numerical simulation of coupled problems is a main task in today's engineering applications, for example, in microsystem technology [4] or multibody system dynamics [13]. In the design of integrated circuits, second order effects become more and more important with increasing integration rates. One example is the treatment of thermal noise in semiconductor devices. To model these effects, noise sources are added to the network: the model description in form of differential-algebraic equations is shifted to stochastic differential-algebraic equations with additive noise [5].

Transmission lines effects such as signal delay, reflection, attenuation, dispersion and crosstalk may yield misfunctions in high speed digital circuits. Due to the highly nonlinear behaviour of MOS transistors a time domain analysis is recommended. Many circuit simulation packages treat these effects within the framework of standard circuit theory: transmission lines are modeled by equivalent circuits of lumped elements, generally coupled RLC elements and controlled sources [11]. However, estimates are missing for the modeling error, which is caused by using companion models.

We consider an alternative approach: by coupling the telegrapher's equations for the transmission lines with the network equations, all system information can be used. The network equations at the coupling nodes define boundary values for the telegrapher's equations in differential-algebraic form. Together with appropriate initial values, this defines an initial/boundary value problem for a mixed system of DAEs and hyperbolic PDEs.

We discuss this modeling approach in the next section and show how to transform the resulting mixed system into a DAE system in time only. The CMOS ring oscillator, an oscillatory circuit with transmission lines as coupling units, will serve as an example. The simulation results show that its behaviour strongly depends on the properties of the transmission lines.

2 Modeling approach for networks with transmission lines

Coupling of networks and transmission lines system. A system of n coupled uniform lossy transmission lines shown in fig. 1 can be characterized by the telegrapher's equations

$$-\frac{\partial V(z,t)}{\partial z} = L'\frac{\partial J(z,t)}{\partial t} + R'J(z,t), \tag{1a}$$

$$-\frac{\partial J(z,t)}{\partial z} = C'\frac{\partial V(z,t)}{\partial t} + G'V(z,t), \tag{1b}$$

where R', L', G' and $C' \in \mathbb{R}^{n\times n}$ are the resistance, inductance, conductance and capacitance matrices per unit length. $V(z,t)$ is an n dimensional vector of line voltages with respect to ground, and $J(z,t)$ is an n dimensional vector of line currents. This first order hyperbolic system of partial differential equations is initialized by a set of initial values

$$V_i(z,t_0) = V_i^0(z) \qquad \forall z \in [0,L], \quad i = 1,\ldots,n, \tag{2a}$$

$$J_i(z,t_0) = J_i^0(z) \qquad \forall z \in [0,L], \quad i = 1,\dots,n \tag{2b}$$

at time t_0. Both voltages and currents are fixed at the end of the lines by $4n$ boundary conditions

$$u_1 := (u_{1,0},\dots,u_{n,0})^T - V(0,t) = 0, \qquad I_1 := (I_{1,0},\dots,I_{n,0})^T - J(0,t) = 0, \tag{3a}$$

$$u_2 := (u_{1,L},\dots,u_{n,L})^T - V(L,t) = 0, \qquad I_2 := (I_{1,L},\dots,I_{n,L})^T + J(L,t) = 0, \tag{3b}$$

which connect the lines system with electrical networks via controlled voltage sources, see fig. 1.

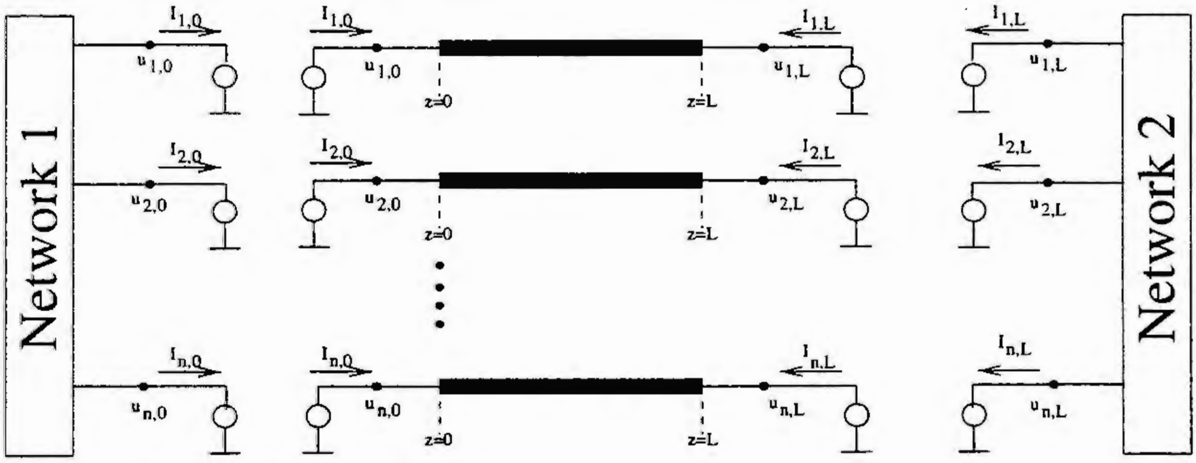


Figure 1: System of n coupled transmission lines, connecting network 1 and 2

Using conventional modified nodal analysis [10], the networks are described by differential-algebraic equations of quasilinear-implicit form:

$$\begin{aligned} C_1(x_1)\cdot \dot{x}_1 + f_1(x_1,t) &= 0, && (4a)\\ C_2(x_2)\cdot \dot{x}_2 + f_2(x_2,t) &= 0, && (4b)\end{aligned}$$

where the generally singular matrices C_1 and C_2 contain voltage and current dependent capacitances and inductances. The nonlinear functions f_1 and f_2 describe the static part of the system. The unknowns x_i consist of the inner node potential and currents through voltage-defining elements of network i, denoted by x_i^I, and the coupling voltages and currents x_i^C of network i. The initial values $x_i(t_0)$ must be consistent not only with (4), but also with the initial conditions in (2). System (1–4) describes an initial/boundary value problem for a mixed system of DAEs and hyperbolic PDEs. Note that the boundary conditions (3) become differential-algebraic equations, if the coupling variables are replaced by the corresponding formulas of (4) where possible.

Space discretization and boundary conditions. To reduce (1) to a problem with t as the only independent variable, time and space are separated by the standard Ritz ansatz [2]

$$V(z,t) = \psi_0(z)u_1(t) + \sum_{j=1}^{N} \psi_j(z)p_j(t) \tag{5a}$$

$$J(z,t) = \varphi_0(z)I_1(t) + \sum_{j=1}^{N} \varphi_j(z)q_j(t) \tag{5b}$$

with a suitable set of ansatz functions $\psi := (\psi_1,\dots,\psi_N)^T$, $\varphi := (\varphi_1,\dots,\varphi_N)^T$ and unknown coefficient vectors $p_j, q_j, j = 1,\dots,N$. We use finite elements which satisfy the boundary conditions (3) at the left end of each line, i. e.

$$\psi_0(0) = \varphi_0(0) = 1, \qquad \psi(0) = \varphi(0) = 0,$$

and are element of some Sobolev space $H^m([0,L])$.

The weak formulation of the boundary value problem (1–3) then yields the initial value problem

$$\underbrace{\begin{pmatrix} L'\otimes M_{\psi,\varphi} & 0 \\ 0 & C'\otimes M_{\psi,\varphi}^T \end{pmatrix}}_{\mathcal{M}:=}\begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix}+\underbrace{\begin{pmatrix} R'\otimes M_{\psi,\varphi} & -\mathrm{Id}\otimes K_\psi \\ -\mathrm{Id}\otimes K_\varphi & G'\otimes M_{\psi,\varphi}^T \end{pmatrix}}_{\mathcal{K}:=}\begin{pmatrix} q \\ p \end{pmatrix}+$$

$$+\begin{pmatrix} (L'\dot{I}_1+R'I_1)\otimes b_{\varphi,\psi} \\ (C'\dot{u}_1+G'u_1)\otimes b_{\psi,\varphi} \end{pmatrix}-\begin{pmatrix} u_1\otimes b_{\psi,\psi'} \\ I_1\otimes b_{\varphi,\varphi'} \end{pmatrix}-\begin{pmatrix} u_1\otimes\psi(0)-u_2\otimes\psi(L) \\ I_1\otimes\varphi(0)+I_2\otimes\varphi(L) \end{pmatrix}=0, \qquad (6)$$

where we have used the abbreviations

$$M_{\psi,\varphi}=\int_0^L \psi\varphi^T\,dz, \quad K_\psi=\int_0^L \psi'\psi^T\,dz, \quad K_\varphi=\int_0^L \varphi'\varphi^T,$$

$$b_{\varphi,\psi}=\int_0^L \varphi_0\psi\,dz, \quad b_{\psi,\varphi}=\int_0^L \psi_0\varphi\,dz, \quad b_{\psi,\psi'}=\int_0^L \psi_0\psi'\,dz, \quad b_{\varphi,\varphi'}=\int_0^L \varphi_0\varphi'\,dz,$$

and $'$ denotes the partial derivative with respect to z. The remaining boundary conditions, which are not automatically fulfilled by the ansatz functions, are added separately as algebraic equations:

$$u_2(t)-(\psi_0(L)u_1(t)+\sum_{j=1}^N \psi_j(z)p_j(t))=0 \quad \text{and} \quad I_2(t)+(\varphi_0(L)I_1(t)+\sum_{j=1}^N \varphi_j(z)q_j(t))=0. \qquad (7)$$

System (4,6,7), together with consistent initial values, describes now an initial value problem for differential-algebraic equations. To solve this system, a variety of integration methods is available [1, 9]

Remark. All boundary conditions are essential and have to be fulfilled by the weak solution. However, the user has some freedom how to treat the boundary conditions: a boundary condition can be satisfied by the ansatz functions, or added as an algebraic relation to the system. For the important classes of linear and cubic Hermite elements the following holds: if two boundary conditions for each line are fulfilled by the Ritz ansatz, then the stiffness matrix $\mathcal{K}$ is regular. Otherwise $\mathcal{K}$ is singular. Regularity of the stiffness matrix is essential for two reasons: in the case of a singular mass matrix $\mathcal{M}$, regularity of $\mathcal{K}$ guarantees the well-posedness of the DAE system to be solved, i. e. the matrix pencil of the linearized system is regular. Additionally, the operating point analysis requires regularity of $\mathcal{K}$ [6].

3 An example: CMOS ring oscillator with transmission lines

The modeling approach discussed in the last section is now applied to a ring oscillator with three CMOS inverters connected by transmission lines, see fig 2. The companion model for a CMOS inverter is shown in fig. 3, left. Here the inverter is modeled by two enhancement MOS transistors of p and n type with a load capacitance C. The input signal is transferred to both transistors. To model the MOS transistor, we use the companion model due to Shichman and Hodges [12] in fig 3, right. Ring oscillators serve as benchmarks for simulation tools in the stability analysis of oscillatory circuits, see e. g., [3, 10, 14].

Semidiscretized model. If we use m cubic Hermite elements for space discretization, then the modeling approach of section 2 yields a DAE system with $36+2\cdot N$ with $N=3*(2\cdot m+1)$ unknowns:

- 6 node voltages $u_1,\ldots,u_6$ and branch currents $I_1,\ldots,I_6$ at the 6 coupling nodes,
- 24 inner node voltages $u_7,\ldots,u_{30}$,
- the N dimensional line voltage vector

  $$p=(p_{1,1},\ldots,p_{1,2*m+1},p_{2,1},\ldots,p_{2,2*m+1},p_{3,1},\ldots,p_{3,2*m+1})^T$$

  consisting of the voltages at the $m+1$ grid points and its derivatives. Note that the line voltage at the left end of each line is not included due to ansatz (5a).
- the N dimensional line current vector

  $$q=(q_{1,1},\ldots,q_{1,2*m+1},q_{2,1},\ldots,q_{2,2*m+1},q_{3,1},\ldots,q_{3,2*m+1})^T$$

  consisting of the line currents at the $m+1$ grid points and its derivatives. Note that the line current at the left end of each line is not included due to ansatz (5b).

Whereas $2N$ differential equations are defined by system (6), the remaining 36 relations come from the network equations of the three CMOS inverters and the remaining 6 boundary conditions.

The resulting DAE system has index 2: there are loops of capacitors and voltage sources, caused by the capacitors of the CMOS inverter model and the voltage sources, which connect the inverters with the transmission lines [7]. However, only the branch currents through the voltage sources are index 2 variables. All node potentials and line voltages are at most index 1.

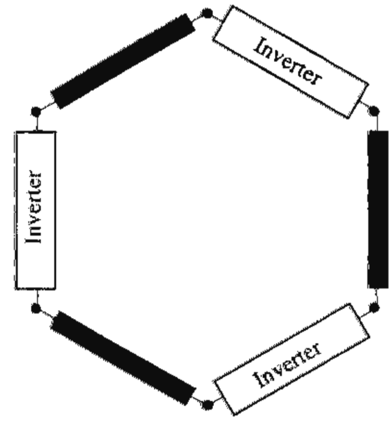


Figure 2: CMOS ring oscillator with three inverters connected by transmission lines

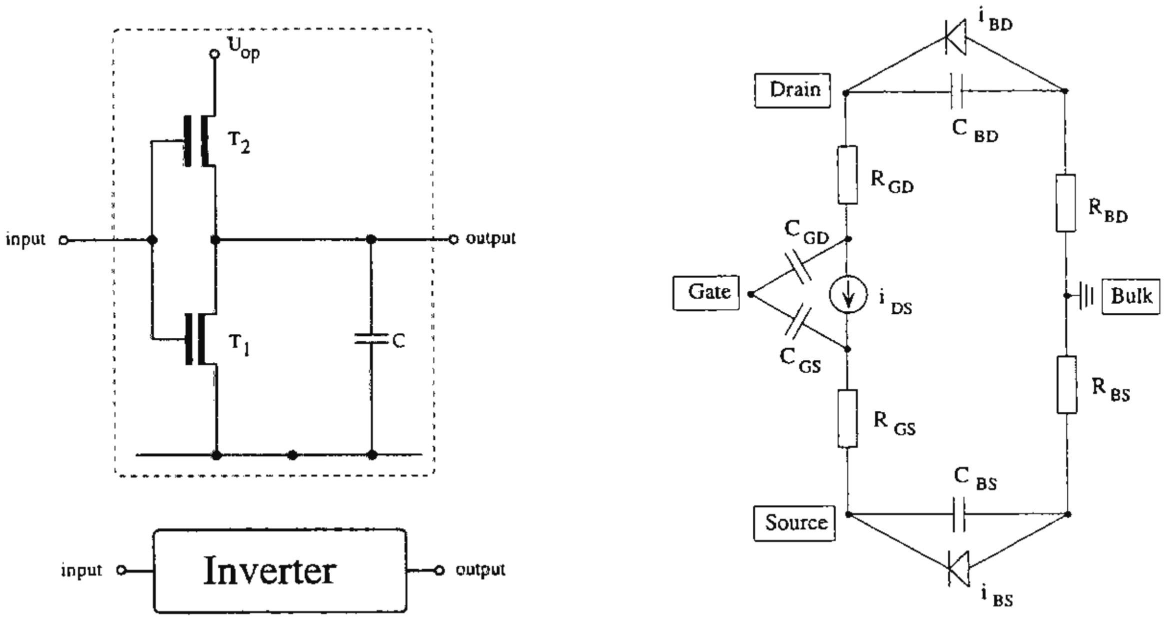


Figure 3: Network model for a CMOS inverter (left) and MOS transistor companion model due to Shichman and Hodges (right)

Simulation results. For the time integration, the code RODAS of [9], a fourth-order L-stable Rosenbrock-Wanner method, was applied to an index-1 implementation. At lower tolerances, high frequency modes, which are introduced by a refined space discretization [13], are damped by RODAS.

Four finite elements are used for each transmission line, which results in a system of 90 state variables. If the transmission lines effects are neglected, we get the stable limit cycle [3] shown in fig 4. Here, of course, all line voltages are equal for each line.

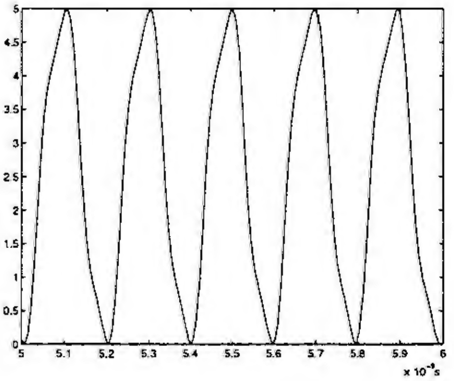


Figure 4: Voltages $u_1 = u_2$ at the first transmission line with $C' = L' = R' = G' = 0$.

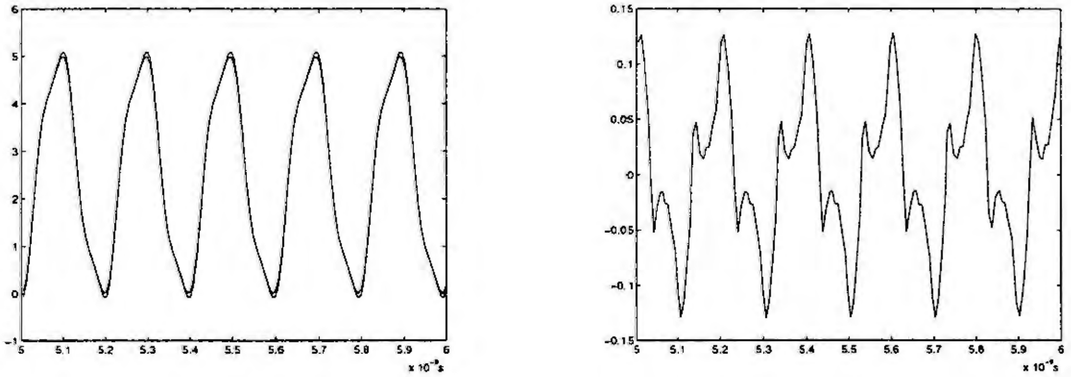


Figure 5: Voltages u_1 and u_2 at the first transmission line (left) with $C' = 10^{-14}\,\text{F/m}$, $L' = 10^{-9}\,\text{H/m}$, $R' = 10^{-1}\,\Omega/\text{m}$, $G' = 0$, and voltage drop $u_1 - u_2$ (right)

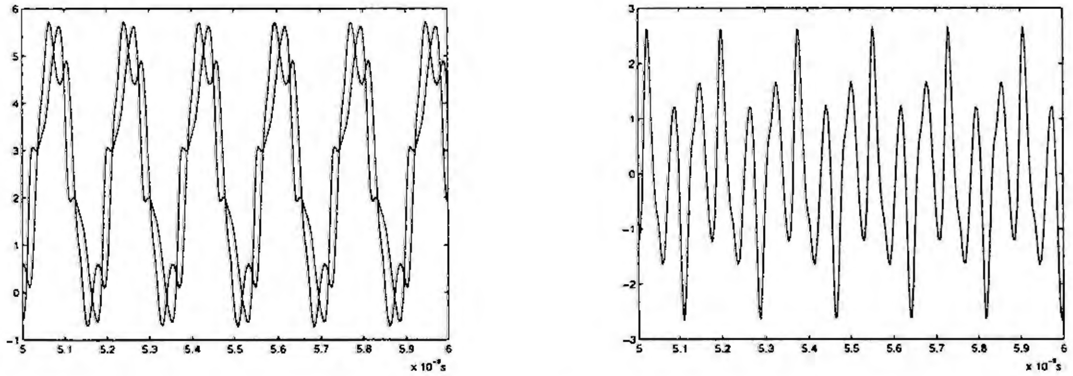


Figure 6: Voltages u_1 and u_2 at the first transmission line (left) with $C' = 5{\cdot}10^{-14}\,\text{F/m}$, $L' = 5{\cdot}10^{-9}\,\text{H/m}$, $R' = 10^{-1}\,\Omega/\text{m}$, $G' = 0$, and voltage drop $u_1 - u_2$ (right).

For moderate transmission lines parameters, the oscillations are only slightly perturbed, see fig 5: the voltage drop between the end of the lines oscillates with roughly the same period and an amplitude of about 0.12 V. Increasing both inductance and capacitance per unit length by a factor of 5 (see fig. 6), amplitude and period increase, and the voltage curves change. The signal delay is superimposed by high oscillations, visible by the voltage drop $u_1 - u_2$ between both ends of the line.

4 Conclusions

Transmission lines effects in integrated circuits are often modeled by a network companion model, which transforms the problem to a problem of electrical network simulation [11]. Another approach is to use independent models for the network and the transmission lines, and solve the joint problem by an outer iteration for the coupling and by arbitrary inner solution processes for each single problem [4]. The joint model discussed in this paper aims at describing the joint behaviour more appropriate: the physical coupling of transmission lines and electrical circuits is reflected by a coupled model: a mixed system of DAEs and hyperbolic PDEs. To derive more appropriate models, nonlinear and frequency dependent effects of transmission lines are to be taken into consideration.

Acknowledgment. The author thanks U. Feldmann and P. Rentrop for many helpful discussions.

References

[1] K. E. Brenan, S. L. Campbell and L. R. Petzold, Numerical Solution of Initial-Value Problems in Differential–Algebraic Equations. SIAM, Philadelphia, 1996.

[2] Brenner, S. C. and Scott, L. R., The Mathematical Theory of Finite Element Methods. Springer-Verlag, New York, 1994.

[3] Bulirsch, R., Selting, P. A., Feldmann, U. and Zheng, Q., Optimale Systeme in der Mikroelektronik — Stabilität von Oszillatorschaltungen. To appear in: Mathematik — Schlüsseltechnologie für die Zukunft, Springer-Verlag, Berlin.

[4] Bungartz, H.-J. and Schulte, S., Coupled Problems in Microsystem Techonology. In: Numerical Treatment of Coupled Systems — Proceedings of the Eleventh GAMM-Seminar Kiel January 20–22,1995, (Eds.: Hackbusch, W. and Wittum, G.) Vieweg & Sohn Verlagsgesellschaft mbh, Braunschweig/Wiesbaden, 1995, 11–24.

[5] Denk, G. and Schäffler, S., Adams Methods for the Efficient Solution of Stochastic Differential Equations with Additive Noise. To appear in Computing.

[6] Feldmann, U., Wever, U., Zheng, Q., Schultz, R. and Wriedt, H., Algorithms for modern circuit simulation. AEÜ, 46 (1992), 274–285.

[7] Günther, M. and Feldmann, U., The DAE-index in electric circuit simulation, Mathematics and Computers in Simulation, 39 (1995), 573–582.

[8] Günther, M. and Rentrop, P., Multirate ROW methods and latency of electric circuits, Appl. Numer. Math., 13 (1993), 83–102.

[9] Hairer, E. and Wanner, G., Solving ordinary differential equations II — Stiff and differential-algebraic problems, Springer-Verlag, Berlin, 1991.

[10] Kampowsky, W., Rentrop, P. and Schmidt, W., Classification and numerical simulation of electric circuits. Surv. Math. Ind., 2 (1992), 23–65.

[11] Ruehli, A. E., Circuit analysis, simulation and Design — Part 2. North-Holland, Amsterdam, 1987.

[12] Shichman, H. and Hodges, D. A., Insulated-gate field-effect transistor switching circuits. IEEE J. Solid State Circuits, SC-3 (1968), 285–289.

[13] Simeon, B., Modeling a Flexible Slider Crank Mechanism by a Mixed System of DAEs and PDEs. Mathematical Modeling of Systems, 2 No. 1 (1996), 1–18.

[14] Zheng, Q. and Dellnitz, M.: Schwingungen eines Ringoszillators — eine numerische Behandlung unter Berücksichtigung von Symmetrien. Z. f. angew. Math. u. Mech., 70 No. 4 (1990), T135–T138.

AN IMPROVED IRON LOSS MODEL FOR ROTATING ELECTRICAL MACHINES

L.R. Dupré * R. Van Keer ** J.A.A. Melkebeek *

* Department of Electrical Power Engineering, University of Gent,
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium

** Department of Mathematical Analysis, University of Gent,
Galglaan 2, B-9000 Gent, Belgium

Abstract. In this paper we deal with a mathematical model for the evaluation of the electromagnetic iron losses in rotating electrical machines under no load conditions. This model is based on a two level machine model, i.e. first level: tooth region model, second level: lamination model. The presented problems of electromagnetic field computations are coupled with refined material models based on the Preisach theory. The model is validated by the comparison of numerical results and experimental values from measurements.

Introduction

Iron losses can account for a significant part of the total losses of an electrical machine. At the other hand, nowadays the efficient use of electricity is strongly emphasized. Electrical drive systems offer considerable opportunity to obtain major improvements in this respect. Consequently, it is important to increase the accuracy and reliability of the modelling and simulation of the iron losses.

A numerical model based on a single valued material characteristic cannot describe adequately the phase difference between the magnetic flux density $\bar{B}$ and the magnetic field strength $\bar{H}$ in the case of a rotating magnetic flux excitation. This type of excitation in electrical machines results from the complexity of the magnetic circuit and of the magnetic motoric force distributions.

In this paper we present the inclusion of the vector Preisach model, as described in [7], in the magnetic field calculations for a 2D-domain D. This domain D represents one tooth region of the stator of an asynchronous machine. The magnetic behaviour of the material can be described in terms of the macroscopic fields, taking into account the hysteresis phenomena.

The boundary ∂D is divided into six parts, namely 3 flux gates and 3 flux walls, as shown in Fig.1. The three enforced flux patterns through ∂D_1, ∂D_2 and ∂D_3 are obtained by numerical field calculations or by local measurements in the electrical machine.

On the basis of the computed field patterns in the domain D, the local excitation conditions for the magnetic material will be derived. The models described in [1] and [3] will be used to investigate the local material response. This will lead to a detailed knowledge of the local iron losses.

Finally, the global machine losses, evaluated from this new method, are compared with the measured machine losses.

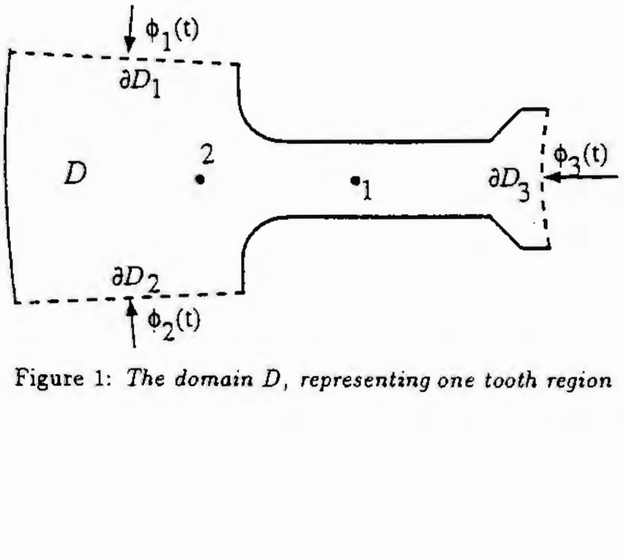


Figure 1: *The domain D, representing one tooth region*

Material models

Scalar hysteresis model

If $\bar{H}$ and $\bar{B}$ are *unidirectional*, the BH-relation can be described by a *scalar* Preisach model in which the material is assumed to consist of small dipoles, each being characterized by a rectangular hysteresis loop as shown in Fig., [9]. The magnetisation of the dipole is given by

$$M_d = \begin{cases} +1 & : H(t) > \alpha \text{ or } (\beta < H < \alpha \text{ and } H_{last} > \alpha) \\ -1 & : H(t) < \beta \text{ or } (\beta < H < \alpha \text{ and } H_{last} < \beta) \end{cases} \tag{1}$$

where H_{last} is the last extreem value of H kept in memory. The characteristic parameters α and β are distributed statistically according to a Preisach function $P_s(\alpha, \beta)$ which is a material parameter. This distribution function can he identified directly when using a proper measurement technique [4].

The BH-relation reads:

$$B(H, H_{past}) = \int_{-H_m}^{H_m} d\alpha \int_{-H_m}^{\alpha} d\beta\ \eta_s(\alpha, \beta, t) P_s(\alpha, \beta). \tag{2}$$

Here $\eta_s(\alpha, \beta, t)$ gives the value of the magnetisation M_d for the dipole with parameters α and β at time t. Consequently, the induction B depends upon the magnetic field $H(t)$ and its history, denoted by $H_{past}(t)$.

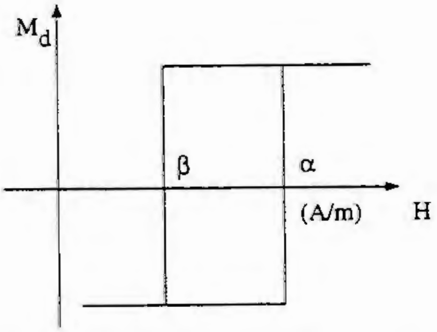


Figure 2: *(M_d,H)-characteristic of a Preisach dipole*

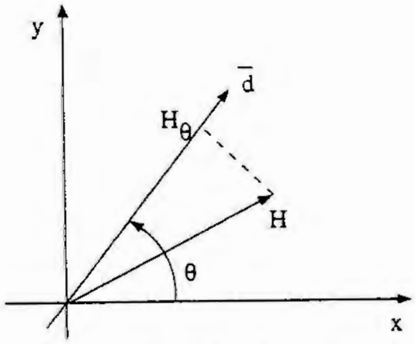


Figure 3: *Vector Preisach model*

Vector hysteresis model

In this model, as described in [7], the magnetic field vector $\bar{H}$ and the magnetic induction vector $\bar{B}$ are no longer unidirectional. The vector $\bar{H}$ is projected on an axis $\bar{d}$, which encloses an angle θ with the fixed x-axis, $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$, see Fig.. The resulting component $H_\theta (= H_x cos\theta + H_y sin\theta)$ is used as the input of the scalar Preisach model on the axis $\bar{d}$.

The $\bar{B}\bar{H}$-relation is now given by, see [8],

$$\bar{B}(\bar{H}, \bar{H}_{past}) = \frac{1}{\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} d\theta B_\theta(H_\theta, H_{past,\theta}) \bar{1}_\theta, \tag{3}$$

with

$$B_\theta(H_\theta, H_{past,\theta}) = \int_{-H_m}^{H_m} d\alpha \int_{-H_m}^{\alpha} d\beta\ \eta_r(\theta, \alpha, \beta, t) P_r(\alpha, \beta), \tag{4}$$

where $\eta_r(\theta, \alpha, \beta, t)$ is obtained from the component H_θ, and thus depends on $\bar{H}(t)$ and $\bar{H}_{past}(t)$. The Preisach function P_r in this rotational model can be evaluated from the distribution function P_s, entering (2), see [3].

Extension to a rate dependent hysteresis model

The frequency dependence of the hysteresis effects may have a large influence on the magnetic behaviour of the material as pointed out in [1]. Therefore, in that paper a *rate-dependent scalar* Preisach model has been incorporated in the magnetodynamic field calculations under arbitrary alternating excitation conditions. Here, the switching of the dipoles is no longer instantaneously, but proceeds at a finite rate. The main consequence of this improvement is the enlargement of the hysteresis loops with increasing frequency. This effect allows the modelling of the extra losses, appearing at increasing frequency, together with the classical eddy current losses.

Unfortunately, at present no experimental validated *rate-dependent vector* hysteresis model is available.

Two level machine model

First level: tooth region model

We consider the single tooth region of Fig.1, where the electrical conductivity σ is assumed to be zero. The relevant Maxwell equations for the magnetic field $\bar{H}=H_x\bar{1}_x + H_y\bar{1}_y$ and the magnetic induction $\bar{B} = B_x\bar{1}_x + B_y\bar{1}_y$, in the 2D domain D with boundary ∂D are, see e.g [6],

$$rot\bar{H} = 0, \tag{5}$$

$$div\bar{B} = 0, \tag{6}$$

where the relation between $\bar{H}$ and $\bar{B}$ is defined by the material characteristics obtained from the vector Preisach hysteresis model, described above.

Enforcing a total flux $\phi_s(t)$ through the parts ∂D_s, s=1,2,3, of ∂D, we arrive at the boundary conditions (BCs)

$$\phi_s(t) = \int_{\partial D_s} \bar{B}\cdot\bar{n}dl,\ t>0,\ s=1,2,3, \tag{7}$$

$$\bar{H}\text{x}\bar{n} = 0 \text{ on } \partial D_s,\ t>0,\ s=1,2,3, \tag{8}$$

where $\bar{n}$ is the unit outward normal vector to the boundary part ∂D_s.
At the other hand an assumed *zero flux leakage* through ∂D_4, ∂D_5 and ∂D_6 results in the additional BCs:

$$\bar{B}\cdot\bar{n} = 0 \text{ on } \partial D_s,\ t>0,\ s=4,5,6. \tag{9}$$

The *demagnetized* state of the material at $t=0$ is expressed by the initial condition (IC)

$$\bar{H}(x,y,t=0)=0, \begin{cases} \eta_r(x,y,\theta,\alpha,\beta,t=0)=+1 & :\alpha+\beta<0 \\ \eta_r(x,y,\theta,\alpha,\beta,t=0)=-1 & :\alpha+\beta>0 \end{cases} \forall(x,y)\in D, -\frac{\pi}{2}\le\theta\le\frac{\pi}{2}. \tag{10}$$

As the enforced fluxes $\phi_s(t)$, $s=1,2,3$, are periodic in time, we may use a complex Fourier decomposition for the local vector fields $\bar{H}(x,y;t)$ and $\bar{B}(x,y;t)$, viz

$$\bar{H}(x,y;t) \equiv \sum_{k=-\infty}^{+\infty} H_k(x,y)\cdot e^{j(k\omega t+\alpha_k)}, \tag{11}$$

$$\bar{B}(x,y;t) \equiv \sum_{k=-\infty}^{+\infty} B_k(x,y)\cdot e^{j(k\omega t+\beta_k)}. \tag{12}$$

Here, ω is 2π times the basic frequency; α_k [resp. β_k] and H_k [resp. B_k] are the phase angle and the amplitude of the k-th harmonic of $\bar{H}$ [resp. $\bar{B}$], see also [2].

Using the local field patterns obtained from the tooth region model, see (12), we may investigate the local material behaviour from a lamination model.

Second level: lamination model

The magnetic behaviour of ferromagnetic laminations can be described in terms of the macroscopic fields, taking into account the interacting hysteresis and eddy current phenomena.

We consider a single lamination of length l, width w and thickness $2d$, see Fig., the cartesian coordinate system being chosen in a natural way as indicated. Throughout the sheet, which is assumed isotropic, the time dependent total flux vector $\bar{\varphi}(t)$ flows parallel to the (x,y)-plane. This flux vector is constructed out of (12). The magnetic field and the magnetic induction in the lamination model take the form $\bar{H}=H_x\bar{1}_x+H_y\bar{1}_y$ and $\bar{B}=B_x\bar{1}_x+B_y\bar{1}_y$ respectively. As $d << w$ and $d << l$, eliminating the edge effects, we may assume H_x, H_y and B_x,B_y to vary in the z-direction only.

Next, we take into account the constitutive relation, $\bar{J} = \sigma\bar{E}$, between the electric field $\bar{E}$ and the current density $\bar{J}$ (both parallel to the (x,y)-plane) in the relevant Maxwell equations, viz

$$rot\bar{E} = -\frac{\partial\bar{B}}{\partial t}, \tag{13}$$

$$rot\bar{H} = \bar{J}. \tag{14}$$

Two different types of excitations can be considered.

a) Alternating excitation conditions

Here, we may assume H_y, B_y and $\varphi_y(t)$ to be identically zero. The equations above simplify to the parabolic DE for H_x

$$\frac{1}{\sigma}\cdot\frac{\partial^2 H_x}{\partial z^2} = \frac{\partial B_x}{\partial t}, 0 < z < d, t > 0, \tag{15}$$

along with the BCs

$$\frac{\partial H_x}{\partial z}(z=0,t) = 0, \frac{\partial H_x}{\partial z}(z=d,t) = \frac{\sigma}{2}\frac{d\varphi_x}{dt}, t > 0. \tag{16}$$

and ICs

$$H_x(z,t=0) = 0, \begin{cases} \eta_s(z,\alpha,\beta,t=0) = +1 & : \alpha+\beta < 0 \\ \eta_s(z,\alpha,\beta,t=0) = -1 & : \alpha+\beta > 0 \end{cases}, 0 < z < d. \tag{17}$$

Here, the magnetic induction $B_x(z,t)$ can be related to the magnetic field $H_x(z,t)$ by either the *scalar* rate independent *or* rate dependent Preisach hysteresis model. A finite element - finite difference approximation method for the BVP (15)-(16)-(17) is described in detail in [1].

b) Rotational excitation conditions

Now, the governing differential equations for the magnetic field (H_x,H_y) are found to be

$$\frac{1}{\sigma}\cdot\frac{\partial^2 H_x}{\partial z^2} = \frac{\partial B_x}{\partial t}, 0 < z < d, t > 0, \tag{18}$$

$$\frac{1}{\sigma}\cdot\frac{\partial^2 H_y}{\partial z^2} = \frac{\partial B_y}{\partial t}, 0 < z < d, t > 0, \tag{19}$$

while the BCs become

$$\frac{\partial H_x}{\partial z}(z=0,t) = \frac{\partial H_y}{\partial z}(z=0,t) = 0, \frac{\partial H_x}{\partial z}(z=d,t) = \frac{\sigma}{2}\frac{d\varphi_x}{dt}, \frac{\partial H_y}{\partial z}(z=d,t) = \frac{\sigma}{2}\frac{d\varphi_y}{dt}, t > 0. \tag{20}$$

The ICs, again describing the demagnetized state at $t = 0$, are now given by

$$H_x(z,t=0) = 0,\ H_y(z,t=0) = 0, \begin{cases} \eta_r(z,\theta,\alpha,\beta,t=0) = +1 & : \alpha+\beta < 0 \\ \eta_r(z,\theta,\alpha,\beta,t=0) = -1 & : \alpha+\beta > 0 \end{cases} -\frac{\pi}{2} < \theta < \frac{\pi}{2}, 0 < z < d. \tag{21}$$

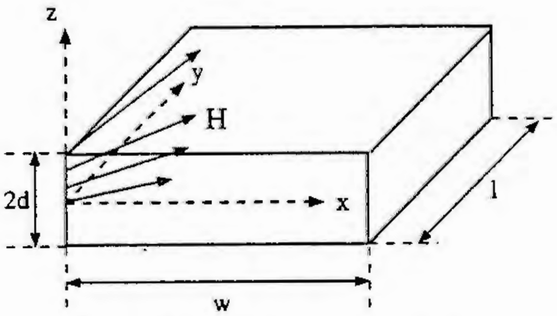


Figure 4: *Magn. model of one lamination*

Here, the magnetic induction $\bar{B}$ is related to the magnetic field $\bar{H}$ by the *vector* rate independent Preisach hysteresis model, introduced in [7]. In [3] we dealed with a modified finite element - Crank Nicholson method to solve numerically the resulting BVP.

Notice that, due to the complexity of the material model used, (15) and (18)-(19) are highly nonlinear partial differential equations with memory.

The total electromagnetic losses in the lamination per unit volume during a time interval $[T_1,T_2]$ (where T_2-T_1 is an integer multiple of the excitation period) are calculated by summing up the hysteresis losses and the eddy current losses. These losses are given by, see e.g. [6],

$$P_h = \frac{1}{2d}\int_{-d}^{d} dz \int_{T_1}^{T_2} (H_x \frac{\partial B_x}{\partial t} + H_y \frac{\partial B_y}{\partial t})dt \tag{22}$$

and

$$P_e = \frac{1}{2d\sigma}\int_{-d}^{d} dz \int_{T_1}^{T_2} \left(\left(\frac{\partial H_x}{\partial z}\right)^2 + \left(\frac{\partial H_y}{\partial z}\right)^2\right) dt. \tag{23}$$

Numerical results

Tooth region model

We consider a three phase 3kW 4-pole induction motor, described in detail in [5]. Due to periodicity, only three neighbouring teeth in the stator must be considered. The enforced total fluxes ϕ_s through the parts ∂D_s, $s = 1,2$ in Fig.1 are obtained from local measurements in the electrical machine (notice that ϕ_3=$-\phi_1 - \phi_2$). The flux through the gate ∂D_1 for each of the 3 neighbouring teeth is given in Table 1 by its Fourier decomposition

$$\phi_1(t) = \sum_k A_k cos(k\omega t + \gamma_k). \tag{24}$$

Table 2 shows the symmetry for each pair of positive and negative harmonics for point 1 in Fig.1 for tooth1. This corresponds to alternating field vectors. Notice that for point 2 this symmetry is lost, reflecting a rotational magnetic induction $\bar{B}$.
Similar remarks could be made for each point in tooth1, tooth2 and tooth3.

	tooth1		tooth2		tooth3	
k	A_k (Wb)	γ_k (°)	A_k (Wb)	γ_k (°)	A_k (Wb)	γ_k (°)
1	0.016625	25.16	0.016697	5.9	0.016631	-13.11
15	0.000234	109.12	0.000089	-154.59	0.000218	-56.71
17	0.000138	-36.28	0.000064	27.06	0.000104	111.11
31	0.000055	-48.45	0.000038	56.72	0.000055	166.34
33	0.000052	157.46	0.000019	-109.78	0.000050	-13.28

Table 1: *Local fluxpatterns through* ∂D_1

k	B_k (T) /point 1	B_k (T) /point 2
-33	7.2474815e-3	6.6953166e-3
-31	9.4345687e-3	6.5405699e-3
-17	1.5741827e-2	1.2912445e-2
-15	3.2655213e-2	1.7600928e-2
-5	1.6068061e-3	2.9070139e-2
-3	3.0426126e-3	2.1086182e-2
-1	0.6948881	8.8371411e-2
0	6.5151327e-3	8.0340728e-3
1	0.7042491	0.8502357
3	2.7993797e-3	5.2110258e-2
5	1.5675295e-3	1.9524982e-2
15	3.2812487e-2	2.5838170e-2
17	1.5398989e-2	7.4374103e-3
31	9.5228665e-3	5.9215301e-3
33	7.1590291e-3	2.5463994e-3

Table 2: *B-spectrum point 1 and point 2*

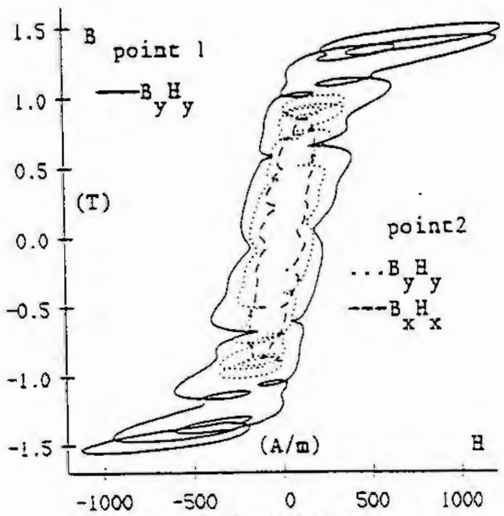


Figure 5: *local BH-loops*

Lamination model

The local flux patterns obtained form the tooth model are used as input for the magnetodynamic model that uses a *vector rate independent* Preisach model to account for rotational effects. The resulting *BH*-loops are shown in Fig.5 for the points 1 and 2 in tooth1.

To evaluate the local losses, we *add* to the losses calculated from the model (18)-(21) *extra* dynamic electromagnetic losses to take into account the *rate dependent* hysteresis effects. These extra dynamic losses represent the difference between the losses evaluated from the model (15)-(17) that is coupled first with the rate independent and next with the rate dependent Preisach model. Here, the alternating excitation used is the excitation obtained when we project the rotating excitation on that axis that gives rise to the maximum amplitude.

Machine losses

The global machine losses (65W), predicted by this combined tooth region- lamination model, may be compared with the machine losses (72W), measured in [5].

Acknowledgement

This research was carried out in the frame of the Inter-university Attraction Poles for fundamental research funded by the Belgian State and with the financial help of the Ministry of Economy of the Flemisch Governement in collaboration with OCAS, the research centre of ARBED-SIDMAR ALZ.

References

1. Dupré L., Van Keer R., Melkebeek J., On a Magnetodynamic Model for the Iron Losses in non-oriented Steel Laminations. Journal of Physics: applied physics, 29 (1996), 855-861.
2. Dupré L., Van Keer R., Melkebeek J., A 2D Finite Element Method for Magnetic Analysis using a Vector Hysteresis Model. Journal of Mathematical Problems in Engineering, (accepted).
3. Dupré L., Van Keer R., Melkebeek J., Magnetodynamic Field Computations using a Vector Preisach Model. In: Proc. ECCOMAS Conference Numerical methods in Engineering, Paris, 1996, John Wiley & Sons, New York, 1996, 312-317.
4. Dupré L., Electromagnetic characterisation of non-oriented electrical steel (in dutch), Phd-disertation, University of Gent, 1995.
5. Gyselinck J., Dupré L., Vandevelde L., Melkebeek J., Calculation of iron losses in non-oreinted Steel Laminations. In: Proc. 3rd International Workshop on Electric and Magnetic Fields, Liege, 1996, 423-428.
6. Halliday and Resnick, Fundamentals of Physics. John Wiley & Sons, New York, 1981.
7. Mayergoyz I.D., Mathematical models of hysteresis. Springer Verlag, New York, 1991.
8. Mayergoyz I.D. and Friedman G., Isotropic vector Preisach model of hysteresis. Journal of Applied Physics, 61 (1987), 4022-4024.
9. Preisach F., Uber the magnetische nachwirkung. Zeitschrift für Physik, 94 (1935), 277-302.

PREDICTION IN POWER SYSTEMS WITH NON STATIONARY LOAD PATTERNS

K. Voigtländer and H.-H. Wilfert
Fraunhofer-Institut für Informations- und Datenverarbeitung (IITB) Karlsruhe
Einrichtung für Prozeßsteuerung (EPS) Dresden
Zeunerstr. 38, D-01069 Dresden

Abstract. A hybrid pattern algorithm is presented combining statistical and neuronal methods to forecast hourly load of an electrical power supplying system. The load characteristics of some selected reference days are corrected with the help of a Σ-Π-Network according to changes of external influences like temperature. These rectified reference patterns are used within a radial basis network to shape a prediction. An attached statistical continuation- and error correction algorithm forms the final prediction. Especially the choice of appropriate model structures is considered. Furthermore a performance measurement of prediction accuracy of the four individual steps of the forecasting algorithm is presented.

Introduction

It has been proved that short term load forecasting plays an important role in operation planning of power systems. This paper deals with the prediction of the hourly electrical power demand for a forecast period of 24 hours [1,6]. The data comes from a regional power utility whose customs area is subject to strong structural changes. So the forecast algorithm has to go along with a small data base only. For that reason one has to be parsimonious with the number of free model parameters. On the other hand the model must be able to cover a wide variety of functional links. So a prediction tool is presented combining two neuronal parts for pattern correction and synthesis and two statistical model parts for error correction. The algorithm has been proved to be robust and effective.

The following part of this paper gives an overview of the whole algorithm and a detailed description of the four parts. Afterwards a comprehensive analysis of the performance is presented.

The prediction algorithm

Fig.1 shows the complete forecasting algorithm. Corresponding to the present forecasting period four reference load data sets were chosen from the data basis with respect to the same day type and special events. So the reference load patterns come mostly from the last four weeks. Each pattern consists of the hourly data from the 24 previous and the next 24 hours data according to current time.

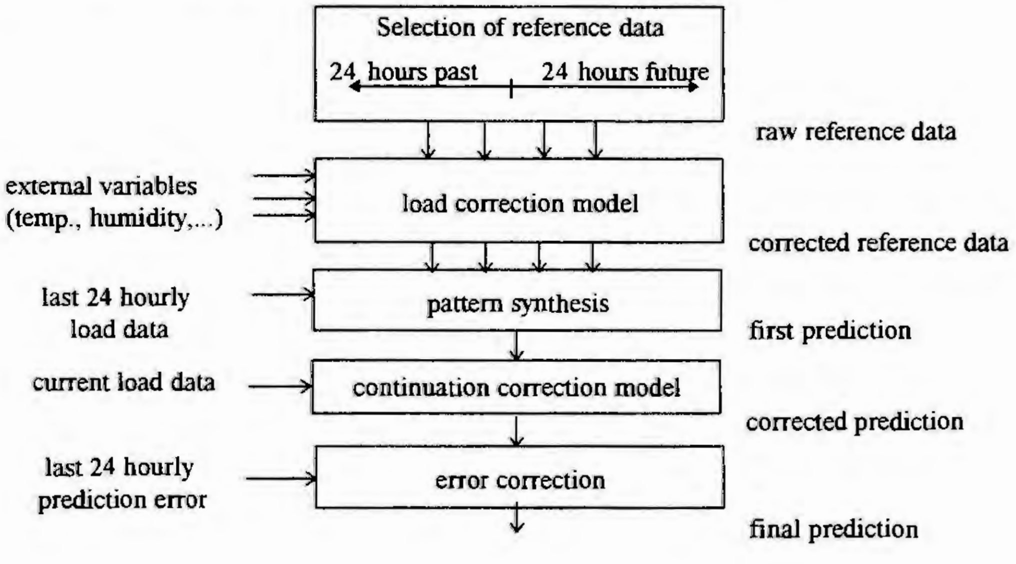


Fig. 1 The complete forecasting algorithm

With the help of the load model these reference data will be corrected with respect to the surrounding conditions (e.g. expected temperatures, wind, and humidity) of the forecasting period. So external influences will be eliminated from the reference data. Then the 24 hours part of the reference data representing the past will be judged with respect to its similarities to the real last 24 hour load. The 24 hours ahead part of the selected similar references will then be dominant in forming the 24 hours prediction. The difference between current true load and current predicted load is then used to do a weighted correction of the prediction. Finally a first order autoregressive model for the remaining forecast error is implemented based on the assumption that the former prediction error is correlated with the error of the current prediction [5].

The load correction model

The purpose of the load correction model is an estimation of load corrections (ΔP) due to changes of external variables (e.g. temperatures, wind or humidity) from reference day conditions to expected forecast day conditions. With the help of these hourly load corrections, the reference data will be adapted to forecast day conditions.

Because of the existing dynamical relationship between external variables and load consumption a wide variety of inputs might be necessary. So one problem is to select the appropriate variables and delays to apply as model inputs. Another problem comes from the expansion of the model function. Using the common approach of composing a superposition of some regressors φ_i

$$\underline{\hat{y}} = f(\underline{x}) = \sum_{i=1}^{N} c_i \underline{\varphi}_i(\underline{x}) \quad , \qquad (1)$$

the question arises how to choose properly the regressors forming a basis for the model output $\underline{\hat{y}}$. To avoid problems of non-linear optimization that usually occur in neural network appliance, an alternative approach is used. Instead of parametrizing some regressors (e.g. input weights of a sigmoidal MLP) a fixed pool of regressors is composed from which suitable ones have to be picked out to form the model. So the task of selecting proper inputs and constructing suited signal-links can be treated in a common framework.

One might apply well known regularization techniques to solve the resulting structural selection problem [7]. Unfortunately this is very difficult in presence of strong disturbed data and the regularization fails when almost linear depending regressors occur. The algorithm will not recognize the regressors as unnecessary. Realizing this an alternative easy to understand method, the orthogonal forward regression [3] is proposed. The basic idea is to extend the number of regressors forming the model base step by step and to trace the model error over a test data set. To find the best possible model while selecting k from N regressors a lot of combinations must be tested. So a hard computational work might be necessary. For that reason a pseudo optimal algorithm is applied which proved to be successful. The idea is to extend the model always with this regressor, which points best in the direction of the current model error. After moving this regressor from the pool into the model basis, the error must be updated and the remaining regressor pool must be orthogonalized with respect to the chosen regressor. So every step will extend the model with the momentary best regressor. Tracing the error over a test data set will give very useful information about optimal model size and essential signal links within the model.

Applying this algorithm to the load data will yield the following. Extensive studies have shown that four temperatures have an essential influence on load consumption (fig.2).

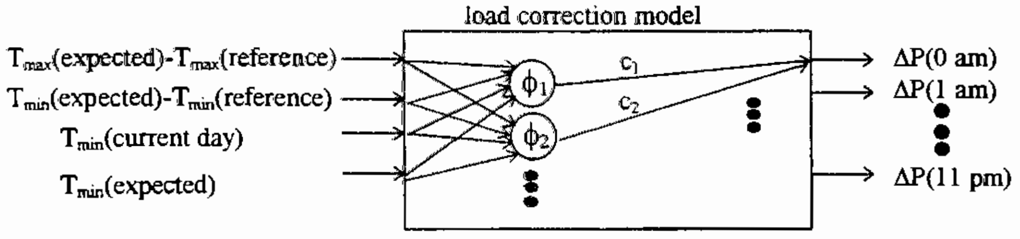


Fig. 2 Employed load correction model

For simplicity a polynomial expansion (Σ-Π-Network [2]) of the model nonlinearities is chosen to form the regressor pool. Thus the regressors ϕ_i are products of powers of the inputs. Tests with normalized Gaussian approximators (RBF-Networks) centred according to the reference data revealed similar results.

Data, collected over one year, with 4 references a day yield about 1400 data sets to structure and parametrize the model. The data was divided randomly into 700 parametrizing data and 700 test data to provide a cross-validation [4]. The regressor pool was formed by a 5 degree polynomial expansion of the 4 inputs providing 126 regressors. Fig. 4 shows the dependence of the ΔP(3 am)-model quality

$$Q = -20\log\left(\frac{\|\hat{\underline{y}} - \underline{y}\|_2}{\|\underline{y}\|_2}\right) \quad . \qquad (2)$$

on the number of regressors forming the model. As expected the model quality increases monotonously with an increase of the model size. But only 7 regressors form the optimal model (fig.3). All additional free parameters will be used to enclose data disturbances into the model.

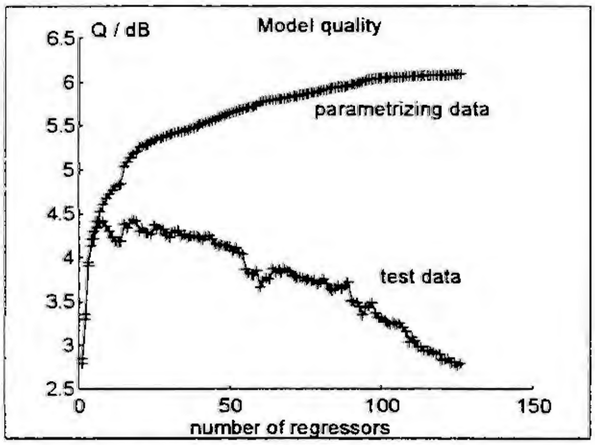


Fig. 3
Model quality in relation to model size

It is very interesting to remark that the full model with 126 regressors has about the same poor quality over the test data as the simplest model with only one regressor. A look at the 7 selected regressors shows that none of the inputs is redundant. So proper inputs, model size and suited links (in the sense of regressors) have been found.

The remaining 7 parameters can be calculated by a simple LMS-algorithm, alternatively adaptive with exponential decreasing weights.

Pattern synthesis

The four 48 hours (24 hours past and 24 hours future) reference pattern (corrected regarding temperatures and other influences) must be assembled into one 24 hours ahead prediction. To determine the weights δ^0 for an average procedure, the similarities between the 24-hours-past-part of the references P_p and the true last 24 hourly load P_t is evaluated by a normalized RBF-Network (fig.4) according to

$$\delta_i = \exp\left(-\sigma\left\|P_{pi} - P_t\right\|_2^2\right) \qquad i = 1,\ldots, 4 \quad . \qquad (3)$$
$$\delta_i^{\,0} = \delta_i \,/\, \Sigma\,\delta_i$$

With the help of these membership values a first 24 hours ahead prediction is computed by

$$P_{\text{predict}} = \Sigma\,\delta_i^{\,0} P_{fi} \qquad (4)$$

where P_{fi} means the 24 hours future part of the i-th reference data.

The only parameter to estimate is σ to assess the pattern similarities. A small σ always yields an equal averaging of the reference data, while a large value for σ causes a 'selection' of the nearest reference pattern for the prediction. Trying to avoid both effects σ is determined with respect to the variance of the patterns.

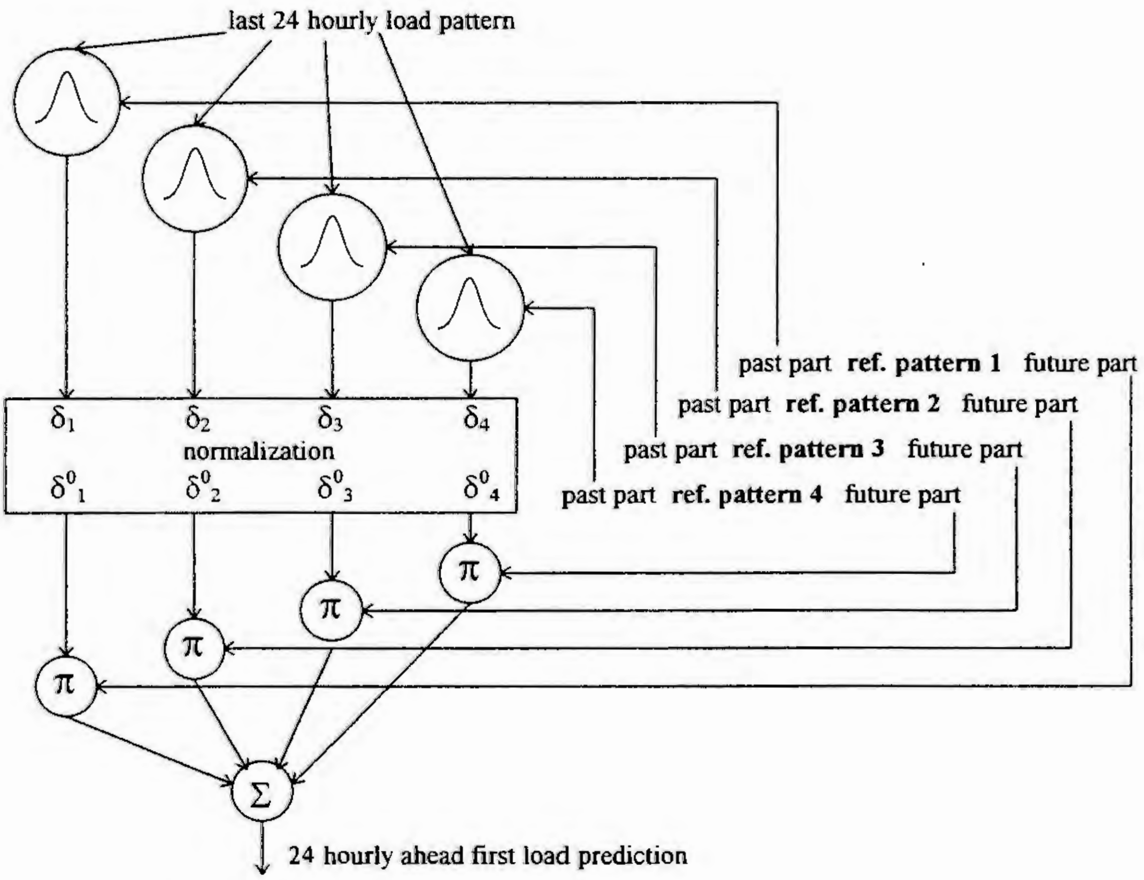

last 24 hourly load pattern
δ_1
δ_2
δ_3
δ_4
normalization
δ^0_1
δ^0_2
δ^0_3
δ^0_4
past part **ref. pattern 1** future part
past part **ref. pattern 2** future part
past part **ref. pattern 3** future part
past part **ref. pattern 4** future part
π
π
π
π
Σ
24 hourly ahead first load prediction

Fig. 4 Pattern synthesis

Continuation correction

There may be a load drift between the reference data and the current load data. Therefore the 24 hours prediction is extended to a prediction of the current load $P_{current\ predict}$. The difference between these prediction and the true current load $P_{current\ true}$ gives a basis for a continuation correction. According to the assumption that the load will not 'jump', the prediction is corrected by a weighted elimination of the estimated difference.

$$P_{correct}(i) = P_{predict}(i) - k_i(P_{current\ predict} - P_{current\ true}) \quad i=1,\dots,24 \tag{5}$$

The 24 parameters k_i can be calculated easily by an LMS-algorithm.

Parameter values of about 0.5 show that one half of the estimated difference is caused by drift and the other half seems to be random.

Error correction

An error correction algorithm is implemented to exploit possible correlation within the remaining prediction error. Based on the assumption of a non-white error series 24 first order moving average models are used and separately parametrized.

$$P_{final}(i) = P_{correct}(i) - a_i(P_{correct}(i-24) - P_{true}(i-24)) \quad i=1,\dots,24 \tag{6}$$

Again the 24 parameters can be estimated with respect to an equation error by a simple LMS-algorithm.

Performance analysis

The presented analysis bases on an average of hourly estimates over one year. Fig.5 shows the hourly mean load and the corresponding standard deviation of the load. Additionally the standard deviation of the simple reference day model (load pattern one week ago) is shown. The same series - in a different scale - is plotted again in fig.6. Furthermore the hourly standard deviation of each of the four presented models is shown.

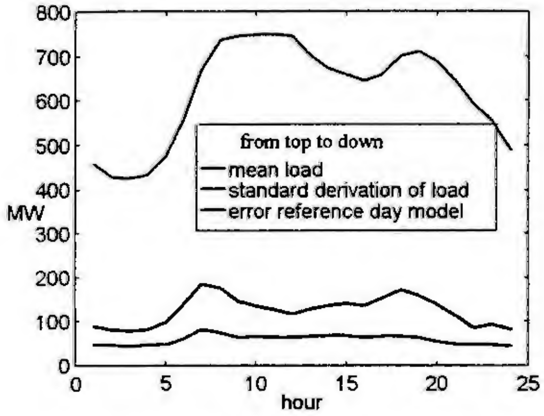


Fig. 5 Hourly year-average load values

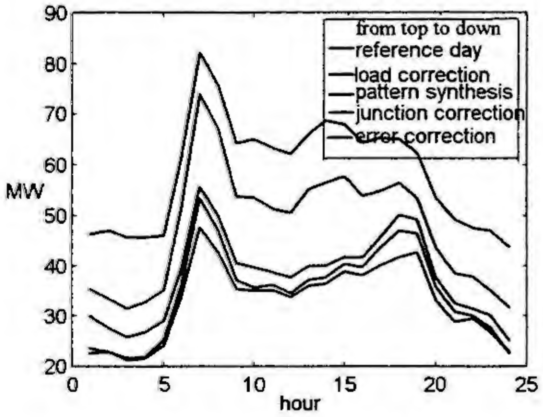


Fig. 6 Model-error standard deviation

From these pictures the contribution of every model part to the reduction of prediction error can be seen. So the intense profit resulting from the load correction model and the pattern synthesis becomes clear, while both correcting models provide only small improvement of the whole model.

Trying to compare the performance with other prediction tools one is interested in relative measures. The usual way is to relate the shown mean squared prediction errors to mean load (fig.7). Unfortunately this is not very significant. In large power utilities stochastic effects will be averaged simplifying the prediction task. Further the composition of customers has a big influence on occurring prediction problems. For these reasons a different reference basis should be used. A suited one might be the error of the simple reference model (load pattern one week ago). The prediction error of any forecasting algorithm related to theses reference model error is a very qualified and conceivable measure.

Fig.8 shows that the final algorithm will reduce this error down to 65-45% depending on the hour of the day. Thus a clear assessable proof of algorithm's achievement is given.

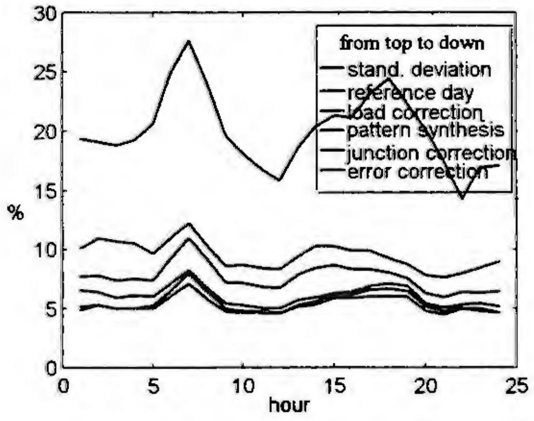


Fig. 7 Relative model error concerning mean load

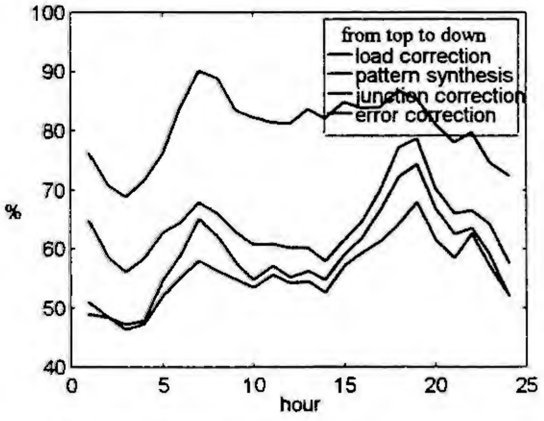


Fig. 8 Relative model error concerning simple reference model

The remaining errors in the early morning and evening hours could also be reduced by the load correction model regarding the intensity of daylight. But suitable data was not available.

Conclusion

An algorithm for short term load prediction was presented. It consists of four separately to parametrize parts. So it was possible to establish appropriate structures - two neural and two statistical - for each segment of the algorithm. To prevent problems of nonlinear optimization within the feedforward net and to obtain an optimal model size and structure selection, forward regression over a polynomial neuron pool was performed. The problem of locating neuron's centres within the normalized Gaussian network was solved by placing them directly at current reference data patterns. The statistical models, a static continuation correction and an autoregressive error model are linear in its parameters (with respect to an equation error) and easy to parametrize. Due to the small number of model parameters and required reference data, the algorithm is suited for nonstationary load pattern.

Much effort was done to illustrate the reduction of the prediction error by each part of the algorithm. In this context a performance measurement - relating the prediction error to a simple 'one week ago reference' - is proposed. The presented prediction tool proved to be very transparent and robust.

References

1. Baumann, T.; Strasser, H.; Landrichter, H.: Short-term load forecasting methods in comparison: Kohonen Learning, Backpropagation Learning, Multiple Regression Analysis and Kalman Filtes. 11th Power Systems Computation Conference (PSCC); Avignon 1993, vol. 1, pp.445-451.

2. Cichocki, A.; Unbehauen, R.: Neural Networks for Optimization and Signal Processing. J. Wiley & Sons Ltd. & B. G. Teubner, Stuttgart, 1993.

3. Draper, N.R.; Smith. H.: Applied regression analysis, John Wiley, New York, 1981.

4. Haykin, S.: Neural networks - a comprehensive foundation. Macmillan College Publishing Company, New York, 1994.

5. Mbamalu, G.A.N.; El-Hawary, M.E.: Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation. Transaction on Power Systems, vol. 8 (1993), no. 1, pp.343-348.

6. Moghram, I.; Rahman, S.: Analysis and evaluation of five short-term load forecasting techniques. Transaction on Power Systems, vol. 4 (1989), no. 4, pp.1484-1491.

7. Sjöberg, J.; McKelvey, T.; Ljung, L.: On the use of regularization in system identification. In: Preprint 12th IFAC World Congress, Sydney, 1993, vol. 7, pp.381-386.

SOME CONSIDERATIONS ABOUT POWER SYSTEMS MODELING CONCERNING TRANSIENT STABILITY ANALYSIS

Newton G. Bretas and Luís Fernando Costa Alberto

Electrical Eng. Dept., EESC - USP
13560-250, São Carlos - SP - Brasil
e-mail: ngbretas@mogno.sel.eesc.sc.usp.br

Abstract. In this work, considerations about the current power system modeling, related to transient stability analysis are presented. It is shown that the swing equations do not present equilibrium solutions if damping coefficients are neglected. In this way it is not possible to characterize the stability of the system in the Lyapunov's sense due to some structural problems related to the system modeling. Only the one-machine versus the infinite bus system has a solution with equilibrium points in the Lyapunov's sense. Those structural problems are highlighted in very simple examples. Stability studies are shown using One Machine as Reference (OMR) and using the Center of Angle as Reference (COA). In fact they are equivalent and do not eliminate the aforementioned structural problems. A possible solution for these difficulties is developed, that is, modeling the system loads as frequency dependent.

I. Preliminaries

Nowadays the increase of load demand and interconnection between systems as well as the necessity of offering better customer services have increased. As a consequence the complexity of the power system operation has become much more complex. The systems have been operating closer to the generating capacity and transmission limits and closer also to the stability limits. The protection schemes must be more efficient and the stability analysis must be faster and more accurate. As a consequence, on-line implementations of stability analysis with efficient computational methods have been one of the main concerns for research. For that purpose, however, no concern on system modeling improvement for transient stability studies has been observed.

The transient stability studies are described by an autonomous system of differential equations. Related to that model, assumptions are made in order to reduce the computational efforts in analyzing the transient stability of power systems. In that way the loads are modeled as constant impedances and the damping coefficients are neglected. It is also assumed that the mechanical input power as well as the electromotive force behind the generator transient reactances are kept constant. By admitting these assumptions, the systems do not present equilibrium solutions (except in the case of one-machine versus infinite bus) after the fault clearing time.

Although synchronism is not equivalent to stability, the power system operators are usually concerned if the power system machines are in synchronism or not after any fault. In order to study the system synchronism researchers use the OMR as well as the COA because the related equations present equilibrium points. In fact using those reference frames, the stability conditions are traduced in synchronism condition of the original system. The synchronism between machine angles does not guarantee stability because the synchronized machines keep accelerating altogether. However the structural problem of stability remains unresolved. In this work all these kinds of concerns are highlighted using easy and simple examples.

To avoid the structural problem (non-existence of equilibrium points), modeling the loads as frequency dependent is proposed. The existence of equilibrium solutions in velocities different to the synchronous velocity becomes possible. An associated disadvantage is the necessity of preserving the network structure.

The main concern of this paper is to make considerations about the current Power System modeling related to transient stability studies. First of all, in section II, the classic power system modeling for transient stability studies is summarized. In section III, the concept of stability in the Lyapunov's sense is reviewed. The difficulties in finding equilibrium points for one-machine and the multi-machine power systems, are described in section IV and section V. In section VI and VII, the stability analysis formulation using One Machine as Reference and using the Center of Angle as reference are presented as well as their equivalence. It is also shown that those representations do not eliminate the structural problems and some observations are made in section VIII. Finally a frequency dependent load modeling is developed to eliminate the structural problems in section IX. Conclusions are presented in section X.

II. Mathematical modeling

A suitable mathematical representation is necessary for the power system stability studies. The differential equations that describe the system behavior are obtained by power balance applied to each machine of the system. In other words, the difference between the mechanical input power of the machine and its output electrical power is equal to the accelerating power plus the damping power, that is:

$$M_i\ddot{\delta}_i + D_i\dot{\delta}_i = p_{mi} - p_{ei} \qquad i = 1,\ldots,n \qquad (1)$$

M_i and D_i are respectively the inertia coefficient and the damping coefficient of ith machine. P_{mi} and P_{ej} are the mechanical input power and the electrical output power machine respectively. δ_i is the power angle of generator i. The output electrical power p_{ei} is an expression based on the synchronous machine differential equations, and on the network algebraic equations. In the intended studies, a detailed machine representation is not necessary. Suitable hypotheses, which simplify the mathematical model of the system dynamic behavior, are normally used.

- It is assumed that the network is in sinusoidal steady state condition, that is, the transmission network time constants are negligible compared to the electromechanical frequency of oscillations.
- The synchronous machine is modeled as a voltage source of constant magnitude in series with a reactance that is commonly called direct axis transient reactance.
- The phase angle behind transient reactance coincides with the power angle δi.
- Loads are represented as constant impedances.
- It is assumed that the mechanical power p_{mi} keeps constant, and equal to the pre-fault value, during the whole time interval of interest. (Nowadays there are already fast governors that invalidate this assumption.)

Let an electrical power system from Figure I be constituted by n generators and by the transmission network as described by admittance matrix $\underline{Y}_{BUS}$. The n generators are connected on the network through their transient reactances in the n-first network nodes. In the next m nodes, there are only loads and the completed transmission network has, in this way, 2n+m nodes.

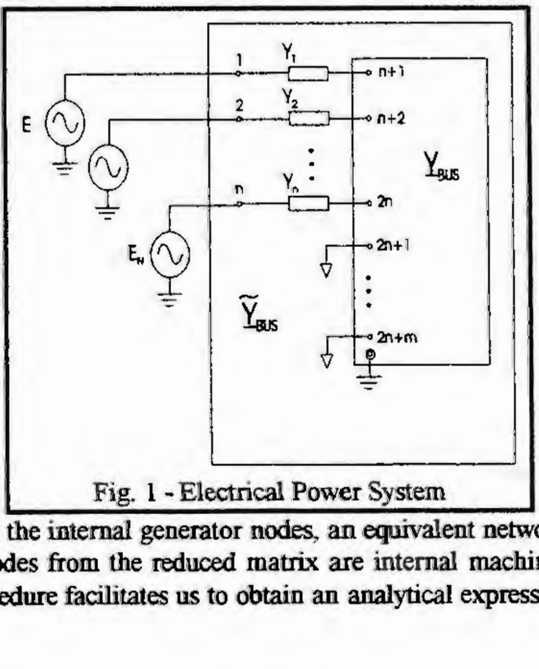


Fig. 1 - Electrical Power System

Reducing the system to the internal generator nodes, an equivalent network will be obtained. It is important to remember that the n nodes from the reduced matrix are internal machine nodes. The network topology is masked although this procedure facilitates us to obtain an analytical expression for p_{ei} as a function of δ_i's, that is:

$$p_{ei} = |E_i|^2 G_{ii} + \sum_{\substack{j=1 \\ \neq i}}^{n} |E_i||E_j||Y_{ij}|\left[\cos\left(\phi_{ij} - \left(\delta_i - \delta_j\right)\right)\right] \quad (2)$$

The swing equation of the system will be described by the following equation:

$$M_i\ddot{\delta}_i + D_i\dot{\delta}_i = p_{mi} - |E_i|^2 G_{ii} - \sum_{\substack{j=1 \\ \neq i}}^{n} \left[C_{ij}\ \mathrm{sen}(\delta_i - \delta_j) + D_{ij}\cos(\delta_i - \delta_j)\right] \quad (3)$$

where:

$$D_{ij} = |E_i||E_j||Y_{ij}|\cos\phi_{ij} = |E_i||E_j|G_{ij} \quad \text{and} \quad C_{ij} = |E_i||E_j||Y_{ij}|sen\phi_{ij} = |E_i||E_j|B_{ij}$$

III. Stability in the Lyapunov's sense

The stability concepts derive from intuitive ideas. We say that a system is stable if it keeps or returns to a normal operating state after some perturbation. Mathematically, an equilibrium solution x_o of an autonomous system described by the differential equation $\dot{x} = f(x)$ is stable in the Lyapunov's sense, or only stable, if for each real number $\varepsilon > 0$, there exists a real number $\delta > 0$ such that for every initial condition $x(t_o)$ satisfying the inequality $\|x(t_o) - x_o\| < \delta$, the system's trajectory $x(t)$ satisfies the inequality $\|x(t) - x_o\| < \varepsilon$ for every $t > t_o$ [5]. In other words, the system's trajectory, produced by the dynamic equations, does not diverge from the equilibrium for initial conditions in the vicinity of this equilibrium point.

IV. Stability studies for one-generator and one load system

The simplifying hypothesis used previously in the classical model reduce the computational efforts in solving the stability problems but they result in conceptual difficulties. To show those difficulties and their origin, let the system constituted by one machine and a load connected through a double transmission line as shown in Figure 2.a.

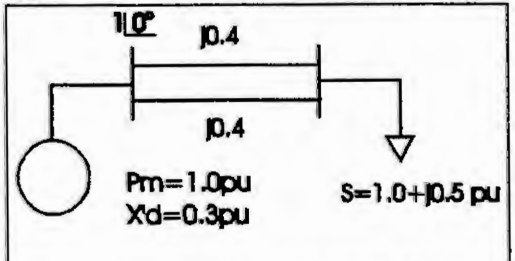


Fig. 2.a - One machine and one load system

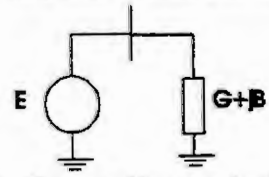


Fig. 2.b - One-machine one-load system reduced to the generator internal node

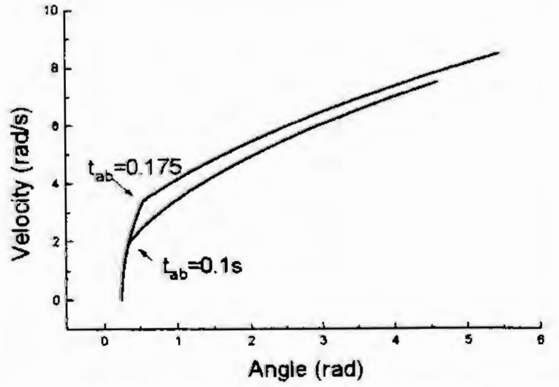


Fig. 3 - Phase Portrait of the system of Figure 2 at two distinct clearing times (0.1s and 0.175s)

Suppose a three-phase short-circuit occurs in the middle of one of the two lines. After some time the fault is eliminated by opening the faulted line. Using the hypothesis described previously, in modeling the system, and reducing the network to the generator node, the reduced equivalent circuit of Figure 2.b is obtained.

The differential equation that describes the system's behavior without damping will be:

$$M\ddot{\delta} = P_m - P_e = P_m - E^2 G \quad (4)$$

The difference between the faulted system and the post-fault system is restricted to the parameter G, which is the real part of the reduced system impedance. Both the mechanical power and the electromotive force are assumed to be constants in the classical model during the time of interest.

The pre-fault is a situation of equilibrium $\left(P_m = E^2 G\right)$. The parameters P_m and E are calculated from this situation. In the fault and post-fault situations, the conductance G will be different from the corresponding pre-fault situation, therefore there will not be any equilibrium point to the system in these situations. This set of differential equations does not have equilibrium points neither equilibrium solutions at all and it is impossible to characterize stability in the Lyapunov's sense. Figure 3 is a phase portrait that shows the results of simulating the system of Figure 2 for two different clearing times. Clearly the concept of equilibrium point in the Lyapunov's sense is not applied.

V. Stability studies for multimachine systems

The questions concerning equilibrium points for the one-machine system also happen in a multimachine system situation. Classically the stability of a system with several machines is realized using an infinite bus as reference. The infinite bus is equivalent to a machine of infinite inertia whose velocity keeps constant (and equal to synchronous velocity) independently of the power it supplies. If the system is stable, the machine velocities will return to the synchronous velocity or at least will oscillate around it. If the system is unstable otherwise, at least one of the system's machine will separate, in angle, from the infinite bus. Figure 4 shows a system of one machine versus an infinite bus. Some of the trajectories of that system corresponding to different clearing times are represented in Figure 5.

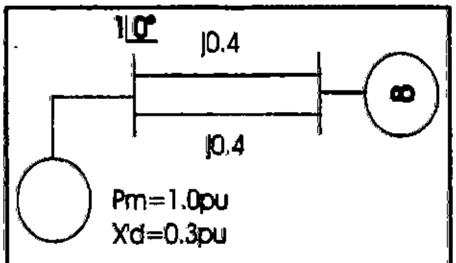


Fig. 4 - .A machine versus an infinite bus

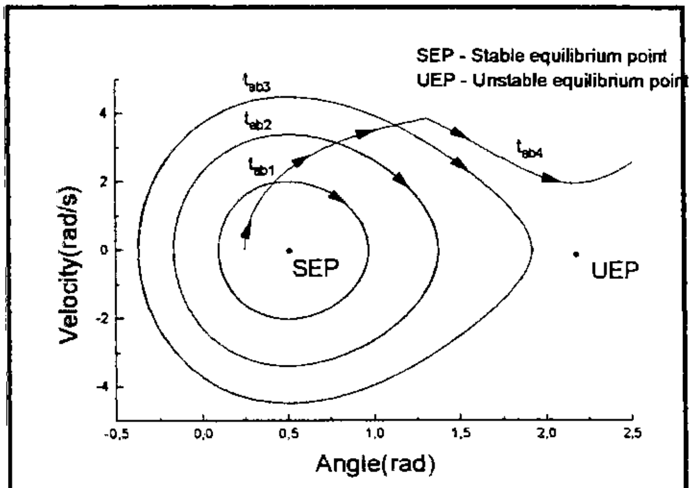


Fig. 5 - Phase Portrait of the system of Figure 4 at different clearing times

In this case the stability concepts in the Lyapunov's sense are perfectly applied because there exists a stable equilibrium point whose stable trajectories do not diverge from it.

In real systems there are no infinite buses, all machines have a dynamic described by their swing equations. In a n-machine system there are 2n first order differential equations where the reference is a synchronously rotating reference frame.:

$$\begin{cases} \dot{\omega}_i = \dfrac{P_{mi} - P_{ei}}{M_i} & i = 1, \ldots, n \\ \dot{\delta}_i = \omega_i \end{cases} \qquad (6)$$

In the p_{ei} expression (equation 2), the power can be expressed only in terms of difference between machine angles and therefore, the system will have one degree of freedom. Thus, one of them must be used as a reference angle so that the system becomes solvable.

When one machine angle is used as a reference, it is known from the power flow studies that the power generated in this machine must be free so that it compensates for the energy unbalance. In other words, the power generated by this machine will be a function of losses in the lines corresponding to the reduced network (in the reduced network the transference conductances are not negligible). As the mechanical input power is assumed to be constant and since after the elimination of the faulted line, the capacity of network transmission is reduced, in general it is not possible to obtain the power balance in the system.

VI. One machine equation as reference

We have seen, in the previous section, that it is necessary to take a **machine angle** as a reference (infinite bus) to define the "equilibrium" of the other machines. This procedure has the inconvenience of resulting in a large power mismatch in the reference machine, so the equilibrium of n-1 machines does not guarantee the system stability neither the synchronism between machines. To overcome this problem, one **machine equation** is taken as a reference. Let us subtract from each equation the equation of the n^{th} machine(reference). In this case a system of 2n-2 first order differential equations is obtained as a function of the new state variables $[\delta_1-\delta_n,......\delta_{n-1}-\delta n|\omega_1-\omega_n,......,\omega_{n-1}-\omega_n]$:

$$\begin{cases} \dot{\omega}_i-\dot{\omega}_n=\dfrac{P_{mi}-P_{ei}}{M_i}-\dfrac{P_{mn}-P_{en}}{M_n} \\ \dot{\delta}_i-\dot{\delta}_n=\omega_i-\omega_n \end{cases} \quad i=1,\dots,n-1 \quad (10)$$

The solution of equation (10) in (δ_{in} ω_{in}) is decoupled from the reference equations, and now that system presents equilibrium points in the Lyapunov's sense. In the equilibrium, the following is obtained:

$$\begin{cases} \dot{\omega}_i-\dot{\omega}_n=0 \\ \dot{\delta}_i-\dot{\delta}_n=0 \end{cases} \quad i=1,\dots,n-1 \qquad (11) \qquad \text{thus,}$$

$$\begin{cases} \dfrac{P_{mi}-P_{ei}}{M_i}=\dfrac{P_{mn}-P_{en}}{M_n} \\ \omega_i=\omega_n \end{cases} \quad i=1,\dots,n-1 \qquad \text{or,} \qquad \begin{cases} \dfrac{P_{m1}-P_{e1}}{M_1}=\cdots=\dfrac{P_{mi}-P_{ei}}{M_i}=\cdots=\dfrac{P_{mn}-P_{en}}{M_n} \\ \omega_1=\cdots=\omega_i=\cdots=\omega_n \end{cases} \qquad (12)$$

In the equilibrium, the n-machine velocities and accelerations must be the same. If the system is stable, the difference between the machine angles will tend to a finite value; that will be the equilibrium point of the system. Therefore, the stability studies, in the Lyapunov's sense, of an equilibrium point of these 2n-2 differential equations are equivalent to study the synchronism between all machines in the system. The advantage of this representation is that the stability in the Lyapunov's sense is applied to study the synchronism between machines although the mismatch in the machines might remain large.

VII. The center of angle as a reference [2]

Using a machine as a reference for angle, the equilibrium points of the system of 2n-2 equations might be far away from the true system equilibrium point. The Center of Angle as Reference has been accepted as a suitable solution for this problem since it attributes part of the power mismatch to each of the system machines. That affirmative however is false as shown below.

The Center of Angle as a reference uses the same principle of center of mass from the mechanics. A new variable "δo" called Center of Angle is defined as:

$$\delta_o=\frac{1}{M_T}\sum_{i=1}^{n}M_i\delta_i \ \text{onde}\, M_T=\sum M_i \qquad (13)$$

The equation that describes the dynamics of the center of angle is obtained derivating the equation (13) two times, i.e.,

$$M_T\ddot{\delta}_o=P_{COA} \qquad (14) \qquad \text{where:} \qquad P_{COA}=\sum_{i=1}^{n}P_i-2\sum_{i=1}^{n}\sum_{j=i+1}^{n}D_{ij}\cos\delta_{ij} \qquad (15)$$

Now the machine angles are measured with respect to the COA. The equations in these new state variables are:

$$\theta_i = \delta_i - \delta_o$$
$$M_i\ddot{\theta}_i = P_{mi} - P_{ei} - \frac{M_i}{M_T}P_{COA} \qquad (16)$$

Writing the system in the new state variables, results in:

$$\begin{cases} \dot{\tilde{\omega}}_i = \dfrac{P_{mi}-P_{ei}}{M_i} - \dfrac{1}{M_T}P_{COA} & i = 1,\ldots,n \qquad (17) \\ \dot{\theta}_i = \tilde{\omega}_i \end{cases}$$

This is a system of 2n differential equations, which has only 2n-2 variables. The equations are linearly dependents and the solution of 2n-2 equations is enough to completely know the system's behavior. The equilibrium will result in:

$$\begin{cases} \dot{\tilde{\omega}}_i = 0 \\ \dot{\theta}_i = 0 \end{cases} \quad i = 1,\ldots,n \qquad (18) \qquad \text{thus,}$$

$$\begin{cases} \dfrac{P_{mi}-P_{ei}}{M_i} = \dfrac{1}{M_T}P_{COA} \\ \tilde{\omega}_i = 0 \end{cases} \quad i = 1,\ldots,n \quad \text{or,} \quad \begin{cases} \dfrac{P_{m1}-P_{e1}}{M_1} = \cdots = \dfrac{P_{mi}-P_{ei}}{M_i} = \cdots = \dfrac{P_{mn}-P_{en}}{M_n} = \dfrac{P_{COA}}{M_T} \\ \tilde{\omega}_1 = \cdots = \tilde{\omega}_i = \cdots = \tilde{\omega}_n = 0 \end{cases} \qquad (19)$$

The equality of power acceleration in (19) can be obtained quite directly:

$$\frac{P_{m1}-P_{e1}}{M_1} = \cdots = \frac{P_{mi}-P_{ei}}{M_i} = \cdots = \frac{P_{mn}-P_{en}}{M_n} = \frac{\sum_{i=1}^{n} P_{mi} - P_{ei}}{\sum_{i=1}^{n} M_i} = \frac{P_{COA}}{M_T}$$

Therefore, the two system representations (OMR and COA) are completely equivalent.

VIII. Synchronism and stability

A dynamic reference (COA or OMR) transforms the problem of synchronism between machine angles in a problem of stability study of a suitable set of differential equations. Those references at the beginning seem to solve the stability analysis difficulties. However the problem is a structural problem and not a reference related problem. In the situation of Figure 2 the system will be in synchronism forever although it is not a stable situation in the engineering sense. With the usual system modeling, only the governor action may force the system going to a stable situation. Another alternative is to introduce the damping and/or frequency dependent loads. This will be the subject of the next section.

IX. Frequency dependent loads and modeling problems

In the classical model, the loads are assumed to be represented by constant impedances. The real systems have loads that are frequency dependent. Suppose that, in a two-generator system studied previously, some of the loads are frequency dependents such that we can represent them in the reduced system by the following equations:

$$M_1\dot{\omega}_1 = P_{m1} - (1 + D_1\omega_1)P_{e1} \tag{20}$$
$$M_2\dot{\omega}_2 = P_{m2} - (1 + D_2\omega_2)P_{e2} \tag{21}$$
$$\dot{\delta}_{21} = \omega_2 - \omega_1 \tag{22}$$

This is a simplified representation where a static reduction of the network was carried out and it was assumed that the power supplied by the generator is frequency dependent as is showed in the equations (20),(21) and (22). Although this representation is simple, it allows us to verify the effect of frequency dependent loads in the stability studies.

In the equilibrium $\dot{\delta}_{21} = 0$, thus, $\omega_1 = \omega_2 = \omega_e$ is the equilibrium velocity of the system. From equations (20) and (21) doing $\dot{\omega}_1 = \dot{\omega}_2 = 0$ the following is obtained:

$$0 = P_{m1} - (1 + D_1\omega_e)\left[E_1{}^2G_{11} + E_1E_2Y_{12}\cos(\phi_{12} - \delta_{12})\right] \tag{23}$$
$$0 = P_{m2} - (1 + D_2\omega_e)\left[E_2{}^2G_{22} + E_2E_1Y_{21}\cos(\phi_{21} - \delta_{21})\right] \tag{24}$$

From equations (23) e (24) the value of δ_{21} and ω are obtained in the equilibrium. In this case the equations (23) and (24) are even coupled. This system has a stable equilibrium point in the Lyapunov's sense with state variables $[\delta_{21},\omega_1,\omega_2]$, and a velocity different from the synchronous velocity.

This kind of load modeling has the advantages of being more realistic than the classical model and the equilibrium points have meaning in the Lyapunov's sense. Consequently the power mismatch is eliminated when the system having an equilibrium point in velocities different to the synchronous velocity is allowed. The disadvantage of this kind of model is that it is necessary to preserve the system structure.

X. Conclusions

In this work, difficulties in the modeling of the transient stability analysis are presented. As a consequence of the simplifier hypotheses used in the classical model, it is shown that only the one-machine versus the infinite bus system has a solution with equilibrium points in the Lyapunov's sense. All other cases have intrinsic difficulties related to modeling problems. In the case of stability studies, the "lines" have large resistive components and consequently large "losses", therefore the power mismatch problem is even worse. A frequency dependent load modeling is shown to be a possible solution for those problems, however in this case the structure preserving approach is necessary which constitutes in more computer work for stability analysis.

Acknowledgments. This work was in part supported by FAPESP (research foundation of São Paulo - Brazil). The authors are also indebted to Prof. Hildebrando Munhoz Rodrigues from Mathematical Department of São Paulo University (São Carlos).

References

1. Anderson, P.M. and Fouad, A.A., Power System Control and Stability, The Iowa State University press, Ames, Iowa, USA, 1977
2. Athay, T. and Podmore, R. and Virmani, S., A Practical Method for the Direct Analysis of Transient Stability, IEEE Trans. on PAS, Vol. PAS-98, n.2, March/April 1979.
3. Bergen, A.R. and Hill, D.J., A Structure Preserving Model for Power System Stability Analysis, IEEE Trans. on PAS, Vol. PAS-100, n.1, January 1981
4. Brauer, F. and Nohel, J.A., The Qualitative Theory of Ordinary Differential Equations, W.A.BENJAMIN INC., New York, Amsterdam, 1969.
5. Pai, M.A., Power System Stability: Analysis by the Direct Method of Lyapunov, North-Holland Publishing Company, Amsterdam, New York, Oxford, 1981.
6. Varaya, P. and Wu, F.F. and Chen, R.L., Direct Methods for Transient Analysis of Power Systems: Recent Results, Electronics Research Laboratory, College of Engineering University of California, Berkeley, September 1984.

Modelling of Unit–Transformers for Nonlinear Analysis

Michael Fette, Ralf Sisterhenn, Jürgen Voss

University of Paderborn
Department of Electrical Engineering
Pohlweg 47 – 49
33098 Paderborn, Germany

Abstract: In this paper we give a nonlinear model of large unit–transformers, with which it is possible to study the effects of the nonlinearities in a very structured manner, where usually a theory which allowed directly for nonlinear effects introduces much complication [1]. Also, it is possible to take different primary and secondary winding connections into account. Similar to studies we have done for synchronous machines [2] it is not the goal to fit for example all measured points of a saturation curve in detail, but we are interested in the typical behavior and the influence of important nonlinear effects on the dynamics of the system. The proposed model has a special orthogonal structure with a very compact representation which allows mathematical analysis in an efficient way. From this a family of reduced and/or simplified models, e.g. linear MIMO systems, can be calculated.

Hysteresis effects of the core are introduced in such a way to meet the objectives of nonlinear analysis methods. The hysteresis curve is represented by a trajectory of a nonlinear oscillator, a Duffing oscillator.

1. Introduction

Most of the major electric power system breakdowns in recent years have been caused by the dynamic response of the system to disturbances. Economical and environmental pressures are causing power systems to be operated ever closer to their limits of stability. Additional the dynamical behavior of power systems changes to be extremely nonlinear. It is necessary to use even improved nonlinear models for the analysis of the dynamic behavior of these systems.

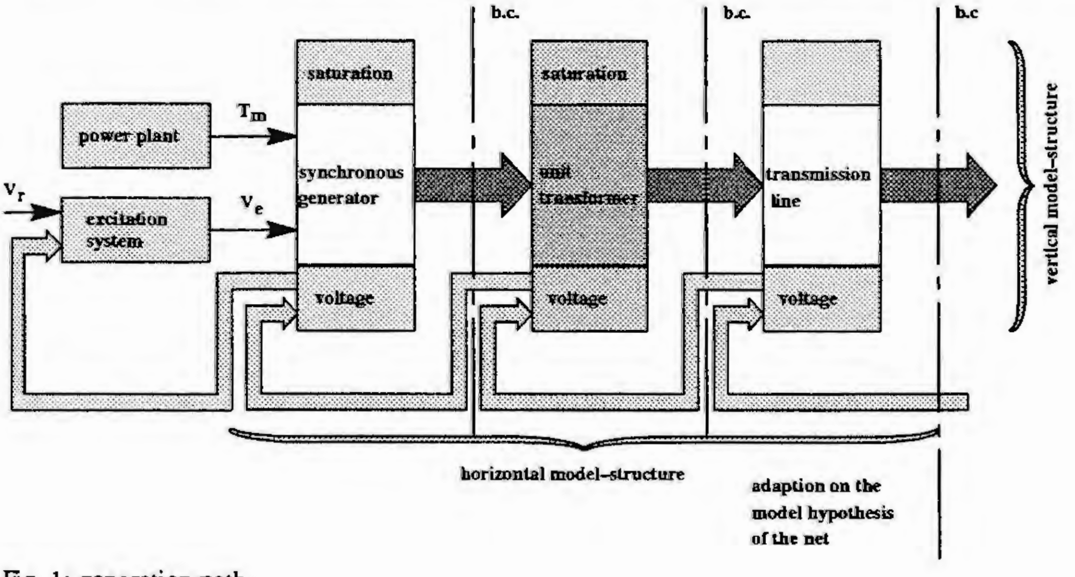


Fig. 1: generation path

An important model is that of a large three–phase unit–transformer, which steps the voltage of the generator up to the desired transmission voltage. Connecting such a step–up transformer between the generator and a transmission line permits a practical design voltage for the generator and at the same time an efficient transmission line voltage. For real–world unit–transformers under normal operating conditions the effect of magnetization of the core is an important factor, and much genuine has been used in devising methods of taking it into account. Most of these methods do not, however, introduce the nonlinear property into the basic theory. They are directed mainly to the determination of appropriate values of constants to suit the particular problem, the constants being defined in relation to a linear theory.

For analysis, power systems can be split by topological aspects generally into two basic structures, into a part with a chain structure, shown in the figure below, called "generation path" and a part with a net structure, called "distribution path" [3].

The generation path has a horizontal model structure for the connection of different models with respect to boundary conditions and a vertical structure to include additional specifying physical effects (e.g. magnetization of the core [4]) into the different basis models, if it's necessary. A model of a unit–transformer has to match the topological and mathematical assumptions made for the generation path, where the proposed mathematical methods are based on differential geometry [5]. With this, the analysis of nonlinear power systems can be done in an efficient way analytically. The proposed structure provides a transformation of known transformer models into it.

The accurate prediction of transformer steady–state and transient performances requires the accurate determination of the transformer parameters. Many of the proposed techniques do not give a clear physical picture of the saturation phenomenon and for example they didn't take into account the different constructions of the core.

2. Basic Transformer Model

For a basic transformer model some assumptions are made for simplification:

- continuously distributed magnetic fields
- neglected are: losses caused by eddy currents
- temperature dependencies of the model's parameters
- the transformers' capacitances

With these assumptions the following figure of the unit–transformer can be given in accordance to the well known Steinmetz model:

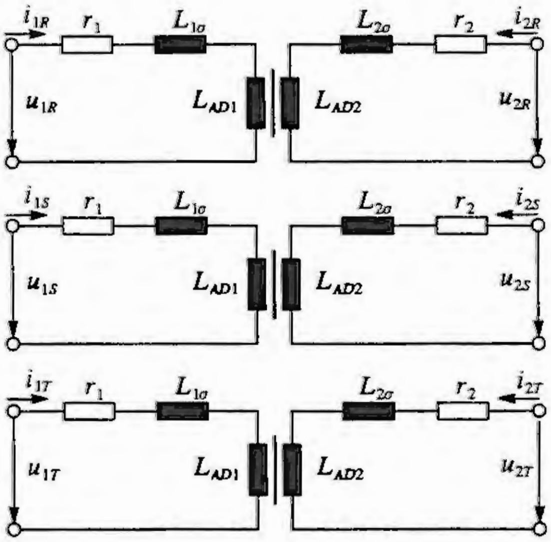


Fig. 2: Schematic representation of a transformer

Voltage equations of the transformer model for the primary (index 1) and secondary (index 2) windings can be derived immediately:

$$u_{1i} = r_1 i_{1i} + L_{1\sigma} \frac{di_{1i}}{dt} + L_{AD1} \left(\frac{di_{1i}}{dt} + \frac{di_{2i}}{dt} \frac{1}{n} \right) \qquad i = R, S, T \qquad (2.1)$$

$$u_{2i} = r_2 i_{2i} + L_{2\sigma}\frac{di_{2i}}{dt} + L_{AD2}\left(\frac{di_{2i}}{dt} + \frac{di_{1i}}{dt}\,n\right) \qquad i = R, S, T \tag{2.2}$$

The primary and secondary voltage equations can be written in a matrix representation

$$u = \left(R_T + \omega N_T\right) i + L_T \frac{di}{dt} \tag{2.3}$$

and explicitly in Park's coordinates:

$$\begin{pmatrix} u_{1d} \\ u_{2d} \\ u_{1q} \\ u_{2q} \\ u_{1n} \\ u_{2n} \end{pmatrix} = \begin{bmatrix} r_1 & 0 & \omega L_1 & n^*\omega L_{AD1} & 0 & 0 \\ 0 & r_2 & n\omega L_{AD2} & \omega L_2 & 0 & 0 \\ -\omega L_1 & -n^*\omega L_{AD2} & r_1 & 0 & 0 & 0 \\ -n\omega L_{AD1} & -\omega L_2 & 0 & r_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2 \end{bmatrix} \begin{pmatrix} i_{1d} \\ i_{2d} \\ i_{1q} \\ i_{2q} \\ i_{1n} \\ i_{2n} \end{pmatrix} + \begin{bmatrix} L_1 & n^*L_{AD1} & 0 & 0 & 0 & 0 \\ nL_{AD2} & L_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & L_1 & n^*L_{AD1} & 0 & 0 \\ 0 & 0 & nL_{AD2} & L_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & L_1 & n^*L_{AD1} \\ 0 & 0 & 0 & 0 & nL_{AD2} & L_2 \end{bmatrix} \frac{d}{dt} \begin{pmatrix} i_{1d} \\ i_{2d} \\ i_{1q} \\ i_{2q} \\ i_{1n} \\ i_{2n} \end{pmatrix} \tag{2.4}$$

with $n^* = \frac{1}{n}$.

Please notice, that the structure of these matrix representation (2.3) is identical to those of a synchronous generator [5]. For a unified analysis of the interconnection of the transformer and the machine in the generation path it is useful to represent both models in the same frame of coordinates without loss of generality. The explicit mathematical representation of the model without an incorporation of magnetic saturation effects is therefore:

$$\frac{di}{dt} = -L^{-1}\left(R + \omega N\right) i + L^{-1} u \tag{2.5}$$

The resulting matrices can be sorted into a primary and a secondary current model:

$$\frac{di_1}{dt} = A_{11} i_1 + A_{12} i_2 + B_{11} u_1 + B_{12}\, u_2 \tag{2.6}$$

$$\frac{di_2}{dt} = A_{21} i_1 + A_{22} i_2 + B_{21} u_1 + B_{22}\, u_2 \tag{2.7}$$

To incorporate the model into the structure of the generation path, it is assumed that the speed variables of the machine (index M) are the same as of the transformer:

$$\omega_T = \omega_M$$

In a linear case it is possible to define transfer functions and to calculate a distortion angle of the transformer, which can be added to the power angle of the machine, which is very useful in stability analysis when the models of the generation path are connected to a net. It is assumed, that the speed terms of the machine are constant.

$$\delta_T = \delta_M + \arctan\left\{\frac{g_2(\omega_0)}{f_2(\omega_0)}\right\} - \arctan\left\{\frac{g_1(\omega_0)}{f_1(\omega_0)}\right\}$$

In a more general multivariable case the difference angle can be calculated by

$$\Delta\Theta(i_1,\ i_2) = \arg\{U_0^H\ G_2(j\omega_0)\ U_0\} - \arg\{U_0^H\ G_1(j\omega_0)\ U_0\} + \lambda \cdot 30^0$$

where the additional term $\lambda \cdot 30^0$ take the displacement angles of the different winding interconnections into account.

3. Modelling of Saturation Effects

In the linear magnetization case the flux linkages can be calculated (3.1) from the transformers currents only by multiplication with the inductance matrix, see (2.4).

$$\begin{pmatrix} \Psi_{1d} \\ \Psi_{2d} \\ \Psi_{1q} \\ \Psi_{2q} \\ \Psi_{1n} \\ \Psi_{2n} \end{pmatrix} = \begin{pmatrix} \Lambda \cdot i_d \\ \Lambda \cdot i_q \\ \Lambda \cdot i_n \end{pmatrix} := \begin{pmatrix} \Psi_{1d0} \\ \Psi_{2d0} \\ \Psi_{1q0} \\ \Psi_{2q0} \\ \Psi_{1n0} \\ \Psi_{2n0} \end{pmatrix} \qquad with\ \ \Lambda := \begin{pmatrix} L_1 & n^*L_{AD1} \\ nL_{AD2} & L_2 \end{pmatrix} \tag{3.1}$$

In general, the time derivatives of the flux linkages must be calculated when saturation of the core is taken into account.

$$\frac{d\Psi_d}{dt} = -R\ i_d - \omega\Psi_q + u_d \tag{3.2}$$

$$\frac{d\Psi_q}{dt} = -R\ i_q + \omega\Psi_d + u_q \tag{3.3}$$

$$\frac{d\Psi_n}{dt} = -R\ i_n + u_n \tag{3.4}$$

The modelling of the magnetic saturation is done in such a way, that in the case of small currents the derived model tends continuously to the model without saturation. In contrast to common saturation models in this section an analytical model for unit–transformers is developed without any constraints on the range of the currents.

The magnetic fluxes in (3.2) to (3.4) can be obtained from the magnetization curve [4]. In view of the fact that the magnetic fluxes of one direction cause a cross–magnetization of the iron core in the other direction, the fluxes of one axis must be weighted additionally by a function of the other axis' currents. Therefore, the magnetic flux of one axis depends on the corresponding current vector and on the norm of the remaining current vector, only

It has also been found experimentally that the effect of cross–magnetizing will reduce the magnetic flux linkages in the d–, q– and 0–axis. Therefore, other authors define the per unit d–, q– and 0–axis mutual reactances as the per unit d–, q– and 0–axis mutual flux linkages divided by the corresponding per unit d– q– and 0–axis ampere–turns respectively, the cross–magnetizing effect could be included [1] in these reactances. From this, the "concept of the saturated reactances" can be derived [1], saturation factors S_d , S_q and S_0 can be determined experimentally, from which the mutual reactances can be calculated.

A good candidate for representing the magnetic flux linkage is given in equation (3.5), where the mathematical expression for the d–axis flux linkage is shown as an example. Note, that the arctan–function does not fit a measured magnetization curve very well, but in a suitable operating region, the approximation of the curve is well. On the other hand these mathematical representation matched the mathematical requirements to make analytical investigations of the dynamic behavior, even in the nonlinear case. The mathematical structure of the curve is represented correctly.

$$\Psi_d = \Psi_d\left(i_d,\ \|i_q\|,\ \|i_n\|\right) = \exp\left(-k_q\|i_q\|^2 - k_n\|i_n\|^2\right)\frac{2\Psi_{sd}}{\pi}\begin{bmatrix}\arctan\left(\dfrac{\pi\Psi_{1d0}}{2\Psi_{sd}}\right)\\ \arctan\left(\dfrac{\pi\Psi_{2d0}}{2\Psi_{sd}}\right)\end{bmatrix} = \exp\left(-k_q\|i_q\|^2 - k_n\|i_n\|^2\right)\Phi_d(i_d) \tag{3.5}$$

Similar expressions can be given for Ψ_q, Ψ_n. The chosen mathematical representation has the properties, that in a limit case, when Ψ_{sd}, Ψ_{sq}, $\Psi_{sn} \rightarrow \infty$ and k_d, k_q, $k_n \rightarrow 0$ the nonlinear model tends to the basic transformer model without incorporation of saturation effects. The validation process of such kind of "limit model" can be made very effectively.

Similar to the linear case (2.5) an explicit model for the currents can be calculated. The resulting model has the following structure with corresponding parameter matrices not stated out here explicitly:

$$\frac{di}{dt} = Ai + L\frac{di}{dt} + C\Psi + Bu \tag{3.6}$$

$$\frac{di}{dt} = (I - L)^{-1}(Ai + C\Psi + Bu) \tag{3.7}$$

Remarks

- Saturation effects of a transformer can be incorporated in the same way as in the model of a synchronous machine. With this, a unified analysis method can be used especially for the generation path where the machine and the transformer are coupled in a chain structure.
- Normally, a more simplified model of a transformer is used for a lot of analysis methods where the transformer is represented with its short–circuit impedance, only. This is the simplest way to incorporate the transformer model into the synchronous machine model in which the short–circuit impedance of the transformer is added to the synchronous impedance of the machine.
- The automatic change of the ratio of the transformer can be modelled via an exosystem.
- This modelling concept can be used for a general transformer. In the case of a unit–transformer the number of state variables can be reduced, since a unit–transformer operates normally under symmetric conditions. For the synchronous machine directly interconnected to the unit–transformer it is necessary to operate under symmetric conditions. The 0–component of the state vector is therefore neglected. On the other hand, from the modelling concept there is no restriction to these special case.

4. Hysteresis modelling

Hysteresis effects of the magnetic core are normally modelled via a chart or a look–up table for numerical simulations. These kind of modelling prevent any kind of analytic analysis of the dynamic behavior. In this paper the hysteresis effects are modelled with the help of a nonlinear oscillator, a Duffing–oscillator [6]. The general structure of this oscillator is

$$\frac{d^2x}{dt^2} + d\frac{dx}{dt} - x + ax^3 = f\cos(\omega t) = i(t) \tag{4.1}$$

Variable x represents the magnetic state Ψ_H of the core. The magnetic flux linkage is calculated by $\Psi = \arctan(x)$. In Fig.4 the obtained result with suitable parameters are shown. The oscillator is controlled by the frequency and the amplitude of the input current. The shape of the hysteresis can be matched with parameter d of equation (4.1), mainly.

To incorporate modelling of the hysteresis effects into the basic transformer model, it is assumed that the magnetic fluxes are depended on the electrical currents and the remanent magnetic field of the core. The describing equations are given by (4.2). $\phi_d(i_d, z_d)$ and $\phi_q(i_q, z_q)$ substitute the former only dependency on the currents. Cross–magnetizing effects are modelled by exp–functions and the saturation by the arctan–function similar to the non–hysteresis case (3.1).

$$\begin{bmatrix}\Psi_{1d}\\ \Psi_{2d}\\ \Psi_{1q}\\ \Psi_{2q}\end{bmatrix} = \begin{pmatrix}\Lambda \cdot \phi_d(i_d, z_d)\\ \Lambda \cdot \phi_q(i_q, z_q)\end{pmatrix} := \begin{bmatrix}\Psi_{1dH}\\ \Psi_{2dH}\\ \Psi_{1qH}\\ \Psi_{2qH}\end{bmatrix} \tag{4.2}$$

To incorporate the Duffing equation into the model, the d–, q– and 0–axis must be assigned to a set of Duffing equations in state space representation. As an example, one yields for the d–axis primary and secondary winding system:

$$\dot{z}_{1d} = \begin{pmatrix}\dot{z}_{1d1}\\ \dot{z}_{1d2}\end{pmatrix} = \begin{pmatrix}\alpha z_{1d2}\\ \beta z_{1d1} + \gamma z_{1d1}^3 + \delta z_{1d2} + \varepsilon i_{1d}\end{pmatrix} \tag{4.3}$$

$$\phi_{1d}(1,\ 0)z_{1d} = c^T z_{1d} = z_{1d1}$$

and

$$\dot{z}_{2d} = \begin{pmatrix}\dot{z}_{2d1}\\ \dot{z}_{2d2}\end{pmatrix} = \begin{pmatrix}\alpha z_{2d2}\\ \beta z_{2d1} + \gamma z_{2d1}^3 + \delta z_{2d2} + \varepsilon i_{2d}\end{pmatrix} \tag{4.4}$$

$$\phi_{2d}(1,\ 0)z_{2d} = c^T z_{2d} = z_{2d1}$$

The magnetic output variables are collected in the vector ϕ_d.

Analogously, to calculate the voltage equations of the basic model, time derivations of the magnetic flux linkages must be computed in the hysteresis–case, too:

$$\frac{d\Psi_d}{dt} = \left(\frac{d}{dt}\exp\left(-k_q\|\phi_q\|^2\right)\right)\Phi_d(\phi_d) + \exp\left(-k_q\|\phi_q\|^2\right)\frac{d}{dt}\Phi_d(\phi_d) \tag{4.5}$$

With equation (3.2) one yields:

$$\begin{aligned}\frac{d\Psi_d}{dt} &= \exp\left(-k_q\|\phi_q\|^2\right)\left[S_d(\phi_d)\,\Lambda\,\frac{d\phi_d}{dt} - 2k_q\left(\phi_q^T\frac{d\phi_q}{dt}\right)\Phi_d(\phi_d)\right]\\ &= -Ri_d - \omega\Psi_q + u_d\end{aligned} \tag{4.6}$$

The saturation matrix $S_d(\phi_d)$ has a diagonal structure and can be inverted since its entities are the derivations of the magnetic curve which is represented by arctan–functions.

$$S_d^{-1}(\phi_d) = \begin{bmatrix}1 + \left(\frac{\pi\Psi_{1dH}}{2\Psi_{sd}}\right)^2 & 0\\ 0 & 1 + \left(\frac{\pi\Psi_{1dH}}{2\Psi_{sd}}\right)^2\end{bmatrix} \tag{4.7}$$

Equation (4.6) can be solved and the resulting mathematical model of the d–axis is represented now by

$$\frac{dz_{1d1}}{dt} = A_{11d} i_{1d} + A_{12d} i_{2d} + L_{11d}\frac{dz_{1q1}}{dt} + L_{12d}\frac{dz_{2q1}}{dt} + C_{11d}\Psi_{1q} + C_{12d}\Psi_{2q} + B_{11d}u_{1d} + B_{12d}u_{2d} \tag{4.8}$$

and

$$\frac{dz_{2d1}}{dt} = A_{21d} i_{1d} + A_{22d} i_{2d} + L_{21d}\frac{dz_{1q1}}{dt} + L_{22d}\frac{dz_{2q1}}{dt} + C_{21d}\Psi_{1q} + C_{22d}\Psi_{2q} + B_{21d}u_{1d} + B_{22d}u_{2d} \tag{4.9}$$

The time derivative terms of equation (4.8) and (4.9) are substituted with the corresponding state variables:

$$\frac{dz_{1d1}}{dt} = az_{1d2};\ \frac{dz_{2d1}}{dt} = az_{2d2};\ \frac{dz_{1q1}}{dt} = az_{1q2};\ \frac{dz_{2q1}}{dt} = az_{2q2};$$

The achieved equation is:

$$\begin{pmatrix} az_{1d2} \\ az_{2d2} \end{pmatrix} = \begin{pmatrix} A_{11d} i_{1d} + A_{12d} i_{2d} + L_{11d} az_{1q2} + L_{12d} az_{2q2} + C_{11d}\Psi_{1q} + C_{12d}\Psi_{2q} + B_{11d}u_{1d} + B_{12d}u_{2d} \\ A_{21d} i_{1d} + A_{22d} i_{2d} + L_{21d} az_{1q2} + L_{22d} az_{2q2} + C_{21d}\Psi_{1q} + C_{22d}\Psi_{2q} + B_{21d}u_{1d} + B_{22d}u_{2d} \end{pmatrix} \tag{4.10}$$

and in a more compact representation:

$$az = A\ i + L\ a\ z + C\ \Psi + B\ u \tag{4.11}$$

From this representation the currents of the transformer can be calculated in dq0–coordinates by

$$i = A^{-1}[(I - L)\ az - C\ \Psi - B\ u] \tag{4.12}$$

The complete mathematical system of a transformer with hysteresis effects is therefore:

$$\dot{z}_C = f(z,\ i) \tag{4.13}$$

with

$$z_C := \left(z_{1d1},\ z_{1d2},\ z_{1q1},\ z_{1q2},\ z_{2d1},\ z_{2d2},\ z_{2q1},\ z_{2q2}\right)^T$$

and the output equation (4.12)

$$i = g(z,\ \Psi,\ u) \tag{4.14}$$

As an example a numerical simulation of primary and secondary currents of one phase of these transformer model with hysteresis effects are shown in Fig. 3. Fig.4 shows a plot of the hysteresis, a trajectory of a Duffing oscillator.

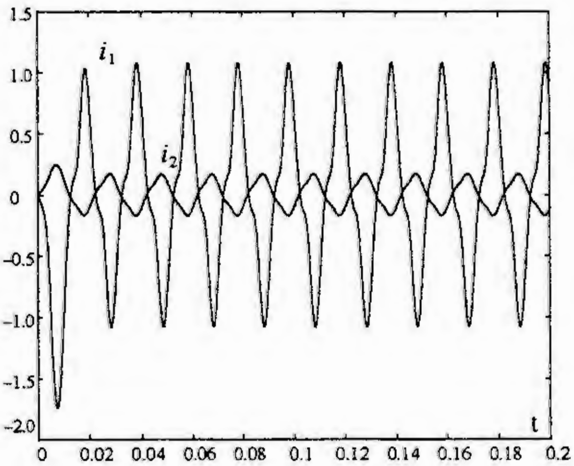


Fig. 3: Primary and secondary currents of one phase of the transformer

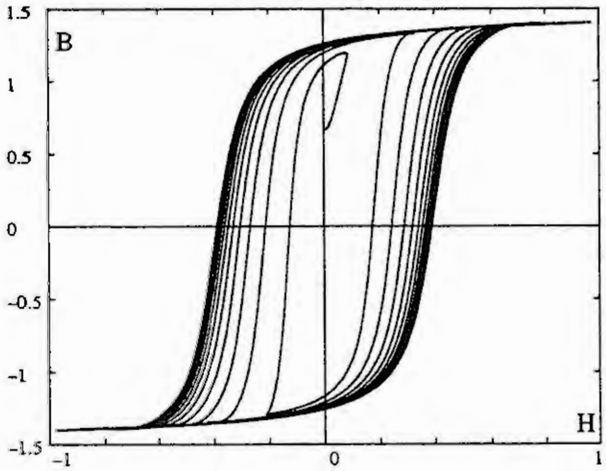


Fig. 4: Hysteresis with time varying input current

5. Conclusion

In this paper a nonlinear model of large unit–transformers is presented, with which it is possible to study the effects of the nonlinearities in a very structured manner. At first a basic transformer model is derived, which is very similar to the well known Steinmetz model. Saturation effects of the core are incorporated into the basic model in such a way that in the case when the transformer currents are small the saturation effects vanishes the extended model tends as a limit process to the basic model. Similar to studies we have done for synchronous machines [5] it is not the goal to fit for example all measured points of a saturation curve in detail, but we are interested in the typical behavior and the influence of important nonlinear effects on the dynamics of the system. In the same sense, to incorporate hysteresis effects into the transformer model, the hysteresis is modelled by a trajectory of a nonlinear oscillator, a Duffing oscillator. The analytical framework for nonlinear analysis is preserved with this representation. It is also possible to take different primary and secondary winding interconnections into account.

References

[1] Adkins, B.; Harley, R.G.: "The General Theory of Alternating Current Machines: Application to Practical Problems", Chapman and Hall, London 1975

[2] Dourdoumas, N.; Fette, M.; Kröger, C.; Voss, J.: "Structured Modelling and Analysis of Saturated Synchronous Machines", Proceedings of the IMACS 1.MATHMOD Vienna, Part 4, February 1994, pp. 703 – 710

[3] Dourdoumas, N.; Fette, M.; Voss, J.: "Modelling and Simulation of Nonlinear Power Systems on Manifolds", Proceedings of the IMACS–Congress, Dublin, July 1991, pp. 1133 – 1134

[4] Ostovic, V.: "Dynamics of Saturated Electric Machines", Springer–Verlag, New York, Berlin, Heidelberg, 1989

[5] Fette, M.: "Strukturelle Analyse elektrischer Energieversorgungssysteme", Fortschrittberichte VDI, Reihe 21, VDI–Verlag, 1993

[6] Duffing, G.: "Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung", Vieweg Verlag, Braunschweig, 1918

On the Solvability of Nonlinear Differential–Algebraic Models of Electrical Power Systems

Michael Fette, Dadi Hisseine, Jürgen Voss

University of Paderborn
Department of Electrical Engineering
Pohlweg 47 – 49
33098 Paderborn, Germany

Abstract. For analysis of the dynamical behavior of electrical power systems more and more nonlinear descriptions of the different systems elements, e.g. the synchronous machine, will be used. The interconnection of such nonlinear differential systems to complex transfer systems will lead to additional algebraic conditions due to Kirchhoff's law and Tellegen's theorem, compulsorily. A general nonlinear description of power systems is therefore given by differential–algebraic equations.

Since for such complex systems an analytical solution is not available, the dynamical behavior of the system must be investigated by simulation studies. But, the existence of solutions must be guaranteed. In mathematics, concepts to proof the existence and reachability of possible solutions are investigated. For this, the index of a system must be calculated.

1. Introduction

Representations of power systems with e.g. various synchronous machine models are available [1]. For multi–machine representations, only simple models of the generators are used, with different frames of reference. In Fig. 1 a schematic representation of a power system is shown, where the different machines are interconnected by a net. These interconnection conditions are given by Kirchhoff's law and Tellegen's theorem.

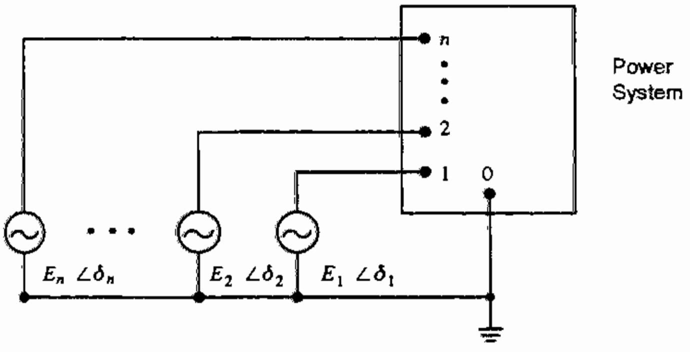


Fig. 1: Schematic representation of a power system

In the classical model of multi–machine model the synchronous generators are represented only by their swing equation. In general, every machine has to fulfill the power equilibrium

$$\frac{dW_{Ki}}{dt} + P_{Di} = P_{mi} - P_{gi} \qquad i = 1, 2, \ldots, n \tag{1.1}$$

from which the classical swing equation can derived immediately

$$M_i\frac{d^2\delta_i}{dt^2} + D_i\frac{d\delta_i}{dt} = P_{mi} - P_{gi} \qquad i = 1,2,\dots,n \tag{1.2}$$

The electrical equations for the network are given from the classical load flow equations. As a results one gets the complete representation of the classical multi–machine model

$$\begin{aligned} M_i\frac{d^2\delta_i}{dt^2} + D_i\frac{d\delta_i}{dt} &= P_i - P_{gi} \qquad i = 1,2,\dots,n \qquad \text{with} \\ P_i &= P_{mi} - E_i^2 G_{ii} \\ P_{ei} &= \sum_{\substack{j\neq i \\ j=1}}^{n}\left(C_{ij}\sin(\delta_i - \delta_j) + D_{ij}\cos(\delta_i - \delta_j)\right) \end{aligned} \tag{1.3}$$

These equations can be solved numerically, but they also have multiple solutions. With assumptions made by physics it is possible to detect the correct solution with physical meaning.

For a more general representation of a power system different degrees of abstraction in modelling are tested on their analytical and numerical solvability. Classical descriptions, different one– and multi–machine modelling concepts are investigated to show their well known solvability properties with these index concepts. The different index concepts are used also to test even compositions of more detailed nonlinear models of synchronous machines, transformers and transmission lines to multi–machine power systems with a more higher degree of modelling accuracy of the distinct models. Transferability conditions of these differential–algebraic systems to pure differential equational descriptions are tested.

In this paper the investigations are focused on a basic synchronous machine model which can be extended with more specific physical effects e.g. the magnetization of the core. Since the model has a "limit model" character [2], which means, that if the physical effect vanishes the mathematical model tends to a model without these effect, the synchronous machine model represents the simplest case of a more detailed model. The compact representation is very useful for the analysis.

2. Differential–Algebraic Equations

A general differential–algebraic equation (DAE) has the form [3]

$$F(\dot{x},\ x,\ t) = 0 \tag{2.1}$$

where $x(t) \in R \rightarrow R^n$, and $F \in R^n \times R^n \times R \rightarrow R^n$ is a function for which sufficient differentiability is assumed. The index, m, of (1.1) is defined as follows:

$m = 0$: If $\frac{\partial F}{\partial \dot{x}}$ is nonsingular, the index is 0. Under this condition, (2.1) can, in principle, be inverted into the explicit ODE $\dot{x} = f(x,\ t)$ so in this case (2.1) is called an implicit ODE.

$m > 0$: Otherwise, consider the system of equations

$$\begin{aligned} &F(\dot{x},\ x,\ t) = 0, \\ &\frac{dF}{dt} = \frac{\partial F}{\partial \dot{x}}x^{(2)} + \frac{\partial F}{\partial x}\dot{x} + \frac{\partial F}{dt} = 0, \\ &\frac{d^2F}{dt^2} = \frac{\partial F}{\partial \dot{x}}x^{(3)} + \cdots = 0 \\ &\dots \\ &\frac{d^sF}{dt^s} = \frac{\partial F}{\partial \dot{x}}x^{(s+1)} + \cdots = 0 \end{aligned} \tag{2.2}$$

as a system of equations in the *separate* dependent variables $\dot{x},\ x^{(2)},\ x^{(3)}, \dots,\ x^{(s+1)}$, and solve for these variables as functions of x and t considered as *independent* variables. Since $\frac{\partial F}{\partial \dot{x}}$ is singular, it will not be possible to solve for $x^{(s+1)}$, or possibly even for $x^{(q)}$ for smaller q. If it is possible to solve for $\dot{x}$ for some finite s, then the index, m, is defined as the smallest s for which (2.2) can be solved for $\dot{x}(x,\ t)$.

As discussed e.g. in Gear and Petzold [4] it is difficult to solve high index problems numerically [5, 6]. In fact, an index 2 example in Gear and Petzold [4] demonstrates that a problem in the form (2.1) can be solved in general only if the index does not exceed one.

The relations between general DAEs and semi–explicit DAEs are discussed very briefly. For more detail see e.g. [3,4,6].

To start with the idea, one can transform the general form (2.1) into the semi–explicit form by replacing $\dot{x}$ with v, then x with u to get

$$\dot{u} = v$$
$$0 = F(v,\ u,\ t) \tag{2.3}$$

which is a semi–explicit system. Its index is no more than one greater than the index m of (2.1) because of differentiation of (2.1) m times it is possible to solve for $v = \dot{u}(u,\ t)$, and with one additional differentiation one can compute

$$\dot{v} = \dot{v}(u,\ t) = \frac{\partial \dot{u}}{dt} + \left(\frac{\partial \dot{u}}{\partial u}\right)\dot{u}(u,\ t).$$

This equation can be reformulated by some substitutions [3] to get a system of index 1, which has the following autonomous form:

$$\dot{x} = f(x,\ y)$$
$$0 = g(x,\ y) \tag{2.4}$$

where $g_y = \frac{\partial g}{\partial y}$ has a bounded inverse in the neighborhood of the solution. Initial conditions $(x_0,\ y_0)$ are consistent which meant $g(x_0,\ y_0) = 0$ [7]. With the help of the implicit function theorem y can be solved from (1.4) $x : y = \hat{g}(x)$. With this expression one yields an ODE $\dot{x} = f\big(x,\ \hat{g}(x)\big)$.

In the next step systems of index 2 are represented by

$$\dot{x} = f(x,\ y)$$
$$0 = g(x) \tag{2.5}$$

where $g_x \cdot f_y = \frac{\partial g}{\partial x} \cdot \frac{\partial f}{\partial y}$ has a bounded inverse in the neighborhood of the solution. The time differentiation of the algebraic part leads to

$$0 = g_x(x) \cdot \dot{x} \qquad \Rightarrow \qquad 0 = g_x(x) \cdot f(x,\ y)$$

With consistent initial conditions $(x_0,\ y_0)$ which fulfills the algebraic equation the new DAE system is

$$\dot{x} = f(x,\ y)$$
$$0 = g_x(x) \cdot f(x,\ y) \tag{2.6}$$

Systems of the index 3 type have the structure:

$$\dot{x} = f(x,\ y)$$
$$\dot{y} = h(x,\ y,\ u) \tag{2.7}$$
$$0 = g(x)$$

In this case $g_x \cdot f_y \cdot h_u = \frac{\partial g}{\partial x} \cdot \frac{\partial f}{\partial y} \cdot \frac{\partial h}{\partial u}$ must have a bounded inverse. Under these conditions

$$0 = g_x(x) \cdot \dot{x} \qquad \Rightarrow \qquad 0 = g_x \cdot f$$
$$0 = g_{xx} \cdot \dot{x}^2 + g_x \cdot \frac{d^2x}{dt^2} \qquad \Rightarrow \qquad 0 = g_{xx} f^2 + g_x f_x f + g_x f_y h$$

The initial conditions $(x_0,\ y_0)$ must be consistent, also.

In general one can conjecture that the conversion between from the general to the semi–explicit form, and the reverse, always raises or lowers the index by one, respectively, except in the special case of a semi–explicit form that is already an explicit ODE, in which case there is no change of index [1].

3. Model of the Machine

A special kind of Park transformation, which is based on an idea of Fouad and Anderson [1], is applied to the well known mathematical description of the synchronous machine [2, 9] connected to an infinite bus. This transformation matrix is orthogonal and therefore the transformed machine model preserves the symmetric structure from the time–variant model. Moreover, the derived model is provided with an essential orthogonal structure. The new model of a synchronous machine is then given by a system of *explicit* differential equations where the currents are divided into two groups according to the d– and q–axis of the Park system.

$$i_1 := \left(i_d \;\; i_E \;\; i_D\right)^T \qquad\qquad i_2 := \left(i_q \;\; i_Q\right)^T$$

Furthermore, with this and with the possibility of defining some important linearly independent weighting vectors g, the system–matrix is partitioned into submatrices **A₁...A₄** compactly, where the **g**–vectors are multiplied by resistances respectively inductances of the machine.

$$A_1 := -\left(rg_1 \;\; r_E g_2 \;\; r_D g_3\right) \; A_2 := -\left(L_q g_1 \;\; kM_Q g_1\right) \qquad A_3 := \left(L_d g_4 \;\; kM_E g_4 \;\; kM_D g_4\right) \quad A_4 := -\left(rg_4 \;\; r_Q g_5\right)$$

As an infinite bus is assumed for this model, the system's only inputs are the mechanical torque $T_m$and the excitation voltage u_E. Due to the used transformation, the following mathematical model is in Park's coordinates:

$$\begin{aligned} \frac{di_1}{dt} &= A_1\, i_1 + \omega A_2 i_2 + g_1\, U \sin\delta - g_2 u_E & \frac{d\omega}{dt} &= i_1^T\, M\, i_2 - \frac{c}{J}\omega + \frac{1}{J} T_m \\ \frac{di_2}{dt} &= \omega A_3\, i_1 + A_4\, i_2 - g_4\, U \cos\delta & \frac{d\delta}{dt} &= \omega - \omega_R \end{aligned} \tag{3.1}$$

An explanation of the model parameters is given in the appendix of this paper. To provide the machine with an excitation power, the excitation voltage u_E has to be negative.

The g–vectors can also be found in the feedback loops from the mechanical into the electrical part and in the path of the excitation voltage u_E. Although the components of the g–vectors are expressions of inductances, they don't have any effect on the equilibrium points, which will be shown later on. The mechanical part of the system is given by the two equations on the right hand side of (3.1), in which the electrical torque is represented by the term $i_1^T M i_2$. $M$is called torque–matrix and contains the following entries:

$$M := \frac{1}{J} \begin{bmatrix} L_d - L_q & kM_Q \\ -kM_E & 0 \\ -kM_D & 0 \end{bmatrix} \tag{3.2}$$

A mechanical damping torque $T_d = c\omega$ is modelled to be proportional to the angular velocity of the shaft. In spite of using a linear magnetization curve for this model, the transformed description has a nonlinear character, which is caused by the feedback loops from the mechanical into the electrical part.

Remark

- These basic mathematical model of a synchronous machine in a compact representation can be extended by additional physical effect e.g. the magnetization of the core in such a way that the model tends to the basic model when the physical effect vanishes [2]. Therefore, this model is the basis of more detailed models. Discussing the index of the interconnected synchronous machines, the obtained results for the basic models are the simplest case [8].

4. Case Study: Two Interconnected Synchronous Machines

To analyze the properties of different interconnected synchronous machines the simplest case of only two machines, is assumed. In Fig. 2 the structure of these "network" is given:

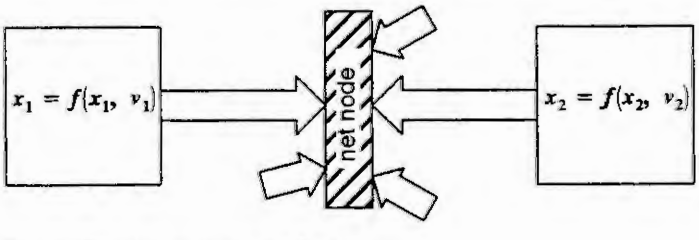


Fig. 2: Two interconnected Machines

The general machine model from Section 3 can formulated for the electrical and the mechanical part of a mathematical representation, respectively:

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} A_{11} & \omega_1 A_{12} & 0 & 0 \\ \omega_1 A_{13} & A_{14} & 0 & 0 \\ 0 & 0 & A_{21} & \omega_2 A_{22} \\ 0 & 0 & \omega_2 A_{23} & A_{24} \end{bmatrix} x - \begin{bmatrix} g_{12}\, u_{1E} + g_{11}\, u_{1d} \\ g_{14}\, u_{1q} \\ g_{22}\, u_{2E} + g_{21}\, u_{2d} \\ g_{24}\, u_{2q} \end{bmatrix} := f(x,\ y,\ u_E)$$

$$\dot{y} = \begin{bmatrix} \frac{C_1}{J_1} & 0 & 0 \\ 0 & \frac{C_2}{J_2} & 0 \\ 1 & -1 & 0 \end{bmatrix} y + \begin{bmatrix} \frac{1}{J_1} T_{1m} + i_{1d}^T\, M_1\, i_{1q} \\ \frac{1}{J_2} T_{2m} + i_{2d}^T\, M_2\, i_{2q} \\ 0 \end{bmatrix} := h(x,\ y,\ T_m) \tag{4.1}$$

$$0 = i_1 + i_2 - i_L = D\, x - i_L \qquad := g(x)$$

If $\omega_{1R} \overset{!}{=} \omega_{2R} \Rightarrow \dot{\delta}_{12} = \dot{\delta}_1 - \dot{\delta}_2 = \omega_1 - \omega_2$.

Boundary conditions given by the net are: $u_{1d} = u_{2d} = u_d = -\, U \sin\delta_{12}$

$u_{1q} = u_{2q} = u_q = U \cos\delta_{12}$

where U is the terminal voltage and $\delta_{12} = \delta_1 - \delta_2$ is the power angle.

This is a DAE system with the representation

$$\begin{aligned} \dot{x} &= f(x,\ y,\ u_E) \\ \dot{y} &= h(x,\ y,\ T_m) \\ 0 &= g(x) \end{aligned} \tag{4.2}$$

To calculate the index of the DAE system two different cases can be considered, where the input of the excitation system u_E respectively the mechanical input T_m from the governor is constant.

Differentiation of the algebraic equation $g(x)$ of the DAE system after time leads to a new algebraic equation of the form $0 = g_x\, f(x,\ y)$. With this a new DAE system is calculated:

$$\begin{aligned} \dot{x} &= f(x,\ y) \\ \dot{y} &= h(x,\ y,\ T_m) \\ 0 &= g_x\, f(x,\ y) \end{aligned} \tag{4.3}$$

According to part 2, in this paper the same notation as in Gears' paper [1] is instead of the Lie notation for vector fields.

A further differentiation of the algebraic equation of (4.3) yields the boundary condition
$0 = g_{xx}\, f^2 + g_x\, f_x\, f + g_x\, f_y\, h$. The obtained DAE system has now the structure of

$$\begin{aligned} \dot{x} &= f(x,\ y) \\ \dot{y} &= h(x,\ y,\ T_m) \\ 0 &= g_{xx}\, f^2 + g_x\, f_x\, f + g_x\, f_y\, h \end{aligned} \tag{4.4}$$

The condition that the DAE system of index 1 requires the existence and the constraints of

$$\frac{\partial\left(g_{xx}\, f^2 + g_x\, f_x\, f + g_x\, f_y\, h\right)}{\partial T_m} = g_x\, f_y\, h_{T_m}$$

With the parameters of the machines one get:

$$g_x\, f_y\, h_{T_m} = \begin{pmatrix} -a_{11}\left(L_{q1}\, i_{q1} + kM_{Q1}\, i_{Q1}\right) & -a_{12}\left(L_{q2}\, i_{q2} + kM_{Q2}\, i_{Q2}\right) \\ a_{71}\left(L_{d1}\, i_{d1} + kM_{E1}\, i_{E1} + kM_{D1}\, i_{D1}\right) & a_{72}\left(L_{d2}\, i_{d2} + kM_{E2}\, i_{E2} + kM_{D2}\, i_{D2}\right) \end{pmatrix}$$

The matrix is regular, if the determinant is not equal to zero. To make a statement about the regularity of the matrix, it is necessary to discuss distinct cases.

A: transient case: The determinant of $g_x\ f_y\ h_{T_m}$ is always regular $\Rightarrow$ system is of index 1

B: stationary case: for an equilibrium case the currents of the damper windings are equal to zero. As a result

$$\det\left(g_x\ f_y\ h_{T_m}\right) \neq 0 \qquad \textit{for} \qquad \begin{cases} a_{12}\ a_{71}\ L_{q2}\ \dfrac{kM_{E1}}{r_{E1}}\ u_{E1} \neq a_{11}\ a_{72}\ L_{q1}\ \dfrac{kM_{E2}}{r_{E2}}\ u_{E2} \\ \hline \dfrac{i_{q1}}{i_{q2}} \neq -\dfrac{a_{12}\ a_{71}\ L_{q2}\left(L_{d1}\ i_{d1} - \dfrac{kM_{E1}}{r_{E1}}\ u_{E1}\right)}{a_{11}\ a_{72}\ L_{q1}\left(L_{d2}\ i_{d2} - \dfrac{kM_{E2}}{r_{E2}}\ u_{E2}\right)} \end{cases} \tag{4.5}$$

The critical case is when two machines are identical. The conditions can be simplified to

$$\det\left(g_x\ f_y\ h_{T_m}\right) \neq 0 \qquad \textit{for} \qquad \begin{cases} u_{E1} \neq u_{E2} \\ \hline \dfrac{i_{q1}}{i_{q2}} \neq -\dfrac{L_{d1}\ i_{d1} - \dfrac{kM_{E1}}{r_{E1}}\ u_{E1}}{L_{d2}\ i_{d2} - \dfrac{kM_{E2}}{r_{E2}}\ u_{E2}} \end{cases} \tag{4.6}$$

Since the currents of machines are bounded $g_x\ f_y\ h_{T_m}$ is also bounded.
The DAE (4.4) is of index 1 if the condition (4.5) or (4.6) is valid. The system is solvable. But, the original DAE (4.2) is therefore of index 3.

$\Longrightarrow$ The original representation of the system (4.2) is of index 3.

A similar calculation can be done in the case when the input torque is constant and the excitation voltage as an input is assumed in the computation of the index. A similar result can be obtained as stated out above, the system is of index 3, also.

The differential–algebraic two–machine model of an electric power system is in the case of constant excitation voltages respectively constant torque input of index 3. The existence and uniqueness of solutions of such kind of systems cannot be guaranteed due to numerical problems in the computation process.

5. Conclusion

In this paper the index of a more general multimachine power system is computed. Since for such complex systems an analytical solution is not available, the dynamical behavior of the system must be investigated by simulation studies. But, the existence of a solution must be guaranteed. For analysis a basic mathematical model of a synchronous machine in a compact representation is used which can be extended by additional physical effect e.g. the magnetization of the core in such a way that the model tends to the basic model when the physical effect vanishes. Even in the simplest case, when only two machines are coupled together without any further nonlinear physical effect the index of the problem is 3. For this kind of mathematical representations solutions cannot be guaranteed.

Appendix

States:

i_d	direct axis synchronous current
i_q	quadrature axis synchronous current
i_E	field current
i_D	damping current of the d–axis
i_Q	damping current of the q–axis
ω	angular velocity of the shaft
δ	rotor angle or torque angle

Inputs:

u_E	excitation voltage
T_m	mechanical forcing torque

Infinite Bus:

ω_R	angular frequency of the infinite bus
U	voltage of the infinite bus

Parameters of the Machine:

J	inertia constant
c	damping constant
r	resistance of the stator windings
L_d	d–axis synchronous self–inductance
k	coupling factor
r_E	resistance of the field winding
r_D	resistance oft the damper winding of the d–axis
r_Q	resistance of the damper winding of the q–axis
L_q	q–axis synchronous self–inductance

M_D	mutual inductance between the damper winding of the d–axis and the stator windings
M_Q	mutual inductance between the damper winding of the q–axis and the stator windings
M_E	mutual inductance between the field winding a the stator windings
M_R	mutual inductance between the field winding and the damper winding of the d–axis

References

[1] Anderson, P.M.; Fouad, A.A.: "Power System Control and Stability", The Iowa State University Press, Ames, Iowa, U.S.A., Volume 1, Fourth Printing 1986

[2] Fette, M.: "Strukturelle Analyse elektrischer Energieversorgungssysteme", Fortschrittberichte VDI, Reihe 21, Nr. 140, VDI–Verlag, 1993

[3] Gear, C.W.: "Differential–Algebraic Equations Index Transformations", SIAM J. Sci. Stat. Comput., Vol. 9, No. 4, 1984, pp. 39 – 47

[4] Gear, C.W.; Petzold, L. R.: "ODE Methods for the Solution of Differential/Algebraic Systems", SIAM J. Num. Anal., Vol. 21, No. 4, 1984, pp. 716 – 728

[5] Hairer, E.; Lubich, C.; Roche, M.: "The Numerical Solution of Differential–Algebraic Systems by Runge–Kutta Methods", Lecture Notes in Mathematics, Vol. 1409, Springer Verlag, Berlin, Heidelberg, 1989

[6] Lötstedt, P.; Petzold, L.R.: "Numerical Solution of Nonlinear Differential Equations with Algebraic Contraints I: Convergence Results for Backward Differentiation Formulas", Mathematics of Computations, Vol. 46, No. 147, 1986, pp. 491 – 516

[7] Reich, S.: "Beitrag zur Theorie der Algebrodifferentialgleichungen", Dissertation, Technische Universität Dresden, 1990

[8] Guckenheimer, J.; Holmes, P.: "Nonlinear Oscillations, Dynamical Systems and Bifurcation of Vector Fields", Applied Mathematical Sciences, Vol. 42, Springer–Verlag, New York, Berlin, Heidelberg, 1990

[9] Abed, E.H.; Varaiya, P.P.: "Nonlinear oscillations in power systems", Electrical Power & Energy Systems, Vol. 6, No. 1, January 1984, pp. 37 – 43

MATHEMATICAL MODELLING OF FAILURE STATES OF INDUCTION MACHINES

R. Fišer and S. Ferkolj
Faculty of Electrical Engineering, University of Ljubljana
Tržaška 25, 1000 Ljubljana, SLOVENIA

Abstract. The paper presents the analysis of dynamic and steady-state performance of induction motor in different operational conditions. The mathematical model of induction motor with stator and/or rotor asymmetry is developed. Based on mathematical model, the computer simulation of the influence of rotor winding defect on the air-gap magnetic field distribution, torque and current characteristics of induction motor, is performed. Simulation represents the contribution to the correct evaluation of the measured data and is a powerful tool in training process and development of methods for condition monitoring and on-line diagnosis of electrical machines.

Introduction

It is expected that although there is a growth of new types of electrical motors (due mostly to new materials), ac drives will be dominant in the future. However, due to its low cost, reliability and ruggedness the cage induction motor will remain the preferred machine for most industrial drives [10, 11].

Because of technological and operational imperfections, electrical machines are never perfectly symmetrical. In a number of cases machines are also working in asymmetrical operational conditions. This facts made it necessary to study the operation of asymmetrical machines [1, 2, 3, 4, 9]. Manufacturing faults and severe operational conditions may cause rotor bars to break, causing poor starting performance, excessive vibration and increased thermal stresses. The detection of rotor imperfections is vital for effective condition monitoring on induction motors, and the ability to calculate the effects of simulated faults is vital to an understanding of the fault mechanisms.

The effectiveness of a digital computer in studying the performance of electrical machines is demonstrated with computer results which show the dynamic and steady-state behavior of 3-phase machines during balanced and unbalanced operation. The computer simulation for this various modes of operation is obtained from the equations which describe the symmetrical and asymmetrical induction machine. Of particular practical interest is the representation which simulate rotor electrical asymmetry [1, 2, 6, 8, 13].

Computer simulation in the diagnostic process

The modern electrical drives are characterized by the application of more and more complicated equipment. The electrical motor together with the load machine as well as supply driver and control systems are run to risk of various types of failures. The main advantage of the diagnostic system, which should be part of the drive, is that it can predict the possible breakdowns by analyzing various parameters of the drive, so in one hand it determines the date when the breakdown is expected and even which part of the machine will cause the breakdown. It is expected that in the future all professional motor drives will be equipped with diagnostic systems.

By continuous tracking of the health-state of the system, information are obtained on various forms of damage (rotor asymmetries, broken rotor bars, interturn short-circuits, several forms of asymmetrical supply, static and dynamic eccentricity) and the existence of potentially dangerous situations can be avoided by indicating at a very early stage the formation of these malfunctions and asymmetries. Research in this area has focused on sensing magnetic flux, stator current, shaft flux, vibrations and speed fluctuations [1, 6, 10].

The basis of any reliable diagnostic method is an understanding of the electric, magnetic and mechanical behavior of the machine in health-state and under fault conditions. The aim of computer simulation of operating characteristics is to foresee the changes of motor performance due to the changes of construction parameters as the consequence of different faults.

The disadvantage of the mathematical modelling and computer simulation is that it requires the knowledge of the controlled system by means of a set of algebraic and differential equations, which relate inputs and outputs analytically. The complete model describing the system is often too complex and, therefore, difficult or even impossible too calculate. In order to overcome these problems, it is possible to use artificial intelligence techniques, such as fuzzy logic, neural networks, genetic algorithms etc. This methods are recently showing a good promise for applications in parameter estimation, condition monitoring and diagnosis of electrical machines. The artificial intelligence methods applied as fault detectors of the drive system can be trained on the base of expert knowledge - the sets of data which determine the drive in various operation conditions. These data can be collected during the experiments as well as on the base of the mathematical model simulations and here computer simulation improves its great practical value.

Simulation of transient-state operation

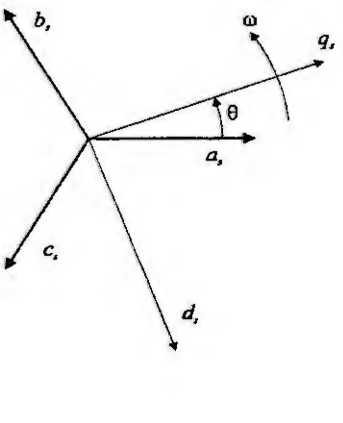
Most methods for rotor fault detection are usually effective under steady-state, full load operating conditions before reliable diagnostic can be carried out [1, 2, 3]. There are however situations where it is either impractical or undesirable to test machines under full load. At such situations the transient analysis of induction machine is more suitable for fault detection.

The developed mathematical model enables the computer simulation of various faults of induction machines. Specially attention was devoted to the study of unequal rotor winding resistance as the source of rotor electrical asymmetry. Equations (1) and (2) are derived by many different authors [5, 7, 8, 11] and represents the induction machine in natural (three phase) coordinate system.

$$\begin{aligned}[u_{abcs}] &= [R_s]\cdot[i_{abcs}] + \mathrm{p}[\Psi_{abcs}] \\ [u_{abcr}] &= [R_r]\cdot[i_{abcr}] + \mathrm{p}[\Psi_{abcr}]\end{aligned} \tag{1}$$

$$\begin{bmatrix}[\Psi_{abcs}]\\ [\Psi_{abcr}]\end{bmatrix} = \begin{bmatrix}[L_s] & [L_{sr}]\\ [L_{sr}]^T & [L_r]\end{bmatrix}\cdot\begin{bmatrix}[i_{abcs}]\\ [i_{abcr}]\end{bmatrix} \tag{2}$$

In equations (1) and (2) are u_s, i_s, Ψ_s the space phasors of the stator voltages, stator currents, and stator flux linkages, respectively, and u_r, i_r, Ψ_r are the space phasors of rotor voltages, rotor currents, and rotor flux linkages, respectively. In order to simplify the mathematical model, the equations established in the natural co-

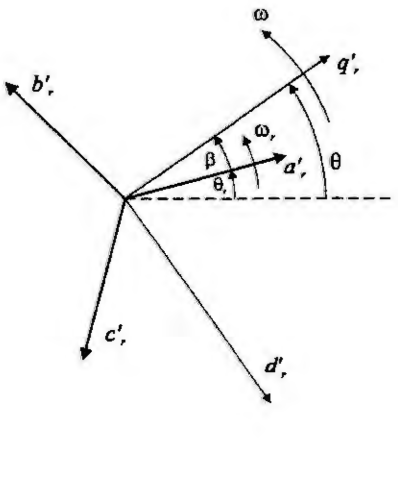


$$[f_{qd0s}] = [K_s]\cdot[f_{abcs}] \tag{3}$$

$$[K_s] = \frac{2}{3}\begin{bmatrix}\cos\theta & \cos\left(\theta-\frac{2\pi}{3}\right) & \cos\left(\theta+\frac{2\pi}{3}\right)\\ \sin\theta & \sin\left(\theta-\frac{2\pi}{3}\right) & \sin\left(\theta+\frac{2\pi}{3}\right)\\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{bmatrix} \tag{4}$$

$$\theta = \omega\cdot t + \theta(0) \tag{5}$$

Fig. 1. Transformation of stator variables from natural *abc* to two-axes *qd0*



$$[f_{qd0r}] = [K_r]\cdot[f_{abcr}] \tag{6}$$

$$[K_r] = \frac{2}{3}\begin{bmatrix}\cos\beta & \cos\left(\beta-\frac{2\pi}{3}\right) & \cos\left(\beta+\frac{2\pi}{3}\right)\\ \sin\beta & \sin\left(\beta-\frac{2\pi}{3}\right) & \sin\left(\beta+\frac{2\pi}{3}\right)\\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{bmatrix} \tag{7}$$

$$\beta = \theta - \theta_r = \omega\cdot t + \theta(0) - \omega_r\cdot t + \theta_r(0) \tag{8}$$

Fig. 2. Transformation of rotor variables from natural *abc* to two axes *qd0*

ordinates *abc* are transformed to the two-axes *qd0* system. With this procedure the mutual inductance between phases is eliminated. In equations (3) and (6) f represents either voltages, currents or magnetic linkages.

The induction motor with rotor asymmetry can be simulated with appropriate changes of the coefficients in resistance $[R]$ and inductance $[L]$ matrix. With incorporation of equation (9) in the expression (1) asymmetrical distribution of rotor currents can be achieved.

$$[K_r][R'_{add}][K_r]^{-1} = \begin{bmatrix} R'_{d11} & R'_{d12} & R'_{d13} \\ R'_{d21} & R'_{d22} & R'_{d23} \\ R'_{d31} & R'_{d32} & R'_{d33} \end{bmatrix} = \begin{bmatrix} \left(\frac{2}{3}R'_{d1}+\frac{R'_{d2}}{6}+\frac{R'_{d3}}{6}\right) & \left(\frac{\sqrt{3}}{6}R'_{d2}-\frac{\sqrt{3}}{6}R'_{d3}\right) & \left(\frac{2}{3}R'_{d1}-\frac{R'_{d2}}{3}-\frac{R'_{d3}}{3}\right) \\ \left(\frac{\sqrt{3}}{6}R'_{d2}-\frac{\sqrt{3}}{6}R'_{d3}\right) & \left(\frac{R'_{d2}}{2}+\frac{R'_{d3}}{2}\right) & \left(-\frac{\sqrt{3}}{3}R'_{d2}+\frac{\sqrt{3}}{3}R'_{d3}\right) \\ \left(\frac{R'_{d1}}{3}-\frac{R'_{d2}}{6}-\frac{R'_{d3}}{6}\right) & \left(-\frac{\sqrt{3}}{6}R'_{d2}+\frac{\sqrt{3}}{6}R'_{d3}\right) & \left(\frac{R'_{d1}}{3}+\frac{R'_{d2}}{3}+\frac{R'_{d3}}{3}\right) \end{bmatrix} \tag{9}$$

A state-variable matrix differential equations are given for the general case of rotor asymmetry in *qd0* system :

$$\begin{aligned} \left[u_{qdos}\right] &= \left[R_s\right]\left[i_{qdos}\right]+\omega\left[\Psi_{qds}\right]+\mathrm{p}\left[\Psi_{qdos}\right] \\ \left[u'_{qdor}\right] &= \left(\left[R'_r\right]+\left[K_r\right]\left[R'_{add}\right]\left[K_r\right]^{-1}\right)\left[i'_{qdor}\right]+\left(\omega-\omega_r\right)\left[\Psi'_{qdor}\right]+\mathrm{p}\left[\Psi'_{qdor}\right] \end{aligned} \tag{10}$$

$$\begin{bmatrix} \left[\Psi_{qdos}\right] \\ \left[\Psi'_{qdor}\right] \end{bmatrix} = \begin{bmatrix} [K_s][L_s][K_s]^{-1} & [K_s][L'_{sr}][K_r]^{-1} \\ [K_r][L'_{sr}][K_s]^{-1} & [K_r][L'_r][K_r]^{-1} \end{bmatrix} \cdot \begin{bmatrix} \left[i_{qdos}\right] \\ \left[i'_{qdor}\right] \end{bmatrix} \tag{11}$$

The simulation of dynamic behavior of induction motor requires simultaneous solution of electrical and mechanical differential equations (12), (14) and (15). Runge-Kutta integration algorithm was used in order to examine the effects of rotor asymmetry on the transient torque, stator current, rotor current and speed characteristics of induction motor.

$$\begin{aligned} u_{qs} &= \frac{R_s\,L'_r}{D}\Psi_{qs} - \frac{R_s\,L_m}{D}\Psi'_{qr} + \omega\,\Psi_{ds} + \mathrm{p}\,\Psi_{qs} \\ u_{ds} &= \frac{R_s\,L'_r}{D}\Psi_{ds} - \frac{R_s\,L_m}{D}\Psi'_{dr} - \omega\,\Psi_{qs} + \mathrm{p}\,\Psi_{ds} \\ u_{os} &= \frac{R_s}{L_{ls}}\Psi_{os} + \mathrm{p}\,\Psi_{os} \\ u'_{qr} &= \left(R'_r+R'_{d11}\right)\left(-\tfrac{L_m}{D}\Psi_{qs}+\tfrac{L_s}{D}\Psi'_{qr}\right)+R'_{d12}\left(-\tfrac{L_m}{D}\Psi_{ds}+\tfrac{L_s}{D}\Psi'_{dr}\right)+R'_{d13}\left(\tfrac{1}{L'_{lr}}\Psi'_{or}\right)+\left(\omega-\omega_r\right)\Psi'_{dr}+\mathrm{p}\,\Psi'_{qr} \\ u'_{dr} &= R'_{d21}\left(-\tfrac{L_m}{D}\Psi_{qs}+\tfrac{L_s}{D}\Psi'_{qr}\right)+\left(R'_r+R'_{d22}\right)\left(-\tfrac{L_m}{D}\Psi_{ds}+\tfrac{L_s}{D}\Psi'_{dr}\right)+R'_{d23}\left(\tfrac{1}{L'_{lr}}\Psi'_{or}\right)-\left(\omega-\omega_r\right)\Psi'_{qr}+\mathrm{p}\,\Psi'_{dr} \\ u'_{or} &= R'_{d31}\left(-\tfrac{L_m}{D}\Psi_{qs}+\tfrac{L_s}{D}\Psi'_{qr}\right)+R'_{d32}\left(-\tfrac{L_m}{D}\Psi_{ds}+\tfrac{L_s}{D}\Psi'_{dr}\right)+\left(R'_r+R'_{d33}\right)\left(\tfrac{1}{L'_{lr}}\Psi'_{or}\right)+\mathrm{p}\,\Psi'_{or} \end{aligned} \tag{12}$$

The instantaneous value of torque can be calculated from the general expression, where P is the number of pole pairs.

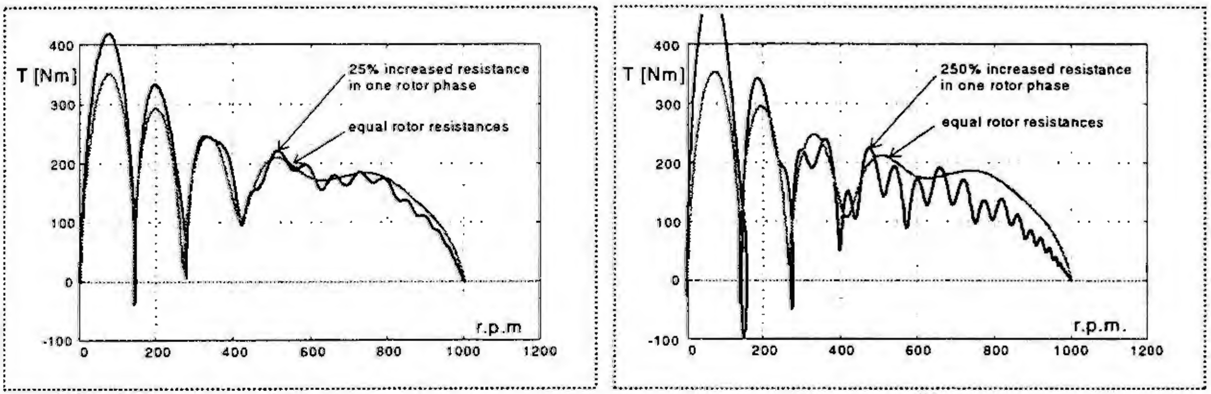
$$T_e = \frac{P}{2}\left(\left[K_s\right]^{-1}\left[i_{abcs}\right]\right)^T\left(\frac{\partial}{\partial\theta_r}\left[L'_{sr}\right]\right)\left[K_r\right]^{-1}\left[i'_{abcr}\right] \tag{13}$$

It is well known that the electromagnetic torque is produced by the interaction of the stator flux linkages and stator currents. It is more practical to solve the differential equations for the fluxes instead for the currents, so the torque equations are represented in the following form:

$$T_e = \frac{3}{2}\cdot\frac{P}{2}\cdot\frac{L_m}{D}\left(\Psi_{qs}\,\Psi'_{dr}-\Psi_{ds}\,\Psi'_{qr}\right) \tag{14}$$

$$T_e = J\frac{2}{P}\mathrm{p}\,\omega_r + T_{load} \tag{15}$$

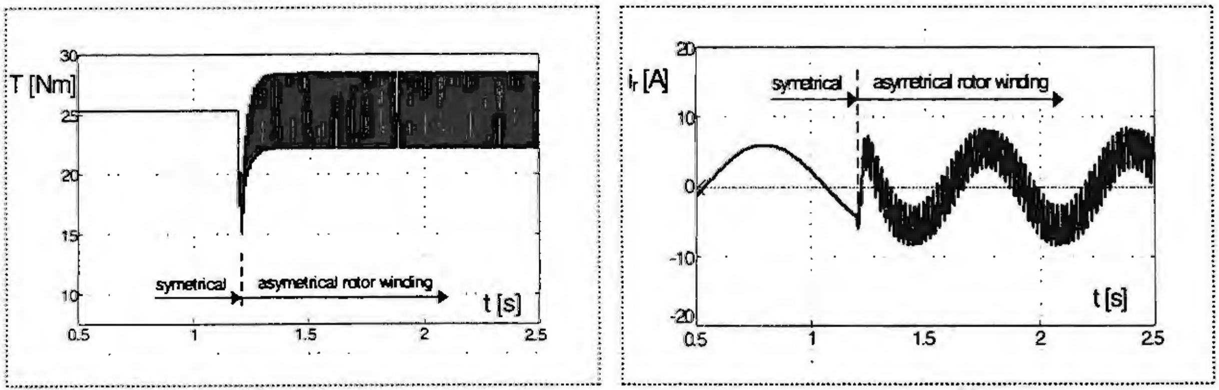
Fig3. represents the computed transient torque-speed characteristics of a 6 pole induction motor with asymmetrical rotor resistance in comparison with symmetrical rotor motor.



(a) (b)

Fig. 3. Transient torque-speed characteristics of an induction motor with asymmetrical rotor winding during start-up in comparison with healthy rotor motor.
(a) lower level of rotor asymmetry , (b) higher level of rotor asymmetry

Due to the increasing rotor asymmetry the level of pulsating torque and harmonics is increased (Fig. 3). Pulsating torque is responsible for the increase of motor vibration. From Fig.4 it can be easily noticed the increased level of pulsating torque when rotor asymmetry occurs.



(a) (b)

Fig. 4. Loaded induction motor - change from symmetrical to asymmetrical operation.
(a) motor torque; (b) rotor current

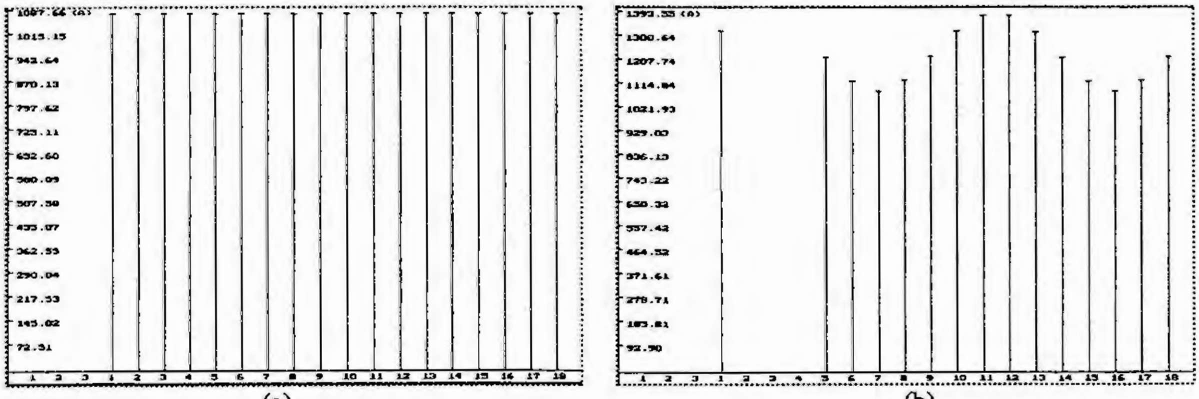
Simulation of steady-state operation

The air-gap field produced by a slip-frequency current flowing in a rotor bar has a fundamental component rotating at slip speed in the forward direction with respect to the rotor, and one of equal amplitude that rotates at the same speed in the backward direction. The field, which rotates at slip frequency backward with respect to the rotor, interacts with the rotor currents, induced by the forward rotating field to produce a torque variation at frequencies of higher harmonics, which is superimposed on steady output torque and results in increased noise, vibrations and as degradation of the induction motor performance.

The simulation of steady-state characteristics uses mathematical models, which has less simplifications in comparison with dynamic analysis, because the calculation of equations is not simultaneously. A computer program was developed for solving such complex set of equations with time dependent coefficients, including skin effect and higher harmonics taken into account. To model one or more broken bars, the bar current is set to zero by assigning very high value for bar resistance in equation (16). There is no restriction on the distribution of the broken bars so a very accurate study can be made. Not only the number of broken bars has the influence on motor performance, but also the position of faulty bars around the circumference of the rotor.

$$\left[\begin{array}{cc|cccc}
R_d + j\omega_r L_d & -\Omega L_d & \begin{array}{c} j\omega_r L_{dr}\cos(0\cdot\beta) - \\ -\Omega L_{dr}\sin(0\cdot\beta)\end{array} & \begin{array}{c} j\omega_r L_{dr}\cos(1\cdot\beta) - \\ -\Omega L_{dr}\sin(1\cdot\beta)\end{array} & \cdots & \begin{array}{c} j\omega_r L_{dr}\cos((N-1)\cdot\beta) - \\ -\Omega L_{dr}\sin((N-1)\cdot\beta)\end{array} \\
\Omega L_d & R_d + j\omega_r L_d & \begin{array}{c} j\omega_r L_{dr}\sin(0\cdot\beta) + \\ +\Omega L_d\cos(0\cdot\beta)\end{array} & \begin{array}{c} j\omega_r L_{dr}\sin(1\cdot\beta) + \\ +\Omega L_d\cos(1\cdot\beta)\end{array} & \cdots & \begin{array}{c} j\omega_r L_{dr}\sin((N-1)\cdot\beta) + \\ +\Omega L_d\cos((N-1)\cdot\beta)\end{array} \\
\hline
j\omega_r L_{rd}\cos(0\cdot\beta) & j\omega_r L_{rd}\sin(0\cdot\beta) & R_r + j\omega_r L_r & j\omega_r L_r\cos(1\cdot\beta) & \cdots & j\omega_r L_r\cos((N-1)\cdot\beta) \\
j\omega_r L_{rd}\cos(1\cdot\beta) & j\omega_r L_{rd}\sin(1\cdot\beta) & j\omega_r L_r\cos(1\cdot\beta) & R_r + j\omega_r L_r & \cdots & j\omega_r L_r\cos((N-2)\cdot\beta) \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
j\omega_r L_{rd}\cos((N-1)\cdot\beta) & j\omega_r L_{rd}\sin((N-1)\cdot\beta) & j\omega_r L_r\cos((N-1)\cdot\beta) & j\omega_r L_r\cos((N-2)\cdot\beta) & \cdots & R_r + j\omega_r L_r
\end{array}\right] \quad (16)$$

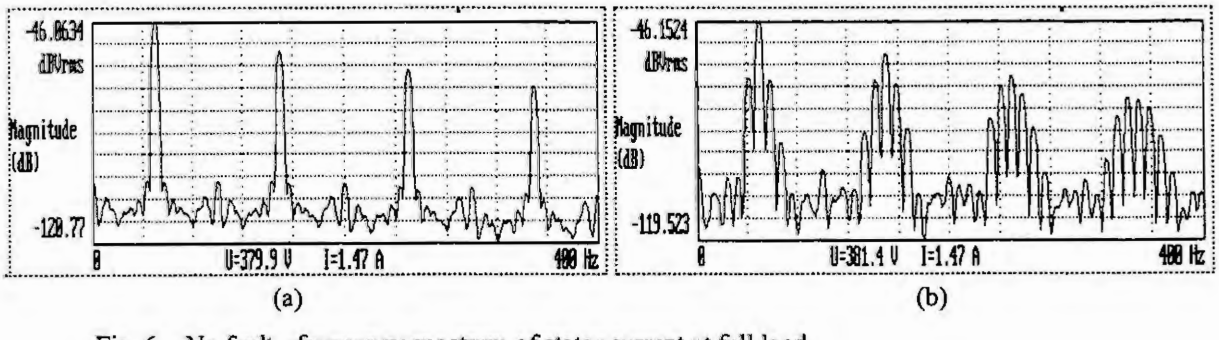
Fig. 5a represents rotor currents distribution of the healthy induction motor and Fig. 5b the one with three adjacent broken bars (number 2, 3, 4) at standstill operating condition. When the rotor cage is symmetrical, that is before a fault occurs, the bar currents are of equal amplitude, with a fixed phase progression around the rotor periphery, after the fault the current distribution becomes elliptic.



(a) (b)

Fig. 5. Current flow distribution in the rotor bars at starting (slip=1)
(a) no fault; (b) three rotor bars broken

It is well established that when rotor asymmetry occurs, rotor harmonic fluxes are produced [3, 10]. They induce currents in the stator at a frequency of $(1-2s)f_1$, where s is slip and f_l is the supply frequency. These currents manifest themselves in the frequency domain, as two sidebands $(\pm sf_1)$ around the supply frequency if rotor asymmetry is present [10, 11, 12].



(a) (b)

Fig. 6. No fault - frequency spectrum of stator current at full load
(a) no fault; (b) three rotor bars broken

On Fig. 6. it can be seen that rotor asymmetry increases the amplitude of the side-bands near the stator frequencies. The presence of side-band frequencies is the indication for damaged rotor cage.

Summary

The results of dynamic and steady-state simulation of induction machine presented in the paper has pointed out that the asymmetry of rotor winding due to arising fault causes the increasing value of higher harmonics in air-gap flux and stator current. Produced pulsating torque leads to the reduction of the reliability of the drive.

The simulation of induction machine dynamic behavior is of great practical importance. The equations have been established in an arbitrary reference frame which enables the observation of the motor quantities in different reference frames. The use of digital computer to demonstrate free acceleration, changes of load torque, breakdown, and other characteristics has proved advantageous. If we are able to build a complex mathematical model of an electrical machine with taking into account as many variable parameters as possible, we can simulate different types of malfunctions and study the changes of the operational characteristics without destroying expensive laboratory machines. Computer simulation represents the contribution to the correct evaluation of the measured data and is a powerful tool for monitoring the possible defects or emergency states and on-line diagnosis of electric machines.

References

1. Elkasabgy N. M., Eastham A. R., Detection of Broken Bars in the Cage Rotor on an Induction Machine, IEEE Transactions on Industry Applications, Vol. 28, No.1, January/February 1992, 165-171.
2. Ferkolj S., Fišer R., On-line Fault Diagnostic Techniques of Induction Motor Drives, Stockholm Power Tech, International Symposium on Electric Power Engineering, Stockholm, Sweden, 18.-22. June 1995, in Proceedings No. Electrical Machines and Drives, 162-166.
3. Filippetti F., Franceschini G., Tassoni C., Vas P., A Fuzzy Logic Approach to On-line Induction Motor Diagnostics Based on Stator Current Monitoring, IEEE PowerTech 95, Stocholm, Sweden, in Proceedings vol. Electrical machines and drives, 156-161.
4. Fišer R., Ferkolj S., Šolinc H., Steady State Analysis of Induction Motor with Broken Rotor Bars, Seventh International Conference on Electrical Machines and Drives, Durham, United Kingdom, 11.-13. September 1995, in IEE Conference Publication Number 412, 42-46.
5. Jereb P., Fundamental Transformations at Electric Machines Analysis, in Electrotechnical Review, Vol.4, Slovenia, 1979, 222-231.
6. Kliman G.B., Koegl R.A., Noninvasive Detection of Broken Rotor Bars in Operating Induction Motors, IEEE Transaction on Energy Conversion, Vol. 3, No. 4, December 1988, 873-879.
7. Krause P. C., Analysis of Electric Machinery, McGraw-Hill Inc., New York, 1986.
8. Krause P. C., Simulation of Symmetrical Induction Machinery, IEEE Transaction on Power Apparatus and Systems, Vol. Pas-84, No.11, November 1965, 1038-1053.
9. Thomson W.T., On-line Current Monitoring to Diagnose Mechanical Shaft Misalignment in Three Phase Induction Motor Drive Systems, ICEM'94, Paris, France, in Proceedings volume 2, 1994, 238-243.
10. P. Vas, Parameter Estimation, Condition Monitoring, and Diagnosis of Electrical Machines, Oxford, Clarendon Press, 1993.
11. P. Vas, Electrical Machines and Drives: a Space-Vector Theory Approach, Oxford, Clarendon Press, 1992.
12. Vas P., Vas J., Transient and Steady-State Operation of Induction Motors, Archiv für Elektrotehnik 59, 1977, 55-60.
13. Williamson S., Smith A.C., Steady-state Analysis of 3-phase Cage Motors with Rotor-bar and End-ring Faults, IEE Proceedings, Vol. 129, Pt. B, No. 3, May 1982, 93-100.

SIMULATION FOR CONTROL OF INTERCONNECTED POWER SYSTEM

M.U. Bogatyriev
Tula State University
Lenin avenue, 92 Tula 300600 Russia
E-mail mbog@tulgtu.tula.su

Abstract. Non-stationary non-real model for analyzing processes and control of Electrical Thermal Furnaces is designed. Simulating results reflecting complicated process behaviour are presented. Symmetry analysis and model decomposition is discussed. A way to operate process using interconnected control signals is suggested.

Introduction

This work is concerned with simulation for control in the area of metallurgy. Electrical Thermal Furnaces (ETF) have been applying there for steels and compound alloys manufacturing. A furnace consisted of three electrodes imbedded into bath (reaction zone) is considered . There is three phase voltage power system on the furnace and every electrode is connected to corresponding phase of A, B, C. There are voltage of 220 volts and currents up to 100 kiloamperes (KA) on the furnace.

The main ETF control problem is to keep phase currents as near to equal values as possible by changing voltage separately in each phase or by moving electrodes in the reaction zone. Every such control action causes changing of all current values in the furnace.

There is one very dangerous and inefficient mode of furnace performance when the current of one of phases has the maximum value, in several times exceeding the value of current of another phase. Such mode is named as Mad and Dead Phases mode [2].

There are industrial ETF current regulators which work in according to negative feedback principle compensating the deviation between current value of a current of each phase and its required value by moving an electrode of the same phase.

Such way of control is quite appropriate when current deviations and electrodes moving are not great. If they have significant values, this way is inefficient and can turn furnace into Mad and Dead Phases mode.

Model description

There are two general ways of modelling Electrical Thermal Furnaces for control. The first one is traditional for control systems and lies in applying input-output or state space models. Another way is using models describing physical properties of the process. Here namely the model of that second type was designed to try to apply ETF process's peculiarities for control . The model reflects interdependence of phases, and also internal symmetry of ETF. It is consisted of following equations. First, the model of electrode servos is in the form of linear differential equations:

$$\left.\begin{aligned} T_1\frac{d^2r_1(t)}{dt^2}+\frac{dr_1(t)}{dt}+r_1(t)&=K_1h_1(t)\\ T_2\frac{d^2r_2(t)}{dt^2}+\frac{dr_2(t)}{dt}+r(t)_2&=K_2h_2(t)\\ T_3\frac{d^2r_3(t)}{dt^2}+\frac{dr_3(t)}{dt}+r(t)_3&=K_3h_3(t)\end{aligned}\right\} \quad (1),$$

where $r_1(t)$, $r_2(t)$, $r_3(t)$ - active resistance of phases, $h_1(t)$, $h_2(t)$, $h_3(t)$ - control signals to every electrode moving, T_i, K_i, i=1,2,3 - equations parameters. Initial conditions for resistance as $r_i(0)= r_{i0}$ are assumed. These equations constitutes all model dynamics since servos are most inertial mechanical parts of the ETF in comparison of its electrical part.

Electrical part of the ETF model is described in following non-real form:

$$\left.\begin{aligned}
I_1(t) &= \left\{ \frac{\left[(\frac{1}{2}\gamma + \alpha) r_2(t) + (\frac{1}{2}\beta + \alpha) r_3(t) + \frac{\sqrt{3}}{2}(\gamma x_2 - \beta x_3) \right]}{A(t)} + \right. \\
&\left. + \frac{i\left[(\frac{1}{2}\gamma + \alpha) x_2 + (\frac{1}{2}\beta + \alpha) x_3 - \frac{\sqrt{3}}{2}(\gamma r_2(t) - \beta r_3(t)) \right]}{A(t)} \right\} U_m \\
I_2(t) &= \left\{ \frac{\left[\frac{1}{2}(\gamma - \beta) r_1(t) - (\frac{1}{2}\beta + \alpha) r_3(t) + \frac{\sqrt{3}}{2}(\beta + \gamma) x_1 + \frac{\sqrt{3}}{2}\beta x_3 \right]}{A(t)} + \right. \\
&\left. + \frac{i\left[\frac{1}{2}(\gamma - \beta) x_1 - (\frac{1}{2}\beta + \alpha) x_3 - \frac{\sqrt{3}}{2}(\beta + \gamma) r_1(t) - \frac{\sqrt{3}}{2}\beta r_3(t) \right]}{A(t)} \right\} U_m \\
I_3(t) &= \left\{ \frac{\left[\frac{1}{2}(\beta - \gamma) r_1(t) - (\frac{1}{2}\gamma + \alpha) r_2(t) - \frac{\sqrt{3}}{2}(\gamma + \beta) x_1 - \frac{\sqrt{3}}{2}\gamma x_2 \right]}{A(t)} + \right. \\
&\left. + \frac{i\left[\frac{\sqrt{3}}{2}(\gamma + \beta) r_1(t) + \frac{\sqrt{3}}{2}\gamma r_2(t) + \frac{1}{2}(\beta - \gamma) x_1 - (\frac{1}{2}\gamma + \alpha) x_2 \right]}{A(t)} \right\} U_m
\end{aligned}\right\} \qquad (2)$$

Here $I_1(t)$, $I_2(t)$, $I_3(t)$ - currents of phases, U_m - mutual phase voltage, α, β, γ - voltage reduction coefficients for each phase A, B, C respectively; x_1, x_2, x_3 - reactive resistance of phase, $i = \sqrt{-1}$,

$$\begin{aligned}
A(t) &= r_1(t)r_2(t) + r_1(t)r_3(t) + r_2(t)r_3(t) - x_1x_2 - x_1x_3 - x_2x_3 + \\
&+ i[x_1(r_2(t) + r_3(t)) + x_2(r_1(t) + r_3(t)) + x_3(r_1(t) + r_2(t))]
\end{aligned} \qquad (3)$$

Mad and Dead Phases mode simulation

It is considered that Mad and Dead Phases mode is a result of asymmetric state of the ETF, caused by different values of resistance of phases [2]. Non-real model (1) - (2) helps to learn what resistance, - active or reactive ones, - play the main role in Mad and Dead Phases mode appearance. Simulations were made by adding model (1) - (2) with feedback for each phase and using module calculations for non-real values since only real values of currents have been measuring on the ETF. All experiments were performed in parametric hyper cube which corresponds with possible natural values of resistance's were practically measured.

Following are several results which demonstrate performance of the ETF controlled by standard regulators. It is found that those regulators work quite well under significant active resistance's asymmetry and asymmetry in regulator's parameters (unequal T_i and unequal K_i) but reactive resistance's asymmetry causes complex behaviour with Mad and Dead Phases.

For simplicity following results were all made under identical conditions of symmetry for active resistance and regulator's parameters except the different value of reactive resistance in one phase A. The range of the reactive resistance's difference is denoted as d. The required current value in each phase was 50 KA.

The Figure 1 shows stable Mad and Dead Phases which are not disappearing when controlling by regulators. The ETF has great asymmetry of d=10.

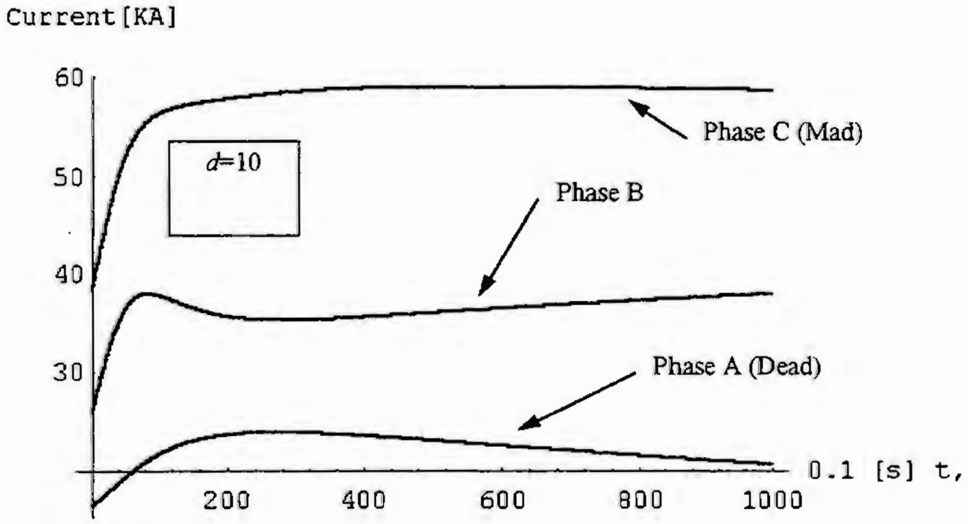


Figure 1.

Simulating continuous reducing of d, we have following results. Figure 2 shows that control system could not regulate currents even with asymmetry of d<10 . It turns ETF to another stable point in its state space with {0,0,0} coordinates. Practically here all electrodes are moving all up till electrical circuit breaking and currents disappearing.

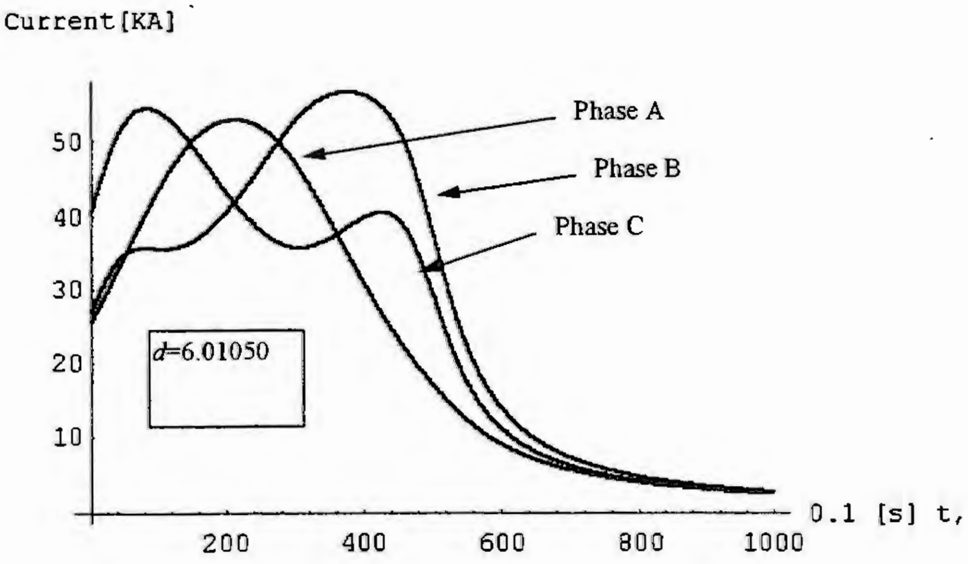


Figure 2.

There was found the value of d , very close to previous one, which changes the situation as it is shown on Figure 3.

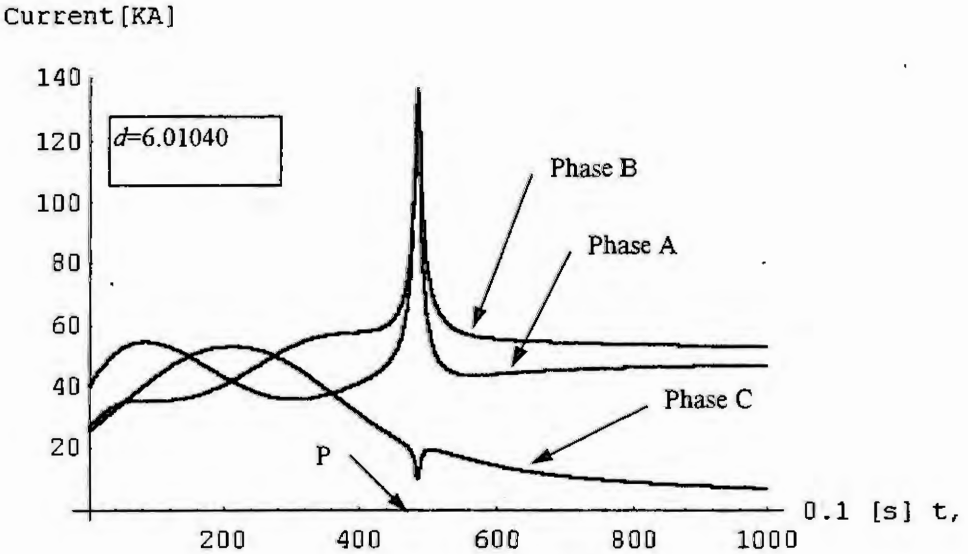


Figure 3.

Note however that in this case control system regulates currents but with great static errors. There exists critical point P of electrodes position where unstable Mad and Dead Phases occur. In practice that kind of Mad and Dead Phases is unexpected and very dangerous. Reducing d farther, we have quite precise regulation of currents has shown on Figure 4.

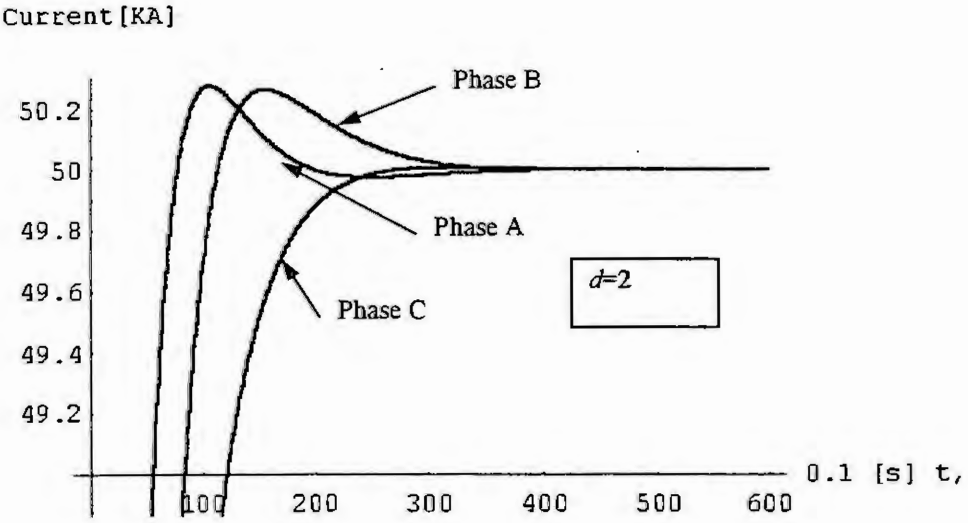


Figure 4.

All those results demonstrate the necessity of another way to control currents than by standard regulators. Control signals must be interconnected. One way to realise this feature is using ETF's model symmetry in control.

Model symmetry

We consider a model having symmetry if its operator $\mathcal{F}(\mathbf{r},t,{}^{d}\!/_{dt})$ satisfies commutation conditions

$$\mathbf{T}(g_i)\,\mathcal{F}(\mathbf{r},t,{}^{d}\!/_{dt})=\mathcal{F}(\mathbf{r},t,{}^{d}\!/_{dt})\,\mathbf{T}(g_i), \tag{4}$$

where matrices $\mathbf{T}(g_i)$ constitute reducible representation of a group G: $\{g_i, i=1,2,\dots,n\}$, named symmetry group for that model . Every $\mathbf{T}(g_i)$ matrix produces those permutations in $\mathcal{F}(\mathbf{r},t,{}^{d}\!/_{dt})$ operator that leave it of the same form. Here we assume G as a discrete group and according to [3] it always can be realized as a set of permutations matrices $\mathbf{T}(g_i)$.

It is known that three-phase systems have cyclic symmetry. Physically such symmetry is caused by process of electromagnetic energy carrying between phases. Rotating electromagnetic field is the carrier of the energy. There is connection between non-real voltages in three-phase system as

$$E_C=e^{-i\frac{2\pi}{3}}E_A,\quad E_B=e^{-i\frac{4\pi}{3}}E_A,$$

where $e^{-i\frac{2\pi}{3}}$ is an operator of rotation on the angle of ${}^{2\pi}\!/_{3}$.

We confirm that the model (1) - (2) has the same symmetry of cyclic group of $\mathbf{C}_3$ as any three-phase system. It can be found not only by checking conditions (4) with equations (1) - (2) but also by simulating. The $\mathbf{C}_3$ group has representation by permutations in the following form

$$T(g_1=e)=\begin{array}{ccc}1&0&0\\0&1&0\\0&0&1\end{array},\qquad T(g_2)=\begin{array}{ccc}0&0&1\\1&0&0\\0&1&0\end{array},\qquad T(g_3)=\begin{array}{ccc}0&1&0\\0&0&1\\1&0&0\end{array} \tag{5}$$

Assuming that the $\mathcal{F}(\mathbf{r},t,{}^{d}\!/_{dt})$ operator maps inputs $\mathbf{h}=\{h_1(t),\ h_2(t),\ h_3(t)\}$ to outputs $\mathbf{I}=\{I_1(t),\ I_2(t),\ I_3(t)\}$ we can consider matrices (5) as sets of input signals $h_i(t)$, i=1,2,3 for electrodes. Realising every matrix from (5) on the model (1) - (2) we shall obtain a matrix of nine step responses . If $I_A(t)$, $I_B(t)$, $I_C(t)$ - corresponding step responses in phases A, B, C, then a matrix of step responses for unit steps (5) has following structure for any matrix from (5):

$$\begin{array}{ccc}I_A(t)&I_B(t)&I_C(t)\\I_C(t)&I_A(t)&I_B(t)\\I_B(t)&I_C(t)&I_A(t).\end{array} \tag{6}$$

Matrix (6) has cyclic symmetry of $\mathbf{C}_3$ group. Simulating results shown on Figure 5 represent possible first row of matrix (6) for absolute parametric symmetry on ETF. They also demonstrate one important property of currents control by standard regulators.

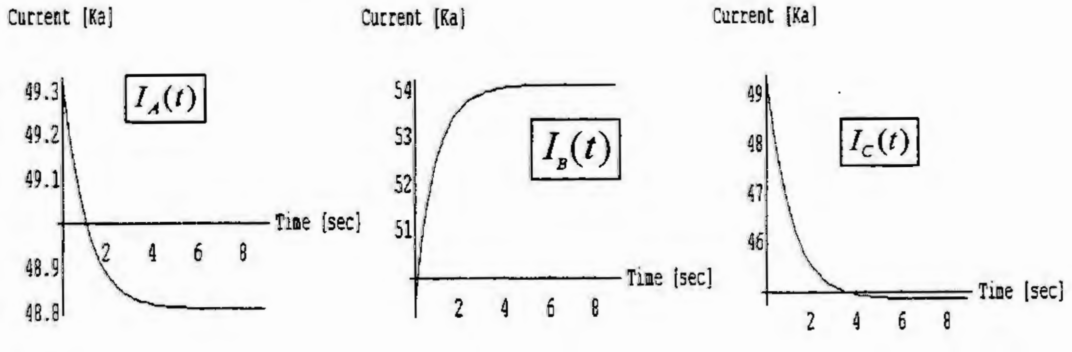


Figure 5

If electrode of phase A moves up to decrease the $I_A(t)$ current of phase A then the current $I_C(t)$ of phase C decreases more than $I_A(t)$ and the current $I_B(t)$ in the phase B even increase. That bad property has known from practice and is resulted from ETF's symmetry.

Model decomposition for control

If an operator $\mathcal{F}(\mathbf{r}, t, {}^{d}\!/_{dt})$ has group of symmetry, then it has a decomposition transformation for its coordinates which changes its structure to block-diagonal form. Canonical basis for this decomposition is the same basis where group representation has its irreducible form [3]. So decomposition symmetric model can be found by solving standard task of reducing group representation [4]. That task does not need calculations for matrix eigenvalues.

For the group of $\mathbf{C}_3$ there is non-real matrix transformation to canonical basis of its representation:

$$\mathbf{M} = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 \\ 1 & \varepsilon & \varepsilon^2 \\ 1 & \varepsilon^2 & \varepsilon^1 \end{bmatrix}, \tag{7}$$

where $\varepsilon = e^{-i\frac{2\pi}{3}}$. A theoretical way to realise interconnected currents control on the ETF is to apply transformation (7) to control signals $\tilde{\mathbf{h}} = \mathbf{M} \quad \mathbf{h}$. If such new signals $\tilde{\mathbf{h}}$ can be practically realised, they provide a 'real-time' decomposition of the ETF to control each phase current independently from others. The problem is that it is needful a model which describes how complex resistance of $z_j(t) = r_j(t) + ix_j(t)$ (j=1,2,3) is varying when each electrode is moving.

Summary

1. The model (1) - (2) can be used for ETF control system's behaviour simulation.
2. Results have shown on Fig. 1 - Fig.4 need careful learning and interpreting by modern dynamic systems analysis as possible example of bifurcation in non-linear system.
3. Some practical 'asymmetric' features of ETF 's performance (Fig. 5) can be explained from its symmetry analysis.

References

1. Bogatyriev M.U. Computer Experiment for the Interconnected Dynamic System Symmetry Analysis. In: Proc. Int. Conference on Algebraic and Analytical Methods in the Differential Equations Theory, Oryel State University , 1996, 47-49.
2. Danzis Y.B. Methods on Electrical Synthesis for Electrical Thermal Furnaces. Energya, Leningrad, 1973. (in Russian)
3. Hamermesh M. Group Theory and its Application to Phisical Problems. Addison-Wesley, London, 1964.
4. Wigner E.P. On the Matrices which Reduce the Kronecker Products of Representations of S.R. Groups, Princeton,1951

FUNCTIONAL MODELLING AND SIMULATION IN MECHANICAL DESIGN AND MECHATRONICS

J. Lückel and J. Wallaschek
Heinz Nixdorf Institut,Universität-GH Paderborn
Fürstenallee11, 33102 Paderborn, Germany

Abstract. The design of mechanical and mechatronic systems is of great importance. Especially in the context of the development of new products, the ability to design correctly cost-effectively and „in time" is a necessary (but not sufficient) condition for success.

In the past, many attempts were made to optimize design processes. Simultaneous engineering, concurrent engineering and many more strategies were coined. Nevertheless, their impact on the design practice has not been overwhelming, because these methods aimed only at speeding up the design process without improving on quality. Most products are still designed by trial and error and most design processes are still organized in a step by step form with functional design and geometric design being separated.

The authors of the present paper believe that it is possible to drastically improve design quality by using functional modelling and simulation in combination with geometric design. The present state of the art in geometric design is well advanced. Functional design, however, is still in an early stage and needs to be improved considerably. The main difficulty lies in the fact that the role of modelling is not well understood. The present paper will give a general view at modelling in the context of mechatronic systems.

Mechatronic systems

The basic structure of a mechatronic system is shown in Fig.1. In a physical sense, mechatronic systems combine a basic mechanical system, sensors, actuators and processors. The main idea is to process the information obtained by the sensors and to use the actuators to actively improve the overall behaviour of the basic system. Examples of mechatronic systems are manifold, including

- antilock braking system, airbag system, active suspension in automotive applications,
- CD-player, walkman, autofocus camera, camcorder and other systems in entertainment,
- autonomous transportation systems and robots in production engineering,
- magnetically levitated rotors, systems for the active vibration damping of tools and many more applications in manufacturing machines.

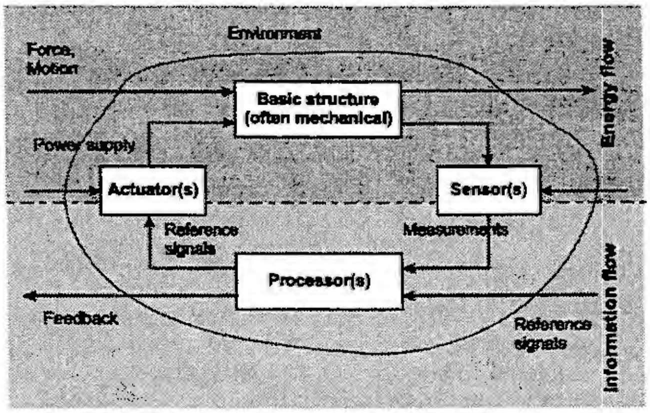


Fig.1: Basic structure of a mechatronic system.

The notion of „function“ in machine design and mechatronics

In machine design theory the term „function“ is used as a synonym for the specification of the principal tasks that a machine or the subsystem of a machine has to fulfill [1]. This description is usually static and given in natural language. In the present paper, the term „mechatronic function“ is used in a different meaning, insofar as it means a description of the kinematic, dynamic and control behaviour of a mechatronic system. In summary, mechatronic function can in most cases be considered as a synonym for a desired controlled motion behaviour of a system. Functional modelling in the sense of mechatronics includes a temporal aspect and is given in a concise mathematical form.

Many basic problems in mechatronical design can be studied by considering the task of developing a new milling tool like the one shown in Fig.2. Some important features of such machines are their robustness against disturbances, their positioning speed and accuracy. Therefore that the design goals will be formulated in terms of these features of the machine.

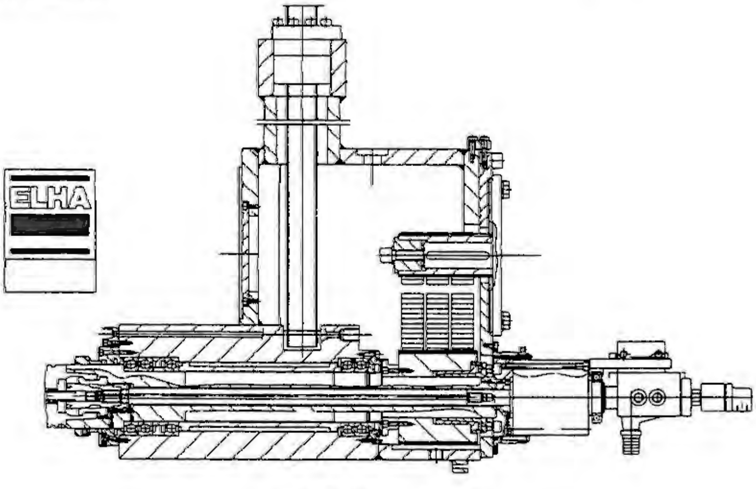
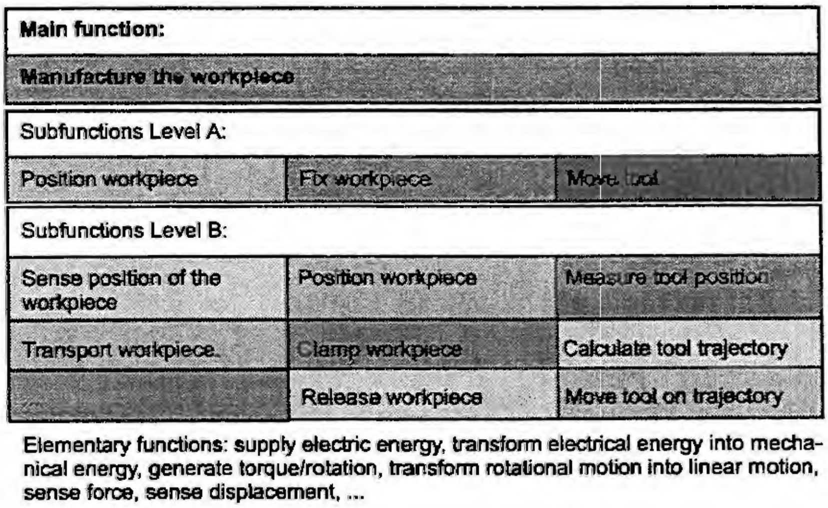


Fig.2: Milling tool.

Let us for the moment concentrate on the drive and tool system of the machine. A functional structure in the sense of machine design theory is given in Fig.3. Although this functional model provides a good hierarchical structuring of the machine system, it allows neither a judgement on whether the design goals are reached by a specific functional design nor a comparison different design alternatives with respect to their relative performance.



Main function:		
Manufacture the workpiece		
Subfunctions Level A:		
Position workpiece	Fix workpiece	Move tool
Subfunctions Level B:		
Sense position of the workpiece	Position workpiece	Measure tool position
Transport workpiece.	Clamp workpiece	Calculate tool trajectory
	Release workpiece	Move tool on trajectory

Elementary functions: supply electric energy, transform electrical energy into mechanical energy, generate torque/rotation, transform rotational motion into linear motion, sense force, sense displacement, ...

Fig.3: Functional structure in the sense of machine design theory of the drive and tool system of a milling tool.

A functional model in the sense of mechatronics should include the most important parameters of the system so that their relative influence on the design goals can be studied. It should also allow to formulate the design goals explicitly. In the ideal case the design engineer should be able to study the behaviour of the system, in all relevant aspects, by using the functional model. In this sense the functional model can be considered as a formal representation of the interactions and constraints existing in a mechatronic system that allows to answer certain design questions by (analytic or numeric or even qualitative) reasoning.

Partial models in geometric and functional design

In the design, the mechanical engineer will most probably start with a sketch of the main mechanical components of the system. He will concentrate on the proper choice of stages, ball bearings, gear wheels and other parts of the system. Most of his model will be based on geometric properties of the system. Soon he will realize that some very important information that is necessary for a proper design is not available at this stage. For example he will not be able to specify the time history of the loading as long as he has not made a decision on the electric motor that is used to drive the stages and the tool of his machine, nor will he be able to specify the robustness of the drive system against variations in workpiece stiffness or other disturbances, as long as the control system has not been taken into account. And finally he will not be able to specify the accuracy of his machine before the sensors and the measurement signal processing have been specified. It is obvious that the mechanical engineer's model of the system is only a partial model which is far from being sufficient for the design problem at hand.

The control engineer on the other hand will start with a sketch of the control structure of the system. One possible starting design could be Fig.4. In contrast to the mechanical engineer's „geometric world“, he will „live“ in a hierarchically organized block-oriented world. In his analysis of the problem, however, he will soon realize that he cannot properly design his control system without proper knowledge of the stiffness of the gears and bearings, accuracy of the stages and resolution of the measurement system. He will need appropriate transfer functions for the mechanical and electrical components of the system and other important information which is not included in his partial model of the system.

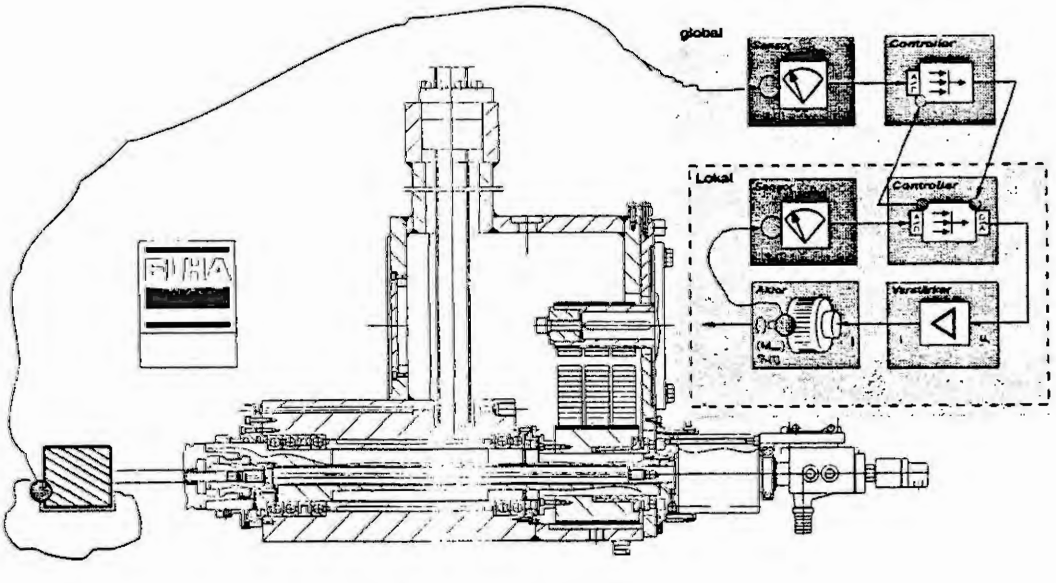


Fig.4: Milling tool and its main components as seen by a control engineer.

It is obvious that an appropriate model for the design task at hand must include the partial model of the mechanical engineer as well as the partial model of the control engineer and, maybe, also further partial

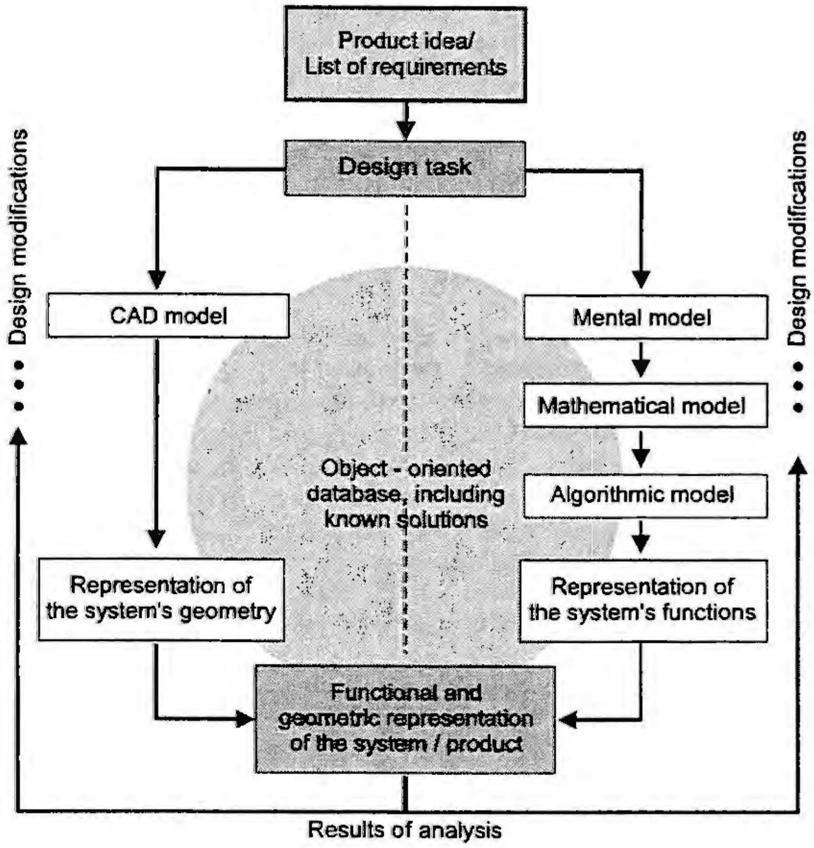

Product idea/
List of requirements
Design task
CAD model
Mental model
Mathematical model
Algorithmic model
Object - oriented database, including known solutions
Representation of the system's geometry
Representation of the system's functions
Functional and geometric representation of the system / product
Results of analysis
••• Design modifications
••• Design modifications

Fig.5: The process of geometric and functional modelling and simulation in the context of product development as proposed by the authors.

When deriving mental models for mechatronic systems it is evident that the structure and form of the mental models can be manifold depending on the engineering disciplines involved. A typical feature of mechatronic systems is that the energy-level description as well as the information level description are equally important. Thus mental models used in the functional modelling of mechatronic systems must include the energy and the information level.

The mental model of a design problem can be regarded as a very efficient way of knowledge representation, because the mental model contains all information relevant for the design problem and implicitly contains all relations and constraints between the design variables. Often the real value of an appropriate mental model is not that it allows the derivation of equations of motion and the subsequent simulation of the behaviour of the system, but that it gives a concise description of the basic laws behind the underlying design problem. Mental models are equivalent to engineering knowledge in the following sense: As a rule, in the context of machine design several phenomena are observed and - at the beginning - not well understood. Then investigations are made in order to find out the laws governing these phenomena. A mental model is developed, which allows to explain the observed phenomena in terms of well-understood basic physical laws. Finally - maybe after several iterations - the mental model which has been derived can be used as a formal representation of the real world and becomes part of engineering knowledge.

The derivation of mental models for mechatronic systems

It has already been pointed out that many different mental models may be appropriate for one and the same system, depending on the problem and questions at hand. Mental models are needed for the dimensioning of machine parts, they are required in the design of a control system for a specific task, they are used for the representation of experimentally observed phenomena and needed in many more contexts. Most often the quality of the mental models determines the quality of the results of an analysis. It is therefore useful to study the process of deriving mental models in some detail.

STEP 1: Structuring the overall system

Complex mechatronic systems are composed of many subsystems which interact with each other. There are numerous methods of structuring mechatronic systems. For example, techniques of object-oriented software design have been used with good success [4]. In this case several hierarchical structures can be used, such as „part-of" or „kind-of" relationships. One of the first steps in the process of modelling mechatronic systems is the modular-hierarchical structuring of the overall system into subsystems. The subsystems may then be subdivided into sub-subsystems and so on, until the elementary subsystems have been defined. Note that at each level of this subsystem hierarchy the principal structure of mechatronic systems, as shown in Fig.1, can be observed. Fig.6 shows one possibility of structuring the milling tool.

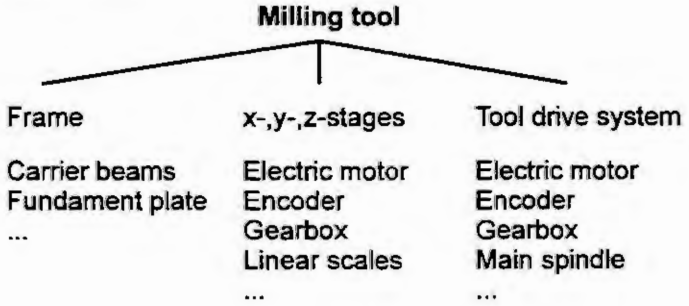


Fig.6: Structuring of the milling tool into subsystems and components.

STEP 2: Determining the overall system model structure

When the process of structuring the overall system into subsystems has been completed, one could - at least in principle - immediately start to derive mental models for the subsystems. The difficulty in deriving „good" mental models is that we are usually not able to evaluate the „quality" of mental models for a certain subsystem without knowing the mental models for the rest of the system. Before starting the process of deriving mental models for the subsystems a concise definition of the modelling aims for the overall system and of the level of detail for each subsystem should be made. This is necessary to avoid inconsistencies between the mental models of the subsystems.

At this point of the analysis all modelling requirements must be transparent. It should for example be clear for which purpose the final model will be used. Depending on the later usage of the model, different model structures will be obtained. The appropriate level of detail of the different subsystems is directly influenced by this. In the case of our milling tool the resulting overall model might be used for a strength analysis of the drive train. In this case a high level of detail for the mechanical parts of the drive train, e.g. the main shaft, will result while a rough model of the electrical motor is already sufficient and only some few loading conditions will be considered. The shaft might be modelled by finite elements taking into account changes of the cross-sectional area as well as material properties of the shaft and the electrical motor together with its power supply might be modelled by a PI-element. If the model was used for a thermal analysis of the electric motor, a rough model of the shaft would be sufficient while a high level of detail of the electrical system would be required. The shaft and drive train would then be modelled as a spring-mass system with few degrees of freedom, while the electric motor and its power supply should be modelled by complex differential equations taking into account the nonlinear behaviour of the electrical components.

It should, again, be clear that there are no universal models which are optimal for all kinds of functional analysis.

STEP 3: Balancing partial models from different disciplines

In mechatronics there is a very complex interaction between different subsystems and also between the partial models used in the design process. As we have already pointed out this creates the problem that the individual subsystems are not independent of each other. Neither are the partial models of different engineering disciples independent of each other. This holds for the overall system level as well as for the level of each individual subsystem.

In the case of our milling tool a „mechanical model" of the drive train is needed in the design of the control system. The design of the control system is to a large extent influenced by the mechanical model. If the dynamics of the overall system is not represented precisely enough, a „wrong" controller will be designed. In the positive case this will only lead to a suboptimal performance of the overall system. The aim of designing robust controllers is motivated by the fact that usually not all dynamic effects of the system and all possible disturbances can be modelled adequately. The robustness of the controller compensates for this lack of model representation. In the negative case, however, the result might be an instability of the overall system. Numerous cases of control- and observer-spillover are reported in the literature. The phenomenon of spillover is a typical consequence of an inappropriate balancing of partial models.

By considering the frequency bandwidth of the different subsystems and by trying to obtain coherent modelling within the respective bandwidth a balanced overall system model can be derived. Part of a balanced overall system is the appropriate description of the interaction between subsystems. This can be done using force-coupling, motion-coupling and several other methods. Another important point is the modelling of sensor and actuator location. It is well known that sensor and actuator placement determine observability and controllability of the system. Therefore special care must be taken that these features are modelled correctly [5].

STEP 4: Deriving mental models for the subsystems

Mental models for functional modelling can - in principle - be obtained in two different ways:

1. Starting from the most general description of the behaviour of the system, the appropriate mental model is derived by successive simplifications. This reduction process is governed by mathematical considerations as well as by engineering expertise.
2. Starting from elementary mental models, the appropriate mental model is composed by combining the elementary models to more complex ones.

Both ways of deriving mental models have their relative merits and disadvantages.

At present, there exist many powerful modelling tools which support the derivation of (partial) mental models. Here again it should be clear that fundamental assumptions such as the way how interaction between subsystems was modelled have a pronounced effect on the quality of the results.

STEP 5: The integration of subsystem models

It has already been pointed out that the overall mental model of a mechatronic system is composed of mental models of subsystems while partial models from different engineering disciplines are used simultaneously. In order to understand the behaviour of the overall system and to study and optimize its dynamic performance, these partial models must be integrated into an overall model, taking into account the mutual interaction between the subsystems. This is a crucial step in the modelling process. Only if steps 2 and 3 have been carried out successfully, we can expect all subsystem models to have the same level of detail and their relative input and output ports to correspond with each other. Quite often, unfortunately, this is not the case because the subsystem models were derived without consideration of the overall model structure. Then much time and effort must be spent in order to make the subsystem models consistent with each other.

The craft of integrating lies in finding a data model that allows the exchange of information between subsystems and a simulation of the overall system. The art of integration, however, is to tune the subsystem models to each other, in such way that a coherent description of the overall system is achieved and a balance is found that allows an understanding of how the subsystems interact and how they mutually influence each other.

Here again the very nature of the process is iterative. During the past years many attempts have been made to develop tools to support this step of modelling [6].

There are many tool coupling concepts and data models which can be used to support the craft of integration and it is only a question of time before powerful integrated design tools will become available and used in the design process. This, however, is only a necessary condition for improving the design process. Although it might be disappointing enthusiasts in the community, the existence of software for the integration of partial models will by itself not lead to better solutions, nor will it help to create better products. This is due to the fact that the integration of different subsystem models to form an overall system model - if it is done in the sense of the craft of integration - is useful only in the analysis of the system behaviour. The design engineer is capable to analyze different system concepts in shorter time than before and - given the same time for analysis - can compare more solutions than before. He will, however, not gain anything in the synthesis of complex systems as long as he does not succeed in drawing the right conclusions from simulation results. The experienced engineer is well aware of the fact that simulation results by themselves are difficult to interpret and that the danger of trying to find explanations even for wrong results may not be neglected. In this context further insight in the dynamics of the system is extremely useful. Fortunately besides pure simulation there are many other methods to gain these information, such as eigenvalue analysis based on the linearized equations, energy flow analysis and many more. Computer algebra methods have been employed successfully in many cases [7].

Self-similarity in mechatronic systems

One difficulty in the process of modelling and designing mechatronic systems lies in the fact that typically there are different levels of system description. With regard to a milling tool we may think of the milling tool itself as being a mechatronic system. It is composed of different subsystems, most of them also being mechatronic systems such as the drive train, the robot arms and grippers of the handling system, and so on. On the other hand the milling tool might be part of an assembly line which can also be regarded as a mechatronic system. A structuring into different levels of information processing, as shown in Fig.7, is equally possible. With each level of abstraction the behavior of the system as well as the design goals for the systems become more and more cognitive in contrast to the purely reactive behavior of the underlying mechanical processes. In most design problems more than one of these different levels must be considered at the same time. It is therefore extremely useful to use techniques of hierarchical, block-oriented structuring for the overall system as well as for all its subsystems and components.

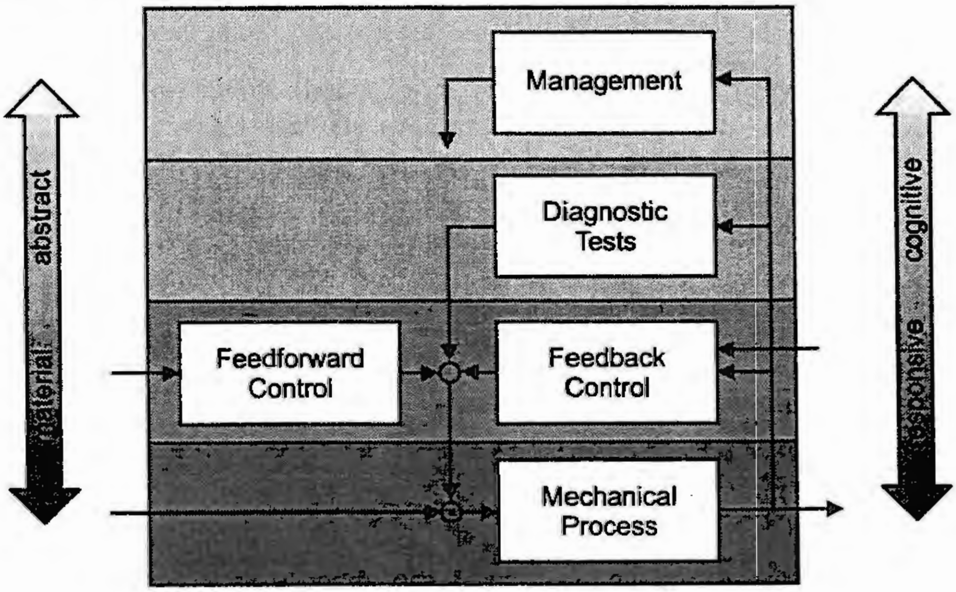


Fig.7 Multi-level information processing in complex mechatronic systems

If a complex mechatronic system is structured into subsystems and the subsystems are structured into sub-subsystems and so on, the question arises if there is a smallest „atomic" system structure on the lowest level of description. We claim that in mechatronics the structure of the basic „atomic" building block is exactly the same as that of the overall system. In other words: mechatronic systems are self-similar. The basic structure already shown in Fig.1 can be found on all levels of description. Systems with this structure are called „mechatronic function modules" [8,9].

This self-similarity of mechatronic systems must be taken into account if modelling and design paradigms for mechatronic systems are discussed. In the context of functional modelling this leads to the postulate that a complete functional model of a mechatronic system includes the mathematical description of the input-output behavior, including kinematical, dynamical and information quantities.

Summary and outlook

Functional modelling of mechatronic systems is a crucial step in the design of systems. In the present paper an attempt was made to systematize the process of functional modelling, which is characterized by different levels of abstraction. It was suggested to distinguish the following steps: structuring of a large system into ever smaller subsystems, determination of an overall model structure, balancing of partial models from different disciplines, derivation of mental models for all subsystems and, finally, integration of subsystem models. The step of formulating mental models was discussed in detail.

References

[1] Koller, R. : Konstruktionslehre für den Maschinenbau. Springer Verlag.

[2] Wallaschek, J. : Modellierung und Simulation als Beitrag zur Verkürzung der Entwicklungszeiten mechatronischer Produkte. VDI-Bericht, Nr. 1215, 1995.

[3] Bub, W. ,Lugner, P. : Systematik der Modellbildung; Teil 1: Konzeptionelle Modellbildung; Teil 2: Verifikation und Validation in: Modellbildung für Regelung und Simulation. VDI-Berichte 925, Langen, 25./26. März 1992.

[4] Hahn, M. ; Richert, J. ; Seuss, J. : Mechatronic object-oriented modelling and control strategies for vehicle convoy driving. In : [9]

[5] Wittler, G. , Moritz, W. , Schütte, H. : Integration of Mechatronic and Structural Design Methods and Design Tools for a Highly Dynamic Robot System: International Conference on Recent Advances in Mechatronics, Preprints. Istanbul 14.-16. August 1995.

[6] Wittler, G. , Moritz, W. : Die Ausstattung von gestaltorientierten CAD-Modellen mit dynamischen Eigenschaften im Sinne des Mechatronik. Zwischenbericht des MLaP im Rahmen des DFG-Schwerpunktsprogramms „ Innovative rechnergestützte Konstruktionsprozesse: Integration von Gestaltung und Berechnung". Paderborn, Juni 1996.

[7] Fuchssteiner, B. : Computeralgebra. In : Teubner-Taschenbuch der Mathematik. B.G. Teubner, 1995.

[8] Hesse, H. ; Wallschek, J. : Optimization of the dynamic behaviour of a wire bounder using the concept of mechatronic function modules. In : [9].

[9] Lückel, J. (Ed.) : From design methods to industrial applications. Proceedings of the 3rd Conf. On Mechatronic and Robotics, October 4-6, 1995, Paderborn. B.G. Teubner.

PARAMETRIC HYDRAULIC VALVE MODEL INCLUDING TRANSITIONAL FLOW EFFECTS

N. Mittwollen, T. Michl and R. Breit
Robert Bosch GmbH, D-70049 Stuttgart
Tel.: +49 711 811 6946, Email: mittwoll@fli.sh.bosch.de

Abstract. A parametric model for a spring controlled hydraulic poppet valve with a chamfered seat is proposed. Non-linear effects of the transition from laminar attached to turbulent separated flow, occurring at low Reynolds numbers around 500, on flow rate and flow forces are taken into account in a consistent way by simple algebraic expressions. These expressions are based on Reynolds principle of similarity, and they shall possess general validity. They contain three non-dimensional parameters, which can easily be determined for a broad range of operating conditions by a limited amount of measurements for a given valve design. The model is applied to a hydraulic system simulation, where cavitation besides transitional flow effects significantly determine the dynamic system characteristic. The simulation results are verified by a comparison with measured pressure transients.

Introduction

Due to the complex nature of fluid flows, the modelling and behavioural simulation of automotive hydraulic systems still remains a formidable challenge, especially in the higher frequency region (f > 1 kHz). Strong non-linearities in the governing conservation equations of mass and momentum, cavitation effects, wave propagation, laminar, transitional and turbulent flow, discontinuities, and stiff system equations have to be tackled within reasonable integration time limits. A popular approach is the use of tabulated flow data from stationary measurements or three dimensional CFD simulations of single hydraulic components. For a given system, accurate results can be achieved with this approach.

However, if the hydraulic system characteristic is to be improved over a wide range of operating conditions by an optimization of valve parameters, such as spring preload and stiffness or maximum valve lift, more generally valid parametric models should be used. Integrating a three dimensional flow model into the behavioural simulation is possible in principle and would provide a complete set of parameters, but is impracticable due to unreasonable demands on computation times. Thus, an appropriate model has to be derived that contains only those parameters that are needed in the optimization process, plus those describing the operating conditions or the special valve design (e.g., valves with conical or spherical poppets or disc valves). To find such models, much insight into the typical flow effects is required, as can be found in [5,7] and as is summarized in [3]. The principles of similarity laws can be applied to get generally valid expressions.

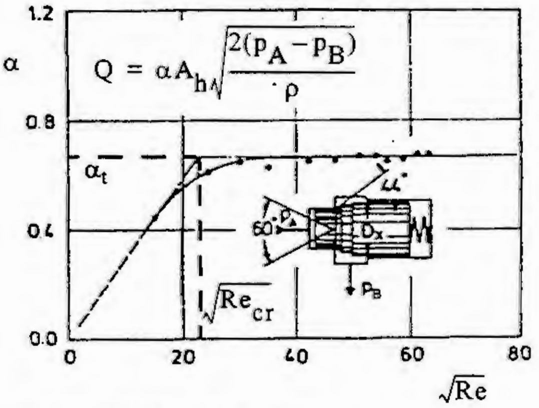


Figure 1: Orifice flow rate [4].

A well known example is the orifice flow rate characteristic (see Figure 1) [2,4]. It contains two generally valid non-dimensional parameters: the flow coefficient α, which depends on the Reynolds number Re of the flow through the orifice. This model can be used for determining the flow rate of a hydraulic valve.

The valve characteristic is significantly influenced by flow forces. They are usually taken into account only by an estimation of the momentum force of the inlet flow and of the turbulent separated jet flow, assumed to be exiting the valve seat [1,3,5,8]. This is insufficient for valves with chamfered seats. For operating conditions at low Reynolds numbers below or around 500, the flow at the narrow seat gap can be laminar and completely attached or transitional as well. A very simple approach to account for the corresponding transitional flow effects is the use of an effective stiffness [6]. Since its validity is rather limited, the authors propose a more general transitional flow force model for poppet valves, which is analogous to the above mentioned flow rate model.

Hydraulic System

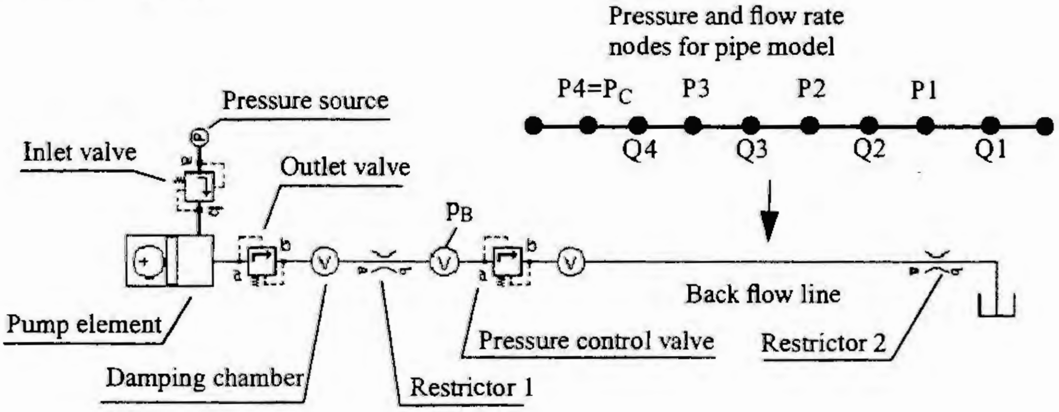


Figure 2: Hydraulic system model (Bath*fp*)

The investigated hydraulic system, displayed in Figure 2, is used as a low cost compact pressure source with the supply pressure p_B. The hydraulic fluid is delivered into the system by a single piston reciprocating pump with inlet and outlet check valves. A high volumetric efficiency is guaranteed by a preloading pressure source. The pulsating pressure transients are coarsely smoothed by the compressibility of the fluid in the hydraulic damping chamber in combination with the restrictor 1 orifice. The level of the operating system pressure p_B is determined by the pressure control valve. An extended back flow line delivers the redundant fluid to a reservoir, which is open to the atmosphere.

Parametric Model for the Pressure Control Valve

The pressure control valve is a spring controlled poppet valve with a ball as the closing body and a chamfered seat. The proposed simulation model for this valve (Figure 3 left) takes into account the dependence of the cross-sectional orifice area (simplified expression)

$$A_h = \frac{\pi}{2}h\left(D + h\sin\frac{\gamma}{2}\right)\sin\gamma \qquad (1)$$

on the valve lift h (γ = seat chamfer angle, D = ball diameter) for computing the flow rate Q with the orifice equation

$$Q = \alpha(\mathrm{Re})A_h\sqrt{2\Delta p/\rho}, \qquad (2)$$

where $\Delta p = p_{inlet} - p_{outlet}$ is the pressure difference over the valve, and ρ is the fluid density. The flow rate coefficient $\alpha(\mathrm{Re})$ is allowed to change linearly with the square root of the Reynolds number $\mathrm{Re} = Q\, d_h/A_h\nu$ of the flow through the orifice for $\mathrm{Re} < \mathrm{Re}_{cr}$, as shown in Figure 1. Here, ν is the kinematic fluid viscosity, and $d_h = 2\ h\ \sin(\gamma/2)$ is the hydraulic diameter of the cross-sectional orifice area. The parameters to be fitted to flow rate measurements or to CFD simulations are the flow rate coefficient α_t for turbulent separated flow conditions and the critical Reynolds number Re_{cr}, identifying the transition from laminar to turbulent flow. A smoothing function is used in the transition region.

All relevant static and dynamic forces, acting on the ball, are to be considered as well. Thus, the equation of motion

$$m\ddot{x} + d_D\dot{x} + cx = F_h - F_0 \qquad (3)$$

with the valve ball mass m and lift x = h, the damping coefficient d_D, the spring stiffness c and preload F_0 is used for the movement of the valve ball (Figure 3 left). The hydraulic force F_h can be determined by carrying out a hydraulic force balance on a control volume around the valve ball, considering pressure and momentum forces (Figure 3 right):

$$F_h = \underbrace{A_p(p_{inlet} - p_{outlet})}_{\text{pressure force } F_p} + \underbrace{\rho Q(v_{inlet} - v_{outlet}\cos(\gamma/2))}_{\text{momentum force } F_m} \tag{4}$$

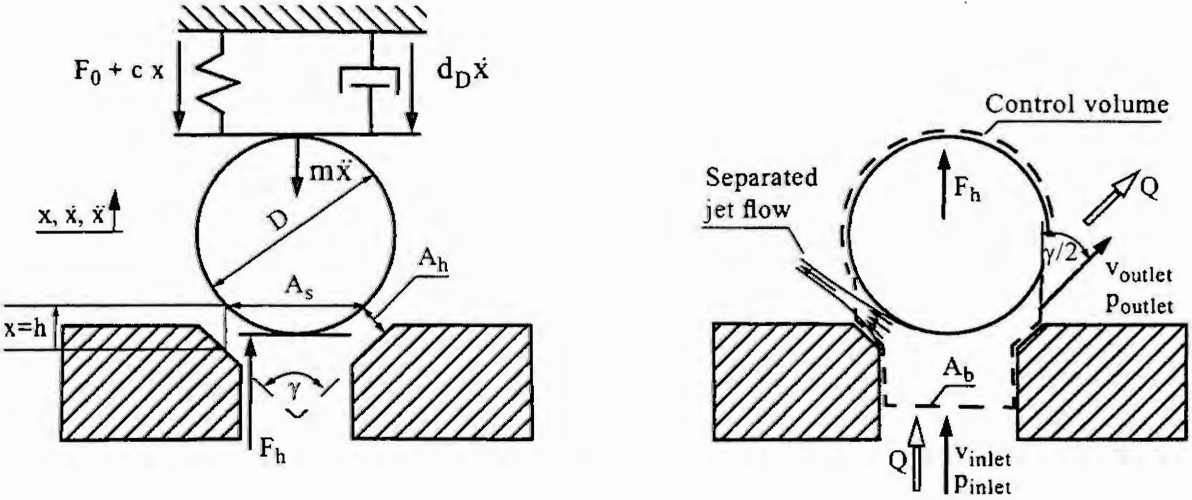


Figure 3: Control valve model (left) and hydraulic force control volume (right)

The flow enters the control volume with an average velocity of

$$v_{inlet} = Q/A_b \tag{5}$$

and leaves it with the velocity

$$v_{outlet} = \sqrt{2\Delta p/\rho}, \tag{6}$$

estimated by Bernoulli's equation [1,3,5]. The area A_p that the pressure difference Δp is acting on is usually set to a constant value (e.g., to the cross-sectional area $A_b = (\pi/4)D_b^2$ of the inlet bore for sharp-edged seats [3,8], or to $A_s = (\pi/4)D^2\cos^2(\gamma/2)$ for chamfered seats, Figure 3 left).

This model is insufficient, because it assumes constant pressures p_{inlet} over the area A_p and p_{outlet} over the remaining areas. However, unknown pressure distributions, strongly varying with flow conditions, develop in the valve seat region and significantly influence the force balance and thus the correct value of the hydraulic force F_h. The authors propose to take this effect into account by making the area A_p (now called *effective pressurized area*), and thus the pressure force F_p as well, depend on these flow conditions, represented by the seat flow Reynolds number Re, in a similar way as the flow coefficient α does:

$$A_p = \begin{cases} A_l + \sqrt{Re/Re_{cr}}(A_t - A_l) & \text{for } Re \le Re_{cr} \text{ (laminar flow)} \\ A_t & \text{for } Re > Re_{cr} \text{ (turbulent flow)} \end{cases}. \tag{7}$$

For zero flow rate (Re = 0), i.e., for a closed valve, the above mentioned assumption of constant pressures over the two regions, separated by the seat's theoretical sealing line, is correct. In that case, the effective pressurized area A_p is equal to

$$A_l = \max(A_s, A_b), \tag{8}$$

whereas for a turbulent flow $Re > Re_{cr}$, A_p is set to

$$A_t = f_p \cdot A_l . \tag{9}$$

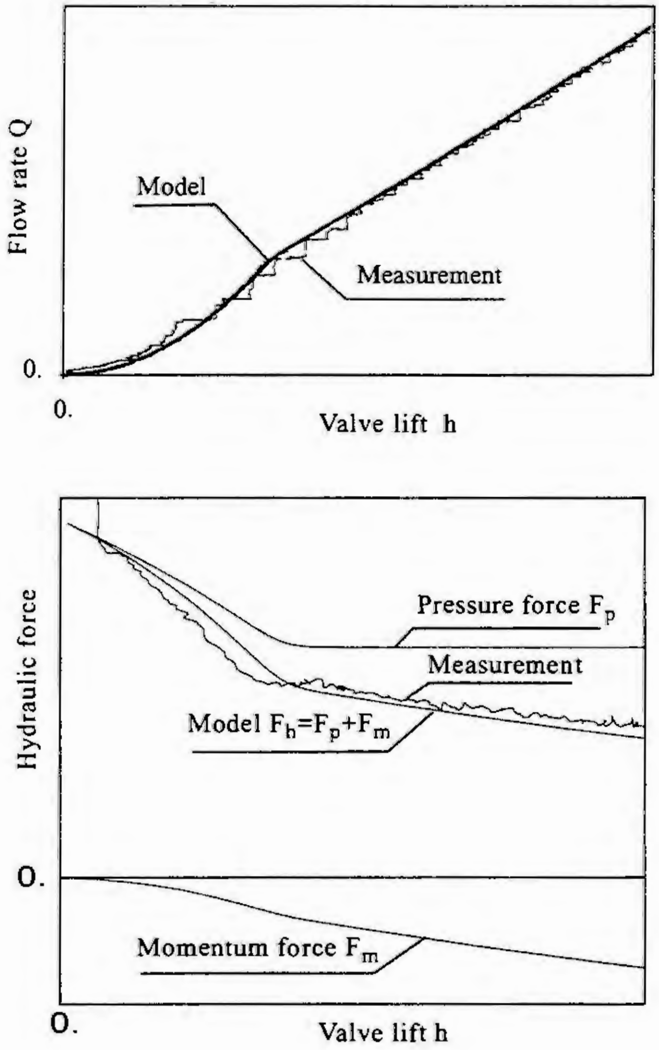


Figure 4: Comparison of measured and calculated valve flow rate and hydraulic forces (Δp = constant).

One additional non-dimensional parameter to be fitted to experimental data is the *turbulent effective pressurized area factor* f_p ($0 < f_p < 1$), describing the influence of the relative pressure distribution in the valve seat gap. Fitted values for f_p can be considerably below 1 due to relative sub-pressures, developing in the seat gap [5,7]. This parameter is mainly influenced by the valve geometry. Equation (7) provides for a continuous transition of the effective pressurized area A_p between the corresponding values for the two states of zero flow (only the hydrostatic pressure force is effective), i.e., A_l and turbulent separated flow, i.e., A_t, respectively. This transition is found to follow the same square root dependency on the seat flow Reynolds number Re as the flow coefficient $\alpha(Re)$ follows (Figure 1). A comparison of model calculations with measurements, where operational ball lifts h were always smaller than the seat chamfer length, demonstrates this relationship (Figure 4).

The reason for the change of the effective pressurized area A_p at low Reynolds numbers Re up to the critical value Re_{cr} is that the relative pressure distribution in the seat gap is determined by the strongly changing ratio of momentum to viscous forces in the flow, represented by the Reynolds number [11]: At very low Reynolds numbers, viscous forces dominate in the flow, and the pressure drops from p_{inlet} to p_{outlet} near the narrowest seat gap and is almost constant to either p_{inlet} or p_{outlet} around it. With an increasing Reynolds number, the rising influence of momentum forces in the flow make this pressure drop move towards the seat inlet, in addition to making substantial relative sub-pressures develop in the seat gap [5,7]. Furthermore, flow separation is caused. A comparison of the two hydraulic force components F_p and F_m in Figure 4 shows that the change in the hydraulic force F_h is mainly determined by the change in the pressure force F_p for $Re < Re_{cr}$. At high Reynolds numbers $Re > Re_{cr}$, momentum forces dominate over viscous forces in the flow. Thus, it is reasonable to set the effective pressurized area A_p to a constant value there, which is smaller than for a laminar flow, and let the change in the momentum force F_m determine the change in the hydraulic force F_h alone.

Further Component Models

The pump element is modelled as a hydraulic volume with a sinuous variation of its size. For the other hydraulic components of the circuit (Figure 2), standard models of the commercial simulation package Bath*fp* [9, 10], used in this investigation, are employed. For the back flow line, the Bath*fp* finite volume distributed model with each 4 internal pressure and flow rate nodes is used to allow for pressure wave dynamics. Cavitation and air release are taken into account by a two-phase homogeneous equilibrium model.

Simulation Results and Model Verification

The operation of the hydraulic system (Figure 2) results in pressure transients, displayed in Figure 5. The comparison of the transients, determined experimentally (left) and by simulation (right), shows that all relevant hydraulic effects are accounted for by the simulation model. The pronounced pulsation of the system pressure p_B is caused by the volume pulses, produced by the reciprocating pump. The pressure control valve operates at Reynolds numbers mostly below $Re_{cr} = 500$, and the described effects of laminar-turbulent flow transition, with the further parameters $\alpha_t = 0.7$ and $f_p = 0.8$, on hydraulic force and flow rate mainly determine the large amplitude and transient shape of the system pressure p_B. These effects act on the closing body like an additional, strong stiffness. The level of the system pressure p_B is mainly determined by the spring preload F_0 and slightly influenced by the transitional flow effects.

Superimposed on the system pressure pulsation are bursts of pressure oscillations with a high frequency. These bursts are correlated with spikes in the pressure transient p_c, downstream of the control valve, which are more pronounced at every second pump cycle. A further analysis of the system transients reveals that the origin of these peculiar pressure transients is the periodic creation and breakdown of cavitation in the extended back flow line, causing fluid column separations and reformations [6].

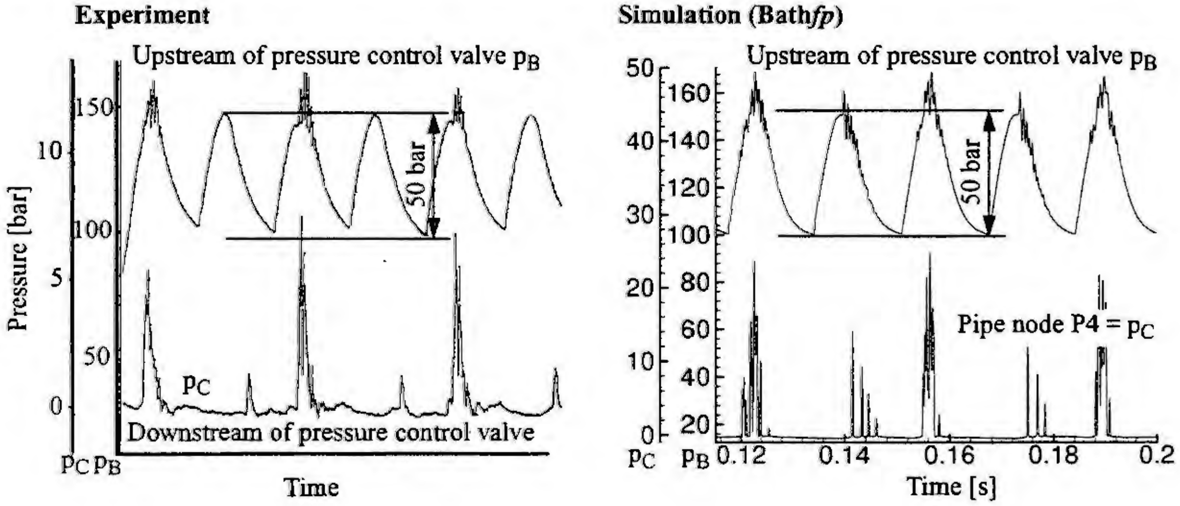


Figure 5: Pressure transients

Conclusions

The proposed simulation model is an attempt to map the strong influence of transitional flow phenomena, observed in valves with chamfered seats, on simple algebraic expressions, which are consistent for the flow rate and the hydraulic force, utilizing the similarity principle of fluid mechanics. A set of three adjusted parameters, the turbulent flow coefficient α_t, the critical Reynolds number Re_{cr}, and the newly introduced turbulent effective pressurized area factor f_p, suffices to describe the static characteristic of the investigated poppet valve design with a spherical closing body over a wide range of operating conditions. Experimentally verified simulation results with this model also demonstrate its dynamic suitability.

Comprehensive experiments remain to be carried out to check the general model validity, as well as to supply guidelines for setting the parameters for different valve designs and geometries.

References

1 Blackburn, J. F., Reethof, G., Shearer, J. L., Fluid Power Control. Wiley, New York, 1960.

2 Ellmann, A., Piché, R., A Modified Orifice Flow Formula for Numerical Simulation of Fluid Power Systems. In: Proc. ASME International Mechanical Engineering Congress and Exposition, Fluid Power Systems and Technology Division, Atlanta, 1996.

3 Johnston, D. N., Edge, K. A., Vaughan, N. D., Experimental Investigation of Flow and Force Characteristics of Hydraulic Poppet and Disc Valves. Proc. Instn. Mech. Engrs. Vol. 205, 1991, 161 - 171.

4 Lu, Y.-H., Entwicklung vorgesteuerter Proportionalventile mit 2-Wege-Einbauventil als Stellglied und mit geräteinterner Rückführung. Dissertation RWTH Aachen, 1981.

5 McCloy, D., Martin, H. R., Control of Fluid Power: Analysis and Design. Wiley, New York, 1980.

6 Mittwollen, N., Hydraulic Simulation of Cavitation Induced Fluctuations with Peculiar Periodicities in a Fluid Power Unit. In: Proc. 8th Bath International Fluid Power Workshop, Design and Performance, Bath, 1995 (Eds.: Burrows, C. R. and Edge, K. A.), Wiley, New York, 1995, 156 - 168.

7 Oshima, S., Ichikawa, T., Cavitation Phenomena and Performance of Oil Hydraulic Poppet Valve (3rd Report, Influence of the Poppet Angle and Oil Temperature on the Flow Performance). Bull. JSME, Vol. 29, No. 249, March 1986, 743 - 750.

8 Post, K.-H., Untersuchungen an elektrohydraulischen Schaltventilen mit fluidischen Kugelelementen. DFVLR Research Report 73-35, Porz-Wahn, 1973.

9 Richards, C. W., Tilley, D. G., Tomlinson, S. P., and Burrows, C. R., A Second Generation Simulation Package for Fluid Power Systems. In: Proc. 9th Internat. Symp. on Fluid Power, 1990, Cambridge, STI, Oxford, 1990, 315 - 322.

10 Richards, C. W., Bath*fp* Manual Vol. 2, Model Reference Guide. Fluid Power Centre, Univ. of Bath, 1993.

11 Streeter, V.L., Wylie, E.B., Fluid Mechanics. McGraw-Hill, 1981.

INTEGRATED PRODUCT MODELING - A CONTRIBUTION TO COMPUTER AIDED PRODUCT DEVELOPMENT

J. Gausemeier, D. Brexel, M. Flath, F. Kallmeyer, M. Miksic
Heinz Nixdorf Institut, Universität-GH Paderborn
Fürstenallee 11, D-33102 Paderborn, Germany

Abstract. In many areas the importance of information technology is rising. Particulary new products consist of the integrative use of mechanics, controllers, electronics, actuators, sensors, integrated circuits and software. To handle the challenge of designing such products, the authors present an approach for the integrated development of complex industrial products. This approach is named Integrated Product Development (IPD). It is founded on a generic model of activity that will be specificied according to a special development task. Therefore a product model is emerged from coherent partial models, and serves as basis for analysis. The IPD is divided in two areas, the conceptual product design and the domain-specific elaboration. Thereby the integration of existing domain specific development methodologies into an overall approach for the development of complex industrial products can succeed.

Introduction

The product development process[1] of many industrial enterprises is increasingly facing new requirements. The market switches from the view of selling to the view of satisfying customers, the duration of the product development must be shorten in order to compete within the international market, and the complexity of the products increases [1]. These requirements can only be fulfilled by a systematic procedure (development methodology[2]) over the whole product development process. In today´s industrial practice there does not exist an overall development methodology for complex industrial products.

The rising capacities of information technology lead to new opportunities for complex industrial products. These products are characterized by an integrative use of mechanics, controllers, electronics, actuators, sensors, integrated circuits and software. Thereby in traditionally fully mechanical products enhanced functions or lower manufacturing costs are possible (e.g. robots or the hexapod machine tool). On the other hand completely new products with new use become feasible like copiers or facsimile machines in the past. For this purpose an integration of the concerned development areas is necessary.

However, there is a basic problem how to proceed within such developments. Special models of action exist for every domain involved. Basically these models contain domain specific crucial points within their approaches [4]. Examples for models of action can be found in the literature for the domain mechanics [3][5][6][7]; for the domain electronics/integrated circuits [8][9]; for the domain automatic control [10][11][12], and for the domain software [13][14][15]. Through their domain specific approach all these models of action are not sufficient for the development of complex industrial products [4]. Especially there is a lack for the early integration of economic aspects in these approaches [1].

Based on these considerations the Integrated Product Development is worked out at the Heinz Nixdorf Institut within the department of Computer Integrated Manufacturing. This is characterized by three essential features: The intensification of the early design stages[3] of the product development process, the support of a system composition (the access to known solution approaches and modules), and the incorporation of computer support into the development process.

Integrated Product Development

The Integrated Product Development is based on existing design methodologies for the purpose of their integration

[1] The product development process is a line of tasks ranging from the product idea to the successful introduction on the market. Thus, it includes product planning/product marketing, development/design, work scheduling and building the means of production.

[2] A development methodology consists of concrete instructions for the development of technical systems [2][3]. With the help of development methodologies, development tasks can be solved. They can be used in every stage of the development process in order to generate (synthesis), judge (analysis), represent (modeling), show (presentation) and develop results.

[3] The early stages of product development consist of product planning and conceptual product design. They result in a principle solution.

and extension. The basic concept is a generic model of activities. Within this model all activities can be employed during the development of complex industrial products. Since the product development depends both on the specific character of a line of business and the nature of the development (novel, adaptation, or variants design) a specific model of activity is derived from the generic model. This contains the activities and the relations between them for the specific development process. The aim is to avoid major iteration loops within the design process. The major loops should be avoided by many small loops.

During the application partial models[1] are generated by the particular activities. Elements of the partial models are linked to each other by means of relations. Therefore a product model[2] emerges from a number of coherent partial models.

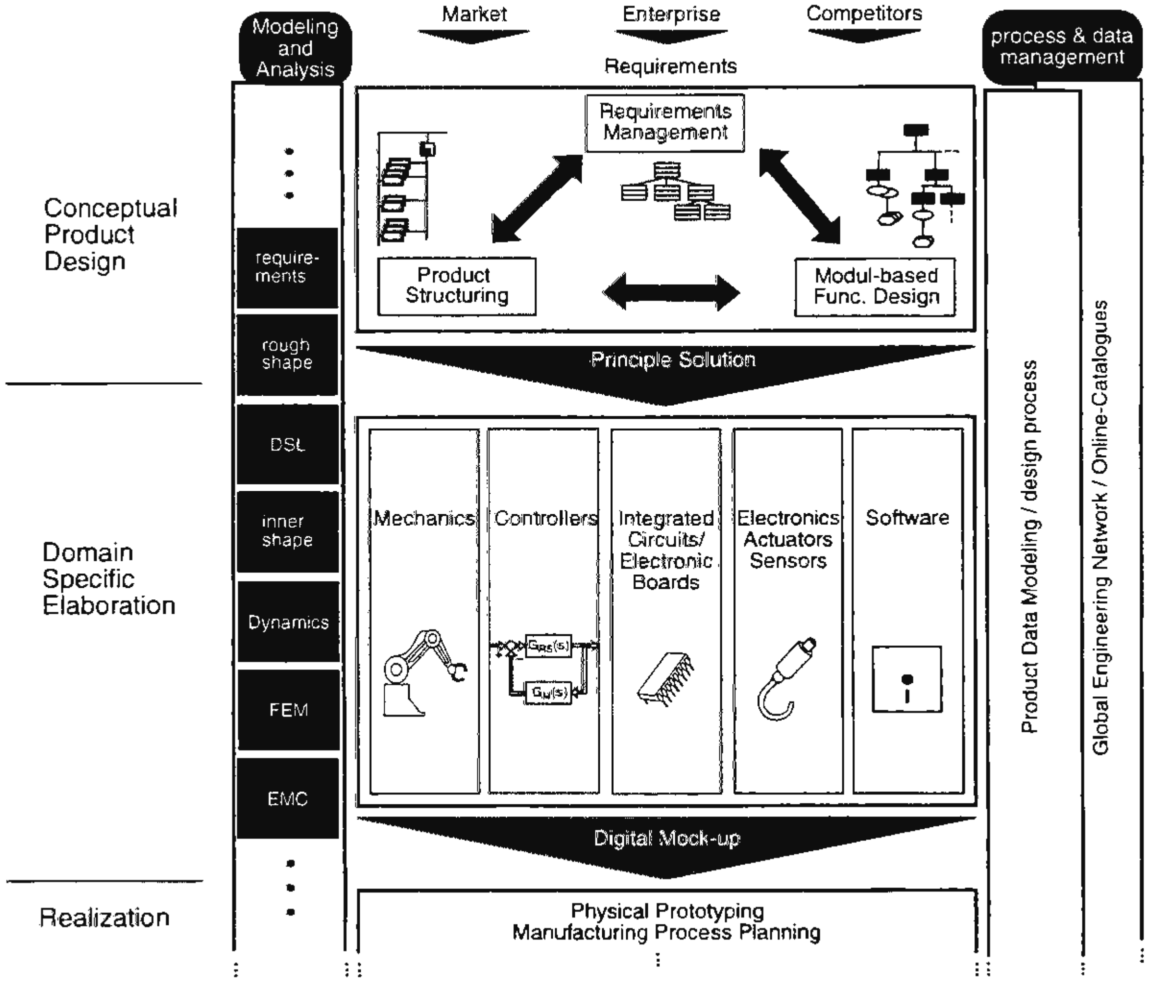


Figure 1: Integrated Product Development
(FEM Finite Elements Method, EMC Electromagnetic Compatibility, DSL Dynamic System Language)

The Integrated Product Development is divided in two major parts: the conceptual product design and the domain specific elaboration (Figure 1). Thereby the conceptual product design holds a special meaning. Considering the external effects as the market, the enterprise and the line of business major decisions are made. The aim is the principle solution which is decisive for the future success of the product [16]. The principle solution does not only determine the function of the product but also the essential manufacturing costs. The conducting activities during the conceptual product design are represented by the requirement management, the product structuring and the module-based functional design.

Starting from the principle solution single partial models are extended in the domain specific elaboration. Accompanying analyses of particular functions are carried out (FEM-analysis, EMC-analysis, etc.). The particular do-

[1] A partial model is a part of a model. Coherent partial models of a complete model do not have overlapping contents.

[2] A product model represents all the data of a product, covering the whole life cycle.

mains are not indepedent from each other, moreover there exist manifold relations. For example, the design of a controller needs parameters from the mechanical design (values of mass, moments of inertia, spring constants, etc.) from the circuit design (values of time delays), and from the electronic design (values of resistors, capacities, inductances, etc.). The result of the domain specific elaborations is a digital mock-up[1]. Therefore the conventional cost- and timeconsuming preparation of a variety of physical prototypes is avoided. All functions will be tested on the internal computer model. Modifications can be done quickly and without large effort.

Furthermore the Integrated Product Development supports explicitly the work with completed modules. Therby the system composition is taken into account. In order to quickly achieve results without extensive costs, the implementation of existing components is indispensable to fulfill the required functions. The components can be searched and implemented systematically using the Internet (Global Engineering Network / Online-Catalogues, [17]). Further the specified generic model and the product model can be implemented within Product Management Systems (e.g. METAPHASE^{TM}).

Requirements Manangement

The purpose of the requirements management is to formally express requirements which are often communicated in every-day language, and to prepare them so that they can be used as concrete guidelines in a product development process. Also a permanent control of the fulfillment of requirements has to be done. Requirements management is based on two models. These are: 1. the requirements model which is easy to handle for a computer, and 2. the model of activities to set up the requirements model.

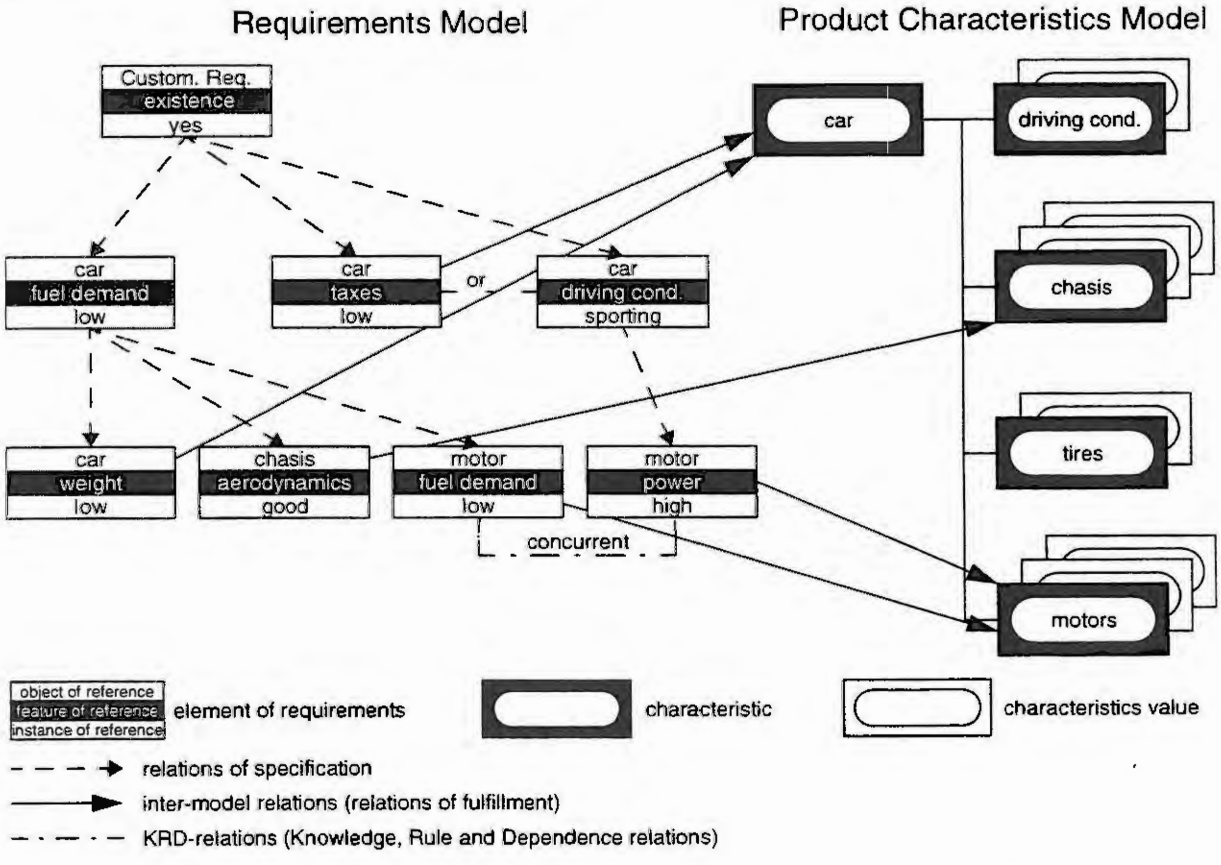


Figure 2: Example of a requirements model with relations to the product characteristics model

The formalized requirements model is a structure of the elements of requirements which are connected by means of model-internal relations. The content of an element of requirements is determined by a tripel [object of re-

[1] A digital mock-up is a complete, elaborated, and workable computer model of an industrial product which fulfills all requirements.

ference, feature of reference, instance of reference]. That means an element of requirements describes a single instance of a single feature of an object of reference [4] (Figure 2). Furthermore, the model-internal relations are divided into relations of specification and KRD-relations.

In the course of the product development new expericences and informations specify some additional requirements. Those new and more specified requirements are connected with the former ones using relations of specification. The relations of specification define a hierarchical structure of elements of requirements and they represent a base for the backtracking of the requirements.

The KRD-relations describe additional knowledge, rules, and dependences between the elements of requirements. This additional information about requirements is also specified during the course of the design process. However, this is not implemented visibly within the requirements model. The KRD-relations play an important role within the analysis of consistency[1] of the requirements model.

Apart from the model-internal relations there are also inter-model relations. They describe connections between requirement elements and the results of the product development. Those results are defined in terms of characteristics whithin the product characteristics model. There are two types of inter-model relations: relations of fulfillment and relations of motivation.

A relation of fulfillment assigns one requirement to one particular element of the partial product model. Therefore a requirement can be used as a guidance for further development. A relation of motivation describes the reason for the emergence of an additional requirement. Reasons arise from the development process and serve for statistical purposes only.

The model of activities for the construction and for the manipulation of the requirements model contains the following five stages: 1. Administration, 2. Initialization, 3. Connection and specification, 4. Assessment, and 5. Analysis.

Modul-based Functional Design

The purpose of the modul-based functional design is to develop the principle solution for a technical system by performing the following activities: building up a hierarchy of functions, constructing a functional flow structure, and conducting the proof of workability.

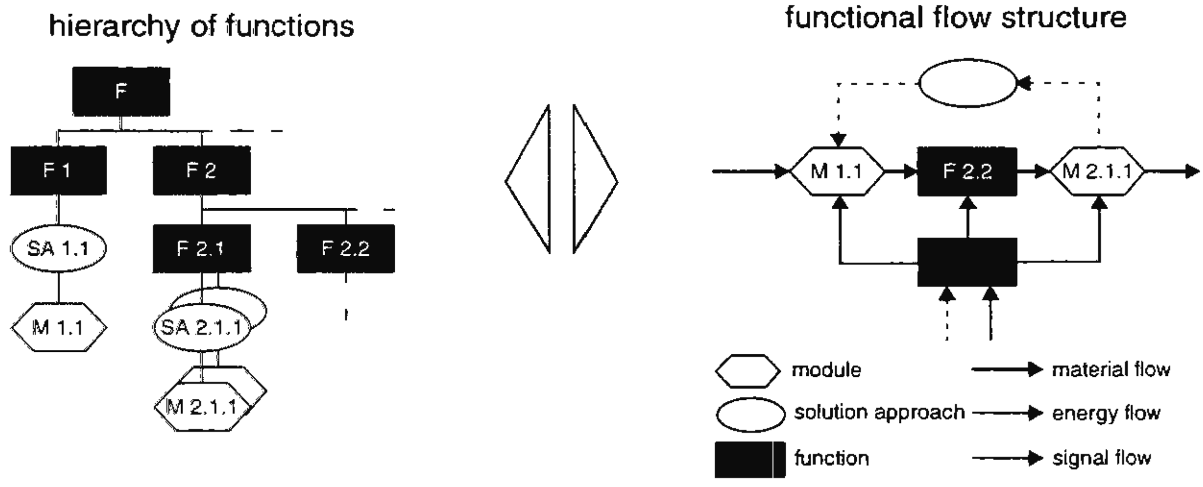


Figure 3: Hierarchy of functions with corresponding functional flow structure

Proceeding from requirements the product is divided into its technical partial functions (Figure 3). As early as possible solution approaches should be proposed to fulfill the partial function. In the next step existing modules based on the specific solution approach should be identified and introduced. A function will be subdivided into more concrete subfunctions if no appropriate module can be found. The result of this process is a hierarchical function structure with different levels of abstraction. Existing modules are found at the leaves of this functional tree.

Parallel to the construction of the hierarchy of functions the functional flow structure must be built up. This

[1] The primary aim of the consistency analysis is to determine whether some specific requierements model contains contradictions or not.

structure contains partial functions and functional connections between those functions. The connections are described by means of energy, material, and signal flows [3]. If an abstract partial function is divided into more concrete partial functions, this one is replaced by the network of more concrete partial functions. On the highest level of concretization a structure of modules is found which depicts the functional flow structure of the principle solution.

For the proof of workability the behaviour of the single partial functions and modules must be formally described. Thereby functional tests can be carried out within a computer. If existing modules are assigned to some partial functions, they are described by their real behaviour. Otherwise, the partial functions are described by their desired behaviour. Due to the theory of automats in computer science the behaviour description is set up by means of states and rules. For every partial function a set of possible states and flow transformations within those states will be defined depending on the input flows. The dynamic of the changing states is described through rules by means of events and actions. For chosen excitations, reactions gained on the computer model are observed. Based on those reactions, the desired statements about the behaviour of the principle solution can be made.

Product structuring

Product structuring serves the mastering of the complexity within business processes and within the marketing of the products. The foundation is a semantic product structure model. This is characterized by three different partial models „characteristics structure", „functional structure", „assembly structure", their relations, and different views onto the product structure model (Figure 4).

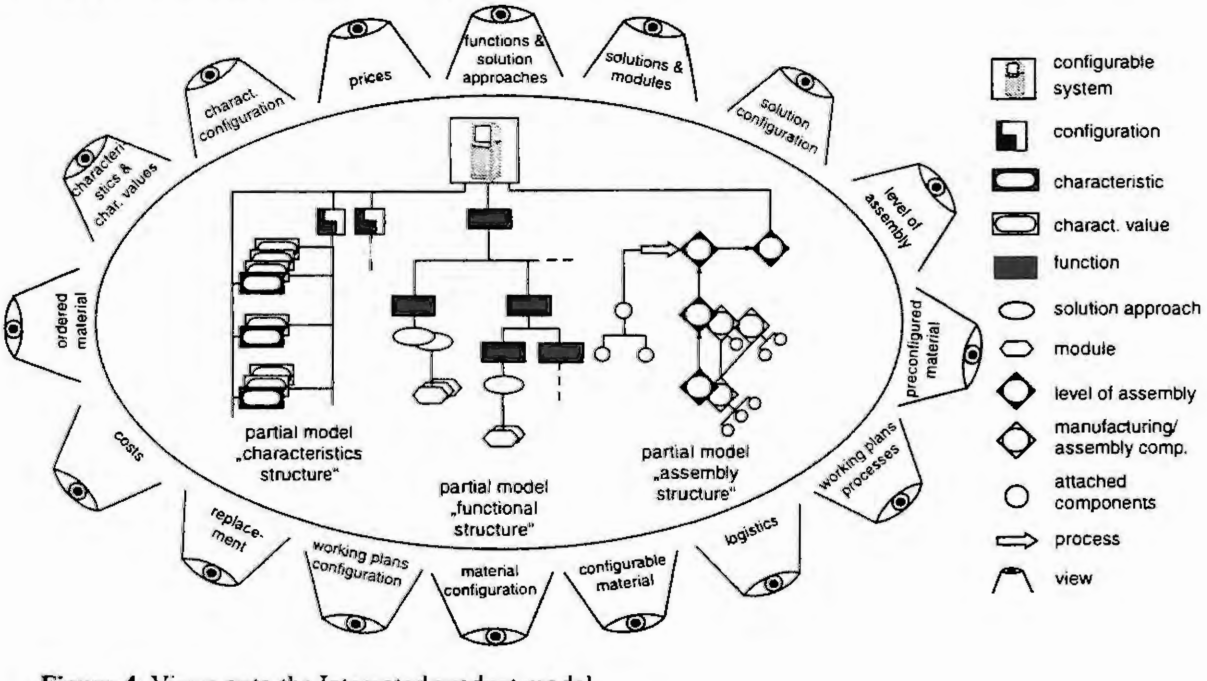


Figure 4: Views onto the Integrated product model

The partial models of the product structure are built up by means of semantic objects and relations between them. A physical component of a product can be represented through different semantic objects within partial models. The partial model „characteristics structure" contains elements for tender, configuration and order activities for a product. The purpose of the characteristics structure is the optimization of the interface to the market, that means to generate the topmost transparency due to prices and possibilities of the configuration. The partial model contains following objects: characteristics, characteristics values and configuration. The characteristics describe neutrally from variants the possible instances of the product which are meaningful for the customer specific configuration of the system. Characteristics values describe the instances of characteristics, and configurations the different kinds of configuration based on the characteristics and the characteristics values of a system.

The partial model „functional structure" is a functional model of system components and connections between them. It contains the integrated and functional oriented structure of all mechanical, electronical, actor, sensor, and software components. It represents the interface to the activity modul-based functional design which was described in the former paragraph.

The partial model „assembly structure" is the integrated structure of the parts list and the working plan of the system. It contains following objects: level of assembly, manufacturing/assembly component, attached component and process. Level of assembly describes neutrally intermediate states of variants during the assembly of the product. Manufacturing/assembly components are concrete instances of levels of assembly and describe preconfigurated, stored resp. ordered components due to a customers commission. Attached components are concrete components which either flow in the preconfigurated manufacturing/assembly components or flow in stored resp. ordered or outside obtained components for the customer specific assembly of systems. Processes are tasks for the assembly of the customer specific configuration of the system.

The integration of the partial models towards the semantic product structure model is done by three levels: 1. Integration characteristics structure - functional structure, 2. Integration functional structure - assembly structure and 3. Integration characteristics structure - assembly structure. Thereby existing relations are depicted through relational objects.

In order to preserve the transparency and consistency of the integrated product and the configuration structure different views are necessary which show a part of the whole model. They differ due to the contained product structure objects and relations and the allowed manipulations for the particular user.

Outlook

The aim of the further work will be the verification of the presented approach within the frame of industrial projects. Furthermore within the domain software design it is planned to achieve a stronger methodical linkage to the special requirements of complex industrial products. Moreover the representation of the principle solution due to the domain specific elaboration must be formalized stronger.

References

1. Gausemeier, J., Brexel, D., Frank, T., Humpert, A., Integrated Product Development - A New Approach for Computer Aided Development in the Early Design Stages. In: Proc. Third Conference on Mechatronics and Robotics, Paderborn, 1995, B.G. Teubner Stuttgart 1995, 10 - 29.
2. Beitz, W., Konstruktionsmethodik in der Praxis. Springer-Verlag, Konstruktion 41, 1989, Berlin, 403 - 405
3. Pahl, G. and Beitz, W., Konstruktionslehre - Methoden und Anwendungen. Springer-Verlag, 3. Edition, Berlin, 1993.
4. Humpert, A., Methodische Anforderungsverarbeitung auf Basis eines objektorientierten Anforderungsmodells. HNI-Verlagsschriftenreihe, 1995.
5. Koller, R., Konstruktionsmethoden für den Maschinen-, Geräte- und Apparatebau. 1. Edition, Springer-Verlag, Berlin, 1976.
6. Rodenacker, W.G., Methodisches Konstruieren. Konstruktionsbücher. Bd. 27, Springer-Verlag, Berlin, 1976
7. Roth, K., Konstruieren mit Konstruktionskatalogen. Band 1, Konstruktionslehre, Springer-Verlag, Berlin, 1994
8. Reinert, D., Entwurf und Diagnose komplexer digitaler Systeme. VEB-Verlag Technik, Berlin, 1983.
9. McDermatt, R.M.: Computer Aided Logic Design. USA, Indianapolis, IN, 1985.
10. Schulz, G., Grundlagen, Analyse und Entwurf von Regelkreisen, rechnergestützte Methoden. Springer-Verlag, Berlin, 1995.
11. Dörrscheidt, F. and Latzel, W., Grundlagen der Regelungstechnik. Teubner-Verlag, Stuttgart, 1993.
12. Föllinger, O., Regelungstechnik - Einführung in die Methoden und ihre Anwendung. Hüthig, 1994.
13. Agresti, W.W. (Hrsg.), New Paradigms for Software Development. 1. Edition, Publ. by DC-Press, Amsterdam 1986.
14. Booch, G., Object oriented analysis and design with applications. 2. ed. Redwood City, Calif., Benjamin/ Cummings Publ. Co., 1994.
15. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., Object-Oriented Modeling and Design. Prentice Hall, USA, Englewood Cliffs, NJ, 1991.
16. Gausemeier, J., Brexel D., Humpert, A., Anforderungsbearbeitung in integrierten Ingenieursystemen. Springer-Verlag, Konstruktion 48, 1996, 119 - 127.
17. Gerber, H., Sabin, A., Basic GEN Architecture and Services, Conference of Integration in Manufacturing (IiM), 13-15 September, Vienna, Austria, 1995.

ON THE OPTIMAL DESIGN OF (ENGINE MOUNTS AND) VIBRATION ABSORBERS IN CABRIOLETS BASED ON AN EXPERIMENTALLY IDENTIFIED CAR-BODY MODEL

K. Reinke, J. Wallaschek
Universität-GH Paderborn
Heinz Nixdorf Institut, Mechatronik und Dynamik
Fürstenallee11, 33102 Paderborn, Germany

M. Heidrich
Freudenberg Technische Dienste, FFD-CAE
69465 Weinheim, Germany

A. Paul
Freudenberg Dichtungs- und Schwingungstechnik KG, TEZ-VA
69465 Weinheim, Germany

Abstract. Vibration absorbers are used in cabriolets to improve ride quality and comfort. This is necessary, because most cabriolet chassis have low resonance frequencies starting at about 15 Hz. The task of determining the optimal location of the vibration absorbers and of determining the optimal mechanical parameters can not be solved by trial and error. Instead numerical simulation and parameter optimization at the early design stage are required. In this context reliable and experimentally validated vibration models of the car-body and of the absorbers are needed.

In the present paper we describe the process of obtaining a vibration model of the car-body of a cabriolet, starting with a multi-degree of freedom representation of the chassis dynamics. Using the Rayleigh-Ritz method a model reduction is carried out, resulting in a model which represents only those vibration modes which are relevant for the absorber design.

Since the dynamics of tires, shock-absorbers, engine body and engine mounts also have a marked influence on the overall system dynamics, their dynamic behavior must also be taken into account. The parameters of the overall vibration model were obtained by dynamic testing using a 4-channel-servohydraulic test-rig in a step-by-step identification procedure.

Introduction

The design of the car-body of a new automobile is a difficult task since many criteria including e.g. aesthetical appearance, acoustic behavior, crash safety, manufacturing technology, reparability, recycling and many more must be taken into account.

The design of the car-body of cabriolet cars is a very demanding task. It is solved quite well by the automobile producer [1]. Nevertheless the stiffness of the cabrio is lower than that of the corresponding limousine. This however leads to a shift of the first torsion mode resonance frequency from 25 Hz to 15 Hz or even lower frequencies [2]. As a consequence, the torsion mode becomes very sensible with respect to vibration excitation, since its resonance frequency now lies in the same frequency range as the resonance frequencies of the rigid body modes of the engine or the axles. In extreme cases the vibrational behavior of the car-body becomes so poor that ride comfort criteria are violated.

One possible solution to increase ride comfort is to improve on the car-body design with the aim of stiffening the structure and shifting the critical torsion mode resonance to higher frequencies. This however is a difficult, costly and time consuming process. As an alternative, vibration absorbers tuned to the critical vibration might also be used. Another possibility is to design the engine mount system in such a way that the engine itself acts as the vibration absorber. It is obvious that this second design alternative is not trivial and that it requires excellent knowledge of the vibrational behavior of the overall system.

In the process of designing an appropriate vibration absorber and/or engine mount system for a cabriolet, an adequate model of the vehicle dynamics is required [3]. The design quality is to a large extend influenced by the models used in the design process [4]. Reliable results can only be expected if correct models are used and if the parameters of these models are known with sufficient accuracy. These parameters do strongly depend on the specific design object at hand and it is often extremely difficult to estimate them a-priori. Therefore suitable parameter identification methods are needed to obtain reliable models.

In this paper a model for the torsional vibration of the car-body and for the relevant rigid body modes of the engine, suspension and wheels is derived. A parameter identification method which has been adopted for this model is introduced and it is shown that a sufficient description of the vibrational behavior of the overall system can be achieved. The model is validated using a four-piston servo hydraulic testbed.

The mechanical model

Since we are mainly interested in designing a vibration absorber and/or an engine mount system tuned to the critical torsion resonance mode of the car-body, only the relevant degrees of freedom of the car-body need to be modeled.

The torsional vibrations of the car-body are described by the lumped mass model of Fig.1. It consists of rigid bodies connected by torsion springs and dampers. Thus the car-body is essentially modeled as a torsion bar.

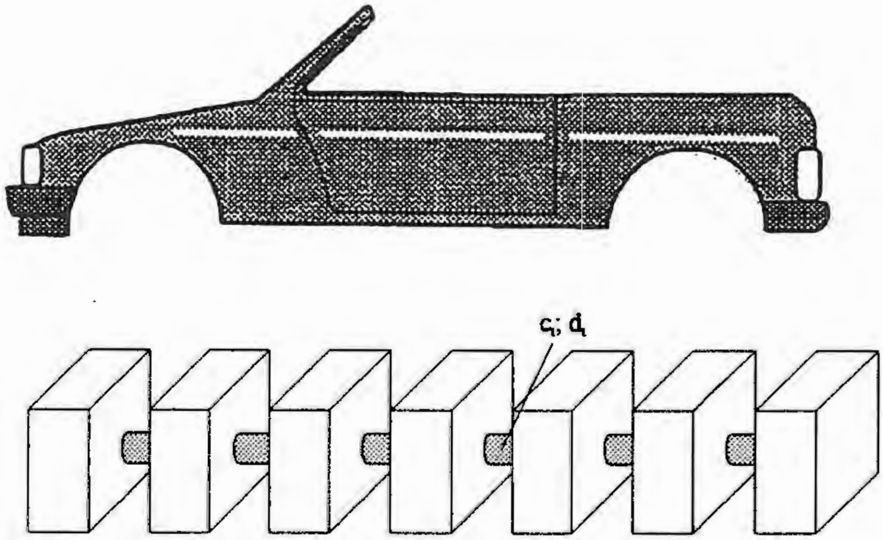


Fig.1 : Lumped-mass model for the torsional vibrations of the car-body.

Even if only a small number of degrees of freedom is taken into account a good representation of the first torsion mode can be obtained. Since vibration absorbers for cabriolets typically are placed in the very front or end section of the car-body, this part of the structure should be discretized with a sufficient number of elements in order to have a good local representation at the points where the vibration absorbers are connected to the car-body. If the number of elements in Fig.1 is sufficiently high, the effect of the coupling of engine, suspension and wheels can be taken into account in a straightforward manner by interconnecting them via visco-elastic mounting elements to the lumped mass model of the car-body.

With respect to the process of parameter identification the number of independent model parameters should be small. In static tests it was verified that the stiffness of the car-body does not vary significantly along the length of the car. It was therefore decided to assign the same stiffness parameter to all torsion springs. Only little is known about structural damping. It can, however, be expected that it is also uniform so that the same damping coefficient may be assigned to all damper elements.

If all mass elements in Fig.1 had the same inertia, the node of the first (free-free) torsion mode would be located exactly in the midpoint of the length. In experiments it was observed that this is not the case. Therefore the hypothesis was postulated that the rotary inertia of all elements is equal except for some elements at the front of the car-body which have higher rotary inertia due to some additional aggregates located at the front end of the vehicle. Thus the inertia of the system is described by three parameters: the rotary inertia of the elements at the end of the vehicle, the rotary inertia of the elements at the front of the vehicle and the relative number of front elements as compared to the number of rear elements.

The representation of the motion of suspension and wheels is straightforward and in summary the mechanical model of Fig.2 is obtained for the description of the torsion dynamics of a cabriolet. This model can easily be extended so that the dynamics of one or more additional vibration absorbers and of the engine can be taken into account, Fig.3. This model is used for the optimization of engine mount systems or vibration absorbers.

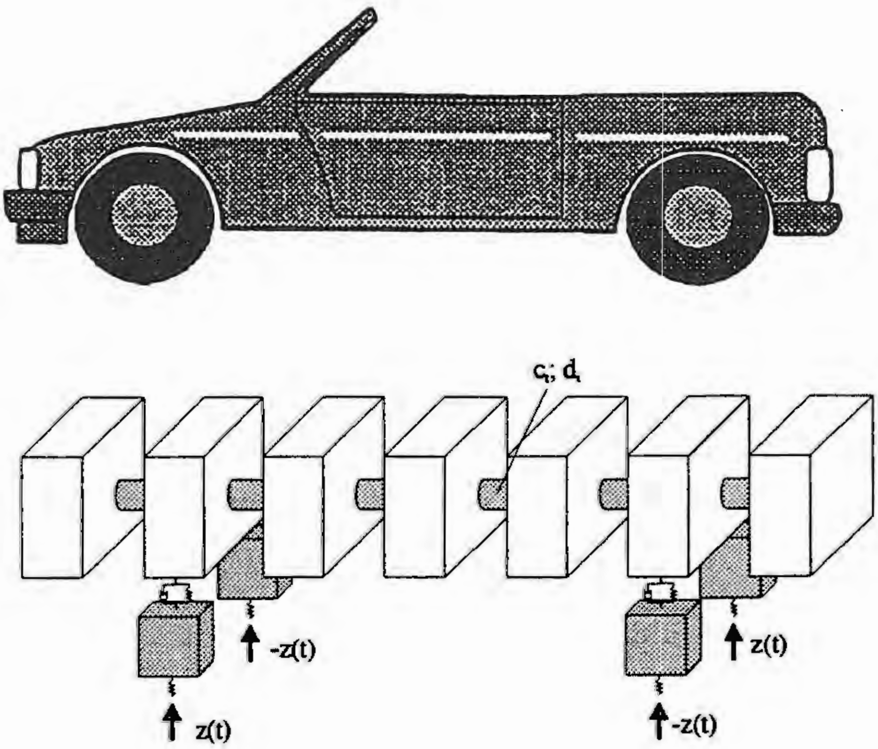


Fig.2 : Mechanical model for the torsion dynamics of a cabriolet.

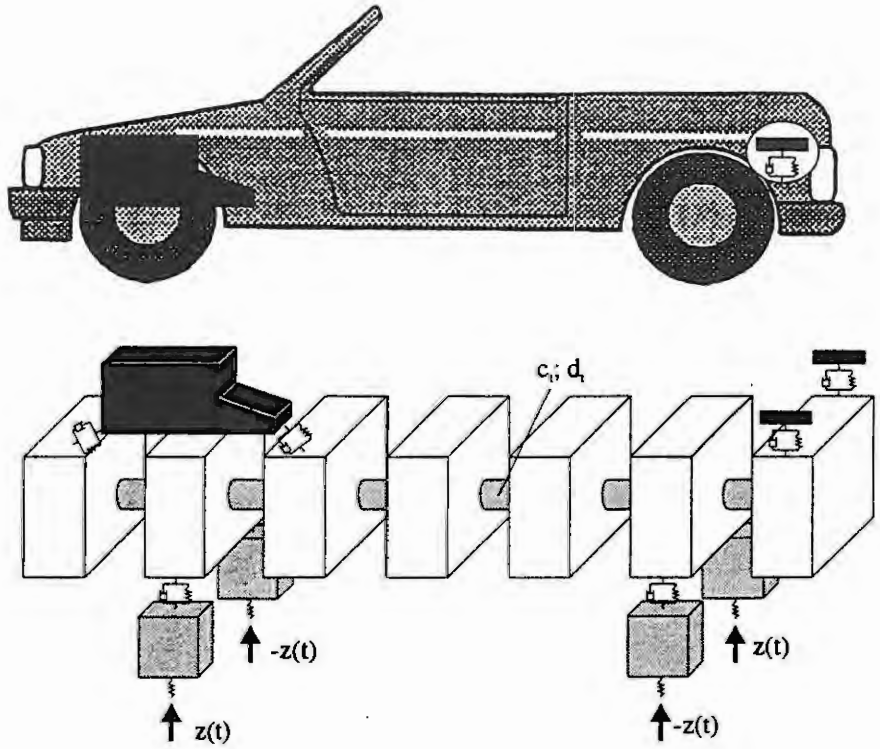


Fig. 3 : Model for the torsion dynamics of a cabriolet including engine and vibration absorbers.

Parameter identification

In the present paper the emphasis is put on the parameter identification of the dynamical model for the torsional vibration of the car-body and the motion of wheels and suspension elements. Therefore the engine mounts which usually have a well defined stiffness and damping have been replaced by rigid mounts during the tests. The characteristics of the engine mounts are well-known from component tests and need not be identified in the overall system.

The geometry parameters of the car-body are either simple to measure or can be obtained from the vehicle documentation. However it is necessary to determine a lot of system parameters which cannot be obtained simultaneously in one single experiment. System parameters like total weight, location of the barycenter as well as the structural and wheel stiffness in all three space directions can be determined by static experiments without significant difficulties. Dynamic experiments are necessary to obtain parameters like damping, equivalent mass of the wheels and suspension as well as moments of inertia for the elements describing the torsional motion of the car-body. These experiments were performed on a servohydraulic four-stamp shaker at Freudenberg Dichtungs- und Schwigungstechnik KG. The large number of parameters however requires a good test strategy and well defined experiments, which have to be performed in a sensible evaluating sequence using the appropriate methodology.

As a matter of fact it was observed that manual inspection of the mode shapes of the overall vehicle model provided good support for the choice of a sensible sequence of experiments. As a result of the inspection process, which initially has to be performed based on typical (empirically guessed) model parameters, it is possible to decouple the complex overall system into mechanically almost decoupled coherent partial models. For example due to symmetry the in-phase stroke motion of the wheels is decoupled from all other modes of the system. It was therefore possible to determine the wheel and suspension parameters by an in-phase excitation of the wheel-stroke mode. The stiffnesses have been determined statically, the equivalent mass and damping parameters were obtained from the frequency response function of the wheel, which was measured under stepped sinus excitation. After this test the model parameters of wheel and suspension are known and can be used during the rest of the identification process.

In a next step the parameters of the torsional mode are determined as follows: the hydraulic stamps at front-left and back-right are excited in-phase, while the stamps at front-right and back-left are excited in antiphase to the other two. This leads to a strong excitation of the torsion mode. During this test, acceleration sensors mounted along the length of the car-body are used to register the angular rotation. The angular rotation is calculated from the acceleration signals under the hypothesis that the central principle axis of rotation is located in the geometric symmetry plane of the car and that it passes through the mass center of the relative segment. If excitation frequency is close to the torsional resonance of the car-body, the mode-shape of the torsional motion and in particular the location of the nodal point can be determined.

Since the location of the nodal point is strongly influenced by the distribution of rotary inertia, while damping and stiffness can be assumed to be homogenous, the distribution of rotary inertia between front and end segments of the torsional vibration model is chosen in such a way that location of the nodal point is well approximated by the theoretical model. Given this inertia distribution and a frequency response function of one or more sensors, the damping and stiffness parameter of the torsional motion are chosen such that the resonance frequency and the magnitude of the frequency response function at resonance are reproduced by the model.

Some experimental results

The tests were performed on a four-stamp servohydraulic testbed and a LMS-CADA-X system was used in the experiments [5]. Fig.4 shows a photograph of an acceleration sensor mounted at one of the mudguards of the cabriolet. It should be clear that there exist numerous side-effects influencing the experimental results which have been neglected in the vibration model. It is therefore interesting to note that regardless of these idealizations a fairly good correspondence between measured and theoretically predicted results could be observed.

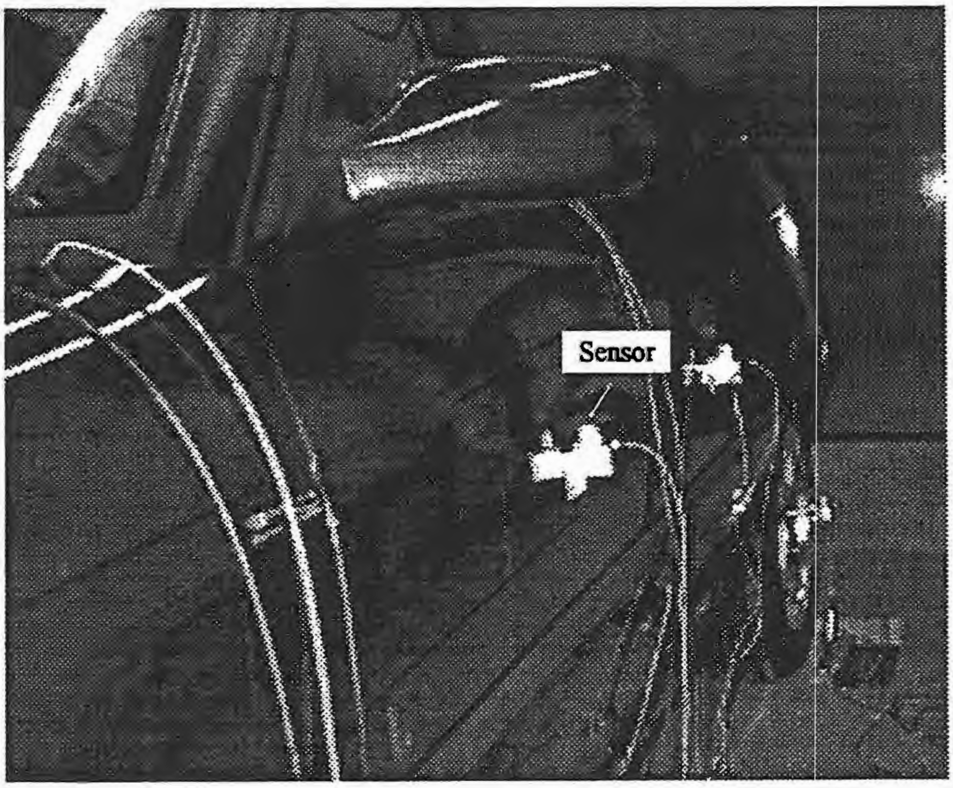


Fig. 4 : Acceleration sensors mounted on the car-body

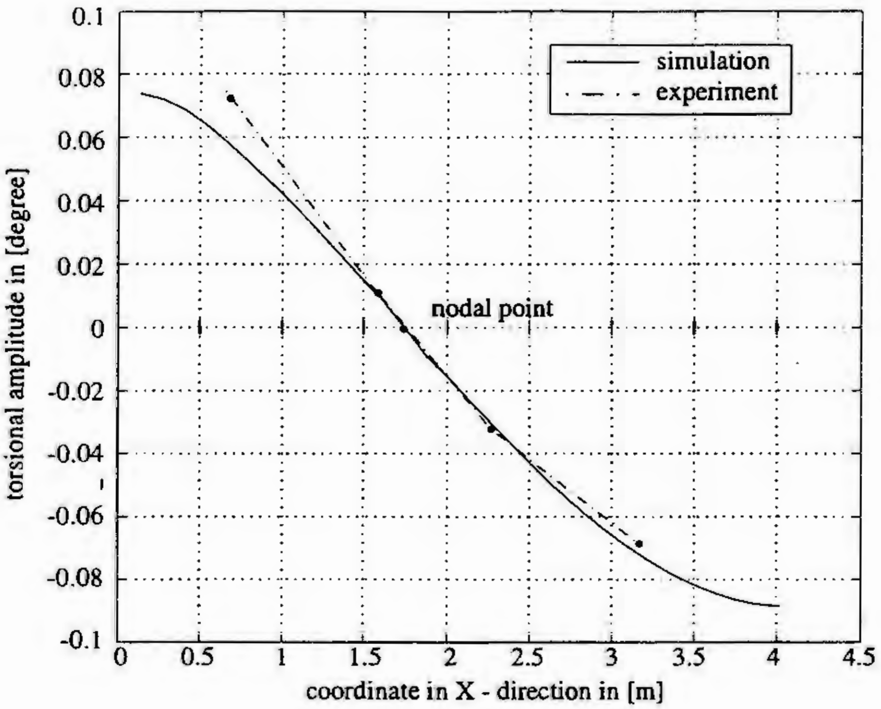


Fig. 5 : Comparison between simulation and experiment : torsion mode of the car-body

Fig.5 shows a comparison between the calculated and experimentally determined torsion mode of the car-body. Note that the location of the nodal point agrees perfectly since this has been one criterion used in the process of parameter identification.

Summary and outlook

A simulation model of the car-body of a cabriolet car has been presented. This model has been integrated into problem-oriented overall vehicle model which is useful for the design of vibration absorbers and engine mount systems. Six degrees of freedom (rigid body modes of the engine) of the model have been frozen during the experiment by using rigid engine mounts, so that the parameter identification on a servohydraulic four-stamp shaker can be performed. A systematic method for the identification of the model parameters has been developed and was used for the torsion model. A comparison between simulation and experiment showed, that the developed method leads to a sufficiently accurate parameter estimation. Now all the parameters for the overall vehicle simulation are available, so that a model-based development of vibration absorbers and/or engine mount system is possible.

References

[1] Müller, M. ; Beyer, R. ; Grunau, R. ; Paul, A. : Auslegung von Karosserietilgern. Tagung , Haus der Technik, Essen, September 1996.

[2] Fischle, R. : Aggregatstarrkörpermoden contra Karroserieschwingungen. Tagung , Haus der Technik, Essen, März 1993.

[3] Müller, M. ; Siebler, T.W. ; Gärtner, H. : Simulation of vibrating vehicle structures as part of the design-process of engine mount systems and vibration absorbers. SAE-paper 952211E, 1995.

[4] Wallaschek, J. : Modellierung und Simulation als Beitrag zur Verkürzung der Entwicklungszeiten mechatronischer Produkte. VDI-Bericht, Nr. 1215, 1995.

[5] LMS : CADA-X, User manual, TWR, Rev. 3.3, 1995.

MODELLING AND SIMULATION OF COMPLEX MECHATRONIC SYSTEMS FOR REAL-TIME SIMULATION ON PARALLEL COMPUTERS

V. Enderlein[1], M. Hahn[2], U. Honekamp[2] and J. Obermüller[1],
[1]Institute for Mechatronics e.V.
✉Reichenhainer Str. 88, D - 09126 Chemnitz
✍{obermueller,v.enderlein}@ifm.tu-chemnitz.de
[2]Mechatronics Laboratory Paderborn (MLaP)
✉Pohlweg 55, D - 33098 Paderborn
✍{hahn, hone}@mlap.uni-paderborn.de

Abstract. This paper describes concepts developed and implemented in the joint research project METRO "Use of massively parallel computers for design and realization of complex mechatronic systems" that is funded by the German Federal Ministry of Education, Science, Research and Technology. The following institutions are involved in this project: GMD mbH Berlin, IfM e.V. Chemnitz, Universität-GH Paderborn (MLaP, HNI), C-Lab Paderborn, Daimler-Benz AG, ETAS GmbH & Co. KG and Fichtel&Sachs AG.

1 Introduction

At the development of new products, great effort is required for the design and manufacturing of prototypes. A significant reduction of this effort can be achieved by applying virtual prototyping methods, where the product is projected to a model, the behaviour of which can be simulated and investigated on computer (see Fig. 1). Therefore, case studies and design optimization can be performed quickly and cost-effectively on computer.

The product is represented by a complex model that includes electrical, electronic, mechanical, hydraulic, etc. subsystems. In the traditional design process the single subsystems are investigated separately. Therefore, the final tuning of entire product is very difficult. In contrast to that, the mechatronic approach takes into account the interaction between the different subsystems. The simulation of complex mechatronic systems is performed under realistic conditions on massively parallel hardware. In order to create hardware- or operator-in-the-loop solutions, the simulation has to be executed in real-time.

The aim of the project consists in developing software tools for conception, design and implementation of mechatronic systems which perform modelling, simulation, result analysis and optimization. Massively parallel computers are used as hardware platform for simulation because substructures of complex mechatronic systems work parallel. By adding new hardware modules, the parallel computers can be adapted to larger systems. During the design process, software components can be replaced by hardware components gradually without changing the system topology. In order to enable real-time simulation, both a special real-time operating system and compact computational models have to be developed.

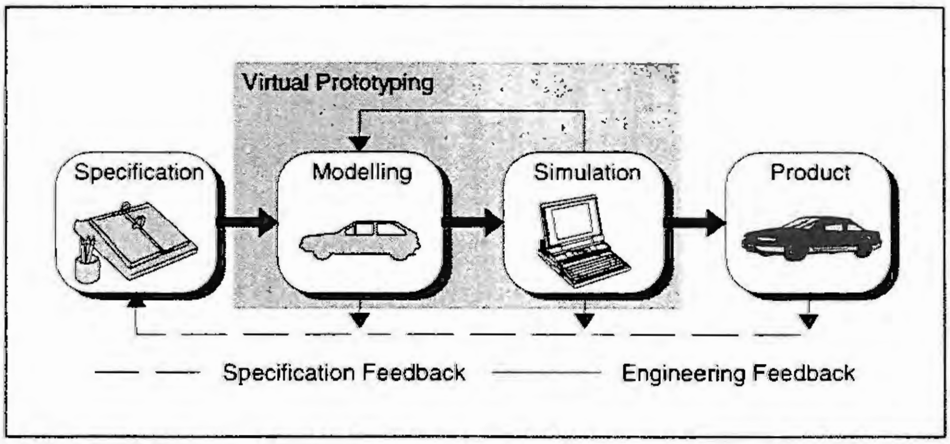


Fig. 1 Modelling and Simulation in the Product Development Process

As user frontend serves the so-called modelling tool which permits the user to integrate and couple several models of subsystems that are generated from different external modelling systems. The modelling tool offers a graphical representation of the entire model as topological scheme. It ensures that the couplings (physical, geometrical, information technical coupling) are consistent and provides a uniform interface for continuous and discrete substructures. The topological model description implies a hierarchical system of substructures that corresponds with the inner structure of the real system. This way, hardware components that replace software components within the hierarchy can be tested and optimized in the virtual entire system.

2 Components of a Modelling and Simulation System for Mechatronic Systems

The operations in the context of the METRO project comprise the model input, the preparation of the model for simulation, and making it run on a real-time simulation platform. These operations exact a neatly defined design methodology [7] and a clear, logical structuring of the working processes. Both in origin and in structure, the models to be interconnected are extremely heterogeneous and proceed from the most diverse tools. To generate the models in the context of the project the tools alaska, ASCET, CAMeL, MatrixX, and SEA are available. They provide partial models which may be both discrete and continuous and which must be able to run on a common simulation platform.

However, we do not have in view to integrate the above-mentioned modelling tools in order to make up an integral tool (*tool-integration*) but to integrate the models generated on the basis of the individual tools (*model-integration*). As an approach to a solution we developed the structure of modelintegration as shown in Fig. 2. In it the modelintegration process is divided into three parts:

- model level
- tool level
- operating-system level

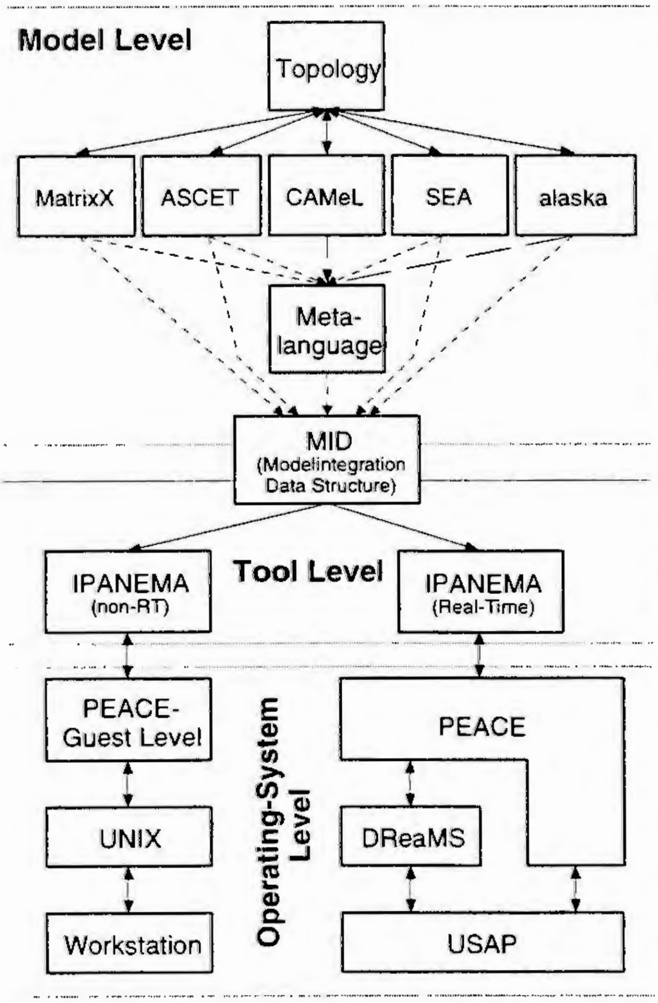


Fig. 2 Modelintegration in the Project METRO

A central part of the model level is the topology editor. Here the topological structure of a model means the interconnection, by means of physical or information-technical connections, of diverse components that may even be provided by different tools. The idea of how to describe the topology and elaborate a matching tool will be dealt with in chapter 2.1.

For a derivation of the mathematical model, different generation formalisms are integrated into the work flow. This is of special importance when formulating and computing multibody systems; for this purpose there are different formalisms to derive the mathematical model. The generation of the simulation code for the multibody system part by means of the program alaska and its capacity will be shown in more detail in chapter 2.2 and 2.3. The formalisms available for the modelling part of CAMeL to derive multibody systems are detailed in [2] and [5].

In practical operation an integral model is developed in a combination of partial models that have themselves been generated by means of different modelling tools. In contrast to the *simulator coupling* commonly used, it is the idea of *model coupling* that the project is centered on.

For this purpose a uniform model interface, the so-called modelintegration data structure (MID), was formulated. All partial models have to be prepared in a way as to be compatible with this interface. The simulation platform (on the tool level) has access to the individual models via the MID.

Below the tool level there is the operating-system level. The METRO project uses the operating system PEACE that exists for massively parallel computer networks and may as well be used, in the shape of a so-called host level, with common UNIX derivates. Thus, the user can have recourse to a mostly uniform operating-system interface, both for real-time simulation and for conventional computation.

2.1 Topology Editor

The simulation platform is essentially an extremely heterogeneous tool; to employ it properly the user needs support in the shape of a graphical tool encapsulating the model input. This level - in the following called the topological level - is used as the basis for implementing a corresponding modelling tool. For this purpose a graphical tool, the so-called topology editor, is under development; by integrating the different partial topologies it will guarantee a consistent access to the simulation platform.

In the modelling phase the user finds support in the topology editor that allows a fast and easy graphical assembly of systems. It visualizes not only the modular-hierarchical structure but also the coupling structure (the topological structure) of the systems. When visualizing the different description elements (relating to the mechanical and control-engineering parts as well as to code integration) the topology editor displays different icons symbolizing the special characteristics of each system. The icons are made available to the user in the shape of catalogues and may be interconnected to form new integral systems. All interventions by the user are then checked for admissibility and the newly generated models can either be inserted into the catalogue or saved as a new class of description elements. Please note that the model formulated in Objective-DSS contains no indication as to its graphical representation (placing of systems, routing of connections). The latter will automatically be derived from the coupling structure of the mechatronic system..

For this purpose the topology editor makes use of the model description language Objective-DSS (Objective-Dynamic System Structure [3]) designed at the MLaP for an object-oriented description of modular-hierarchical systems. For Objective-DSS we elaborated a large number of different description elements for a representation of mechanical and control-engineering systems that are available to the user in the shape of classes. Moreover, the user has the possibility to define discipline-specific classes as an extension (subclasses) to the basic classes whose objects can again be employed for model-description purposes (instantiation). Within the METRO project the class hierarchy will be extended by description elements for modelintegration on the basis of the modelintegration data structure (MID). Internally the modular-hierarchical integral system and its topological coupling structure (topological model) are represented in a description graph. This description graph will be built up as soon as the system described in Objective-DSS has been read from file and will serve as a basis for the derivation of the mathematical model.

2.2 Mechanical Substructures

The kernel of the multibody system (MBS) software alaska is used for the modelling and simulation of mechanical substructures of complex mechatronic systems based on the Lagrangian multibody system dynamics (see [8] and [9]).

alaska allows the automatic generation of both the nonlinear equations of motion for large scale motion and the linearized equations of motion (the so-called disturbed equations) for small perturbations nearby a static equilibrium position of an MBS. Mechanical systems either represent a kinematical tree structure (open chain) or a constrained mechanical system (CMS = kinematical tree structure with additional constraints). Bodies are coupled geometrically and/or physically to each other. Geometric coupling means that geometric objects of the coupled bodies coincide, physical coupling means that the state of motion is influenced by applied forces and torques.

For the model description alaska provides an input language which contains basic modelling elements for the mass geometry (mass, inertia tensor) and the kinematics (points, frames, joints, and initial values) as well as a number of special elements for the modelling of forces, torques, constraints etc. Especially the mathematical modelling elements such as arrays of arbitrary dimensions, numerical functions, the description of complicated functional relations by so-called symbolic functions, and the coupling of nonmechanical differential equation systems are essential for the modelling of applied forces, spatial force coupling elements, and control systems. Modelling elements for the facilitated description of complex physical couplings (tire, friction etc.) are available too [1].

One of the most important requirements of the real-time simulation is the determined number of operations per each simulation step (i.e. the knowledge of the computational effort). Within the MBS simulation this requirement is a priori fulfilled only in case of open chains and the use of fixed step integration methods, in case of CMS an iteration of the constraints is performed. Experiences have shown, if a smooth estimation of the initial position exists and the integration step size is sufficiently small, the iteration of the constraints will take only a constant number of steps. Hence the real time capability for the mechanical system can be evaluated in every case. The coupling of the mechanical subsystem with external subsystems is done strictly physically using the MID, a geometrical coupling is not intended.

2.3 Symbolic Code for Real-time Simulation

The tool alaska developed at the Institute of Mechatronics is used for modelling and simulation of multibody systems and the testing and investigation of stability of mechanical subsystems. alaska generates the equations of motion on the base of a numerical general purpose code. In general, that code is too slow for real-time simulation because a great portion of overhead (indexing, program flow control) is included and no advantage of the model properties is taken. A model dependent symbolic code generated by additional functions of alaska provides better efficiency (see [6]). Low overhead (only calls of subroutines to control the formalism: Lagrange - $O(n^3)$, Newton-Euler - $O(n)$) and use of model properties enable a fast evaluation of the governing equations.

In addition, certain simplification strategies can be applied to the symbolic code, e.g. elimination of expressions that have no influence on the acceleration and substitution of expressions that are used only one time (see [11]). Furthermore, application of numerical rules simplifies the code.

The generation of symbolic code has some advantages over binary code. The code is portable to various platforms, readable and can be used as input for computer algebra systems for further simplification or other investigation. Disadvantageously, the code has to be compiled before execution.

To link the generated code to the modelintegration data structure, the code is divided into parts for initialization, input/output and computation. The main part of initialization routines consists of code that computes a data structure for fast and efficient analysis from a set of variable parameters and constants. Input/output code controls the communication with external models of substructures.

The generated code for computation represents the multibody formalism and computes the derivatives for all state variables. To enable the computation of external forces or control forces depending on inner state of the multibody system, the computation code has to be divided into submodules. To perform parameter studies and system optimization on the simulation platform, specific parameters of the model should be included in symbolic form in the code and in the compiled version too. So the change of parameter values is possible on the simulation platform without new generation of code and recompilation.

2.4 Simulation Platform

Distributed simulation of the respective models requires a high-performance infrastructure tailored to this special purpose. We call this infrastructure the simulation platform. The one implemented for this project is based on the simulation platform IPANEMA [4] (Integration Platform for Networked Mechatronic Applications) developed at the MLaP. IPANEMA structures the individual partial tasks in distributed simulation in an object-oriented way. This results in four classes that form the basic structure of IPANEMA. Objects of the calculator class will implement the simulation kernels which will solve the equations of the individual partial models. Calculator objects will contain only indispensable functionality concerning administration and data management.

These tasks will be managed in a far more exacting way by objects of the assistant class. To each calculator object, an assistant is assigned. In a way the two of them form a team. Assistants relieve the burden of calculator objects and thus provide a clear distinction between those parts of a simulation environment which have to run under hard real-time conditions (calculator objects) and that part where no more than soft real-time conditions hold. Assistant objects encapsulate their corresponding calculator object against an object of the moderator class that acts as a kind of interface to the user and to the control stand on the one hand; on the other hand moderator objects take over management tasks, in a way: they coordinate the actions of the assistant/calculator teams whenever it becomes necessary (example: to start or stop the simulation).

Furthermore, assistant objects will act as a kind of high-performance cache memory in view of the moderator and the control stand. During the start-up phase the assistant evaluates all relevant information relating to the partial model simulated

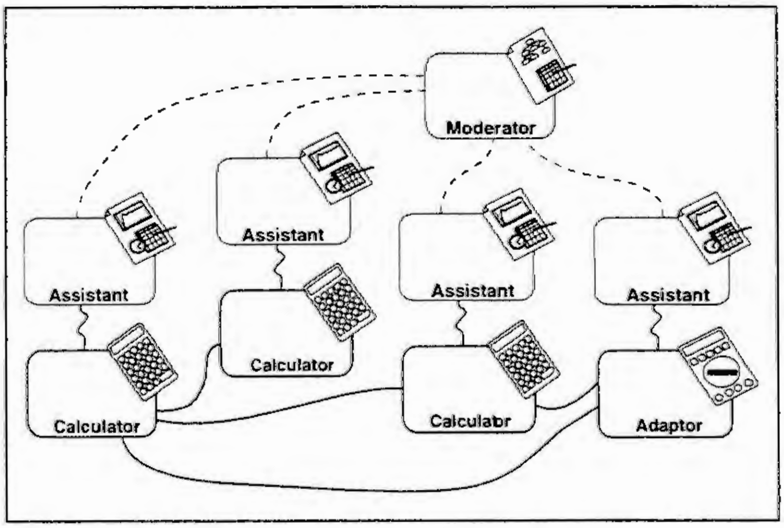


Fig. 3 Typical Topology of an IPANEMA Application

on the corresponding calculator. As far as necessary these information will be brought up to date at runtime. If for instance a variable value is asked for the assistant can provide the moderator with the value in question without bothering the calculator. For combining a technical process (which is the physical part of a mechatronic system) with digital information processing (in this case, the simulation) there is a certain class, called adaptor. Adaptor objects will convert the physical values relevant for the simulation into their numerical equivalents (including scaling and offset). Within an IPANEMA application adaptor objects have a status similar to that of calculator objects and are therefore also assigned an assistant object. Fig. 3 shows a typical object topology of an IPANEMA application.

3 Application: Hybrid Vehicles

The design cycle of complex mechatronic systems and the simulation platform are presented by modelling the operation strategy for a serial hybrid drive train. The serial hybrid drive train investigated in the project represents actually an electric vehicle where the electrical energy is supplied by the battery or by the combination consisting of internal combustion engine and generator. This hybrid configuration has the advantage that the vehicle can be driven purely electrically (e.g. in inner cities). If more power is required the combustion engine can be started. Thus, in contrast to purely electric vehicles, the operating range is similar to that of a conventional one.

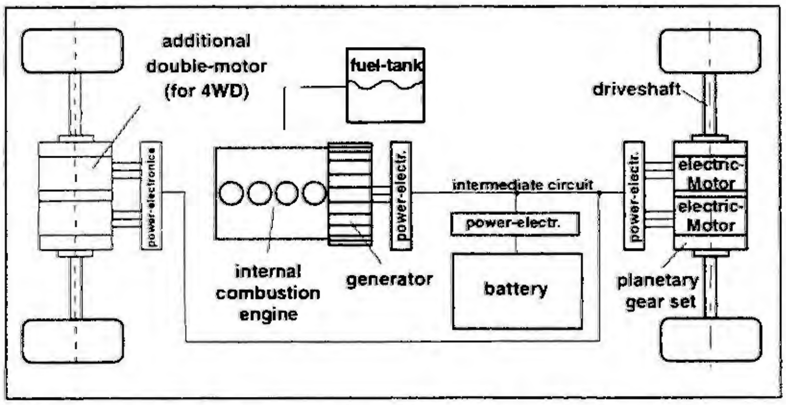


Fig. 4 The Hybrid Vehicle

The designed operation strategy is proved on a hardware-in-the-loop test-bench and will be enhanced and redesigned. Therefore the combustion engine and the electrical components of the drive train are arranged as hardware on the test-bench while all the other components of the real vehicle are simulated by mathematical models on multi processor hardware and additional actuators. The models of the subsystems are supplied from different software environments and companies, e.g. the Daimler-Benz AG delivers the mathematical model of the combustion engine, ETAS GmbH&Co. KG in cooperation with Fichtel&Sachs AG and the MLaP the models of the electric motors and the battery, and the Institute of Mechatronics the multibody models of the vehicle dynamics, respectively.

In the design and verification process of the operation strategy the tuning of single components is important to obtain a maximum total efficiency. On the other hand the improvement of the vehicle dynamics is aspired, e.g. the optimization of longitudinal vehicle dynamics (handling, acceleration behaviour etc.). Due to the separate control of each wheel the optimization of lateral vehicle dynamics is also possible [10].

4 Conclusions

The modelling and simulation of mechatronic systems are important parts of modern product development. Design tools should consider the complexity and the heterogenous structures of mechatronic systems as well as a quick and easy modelling and a simulation of these systems in real-time.

The presented concept corresponds to these requirements. Segmentation of the developed tool into modelling level, tool level and a level of operation system implies a good partitioning of model input, modelintegration and simulation of model applicated on parallel hardware. The user operates with the topology editor, which. represents the entire system or several subsystems consisting of different submodels and their couplings. Almost all other operations are executed automatically e.g. the integration of models of multibody systems into the entire model and the generation of the equations of motion. The applied tool alaska offers a variety of description and modelling capabilities for mechatronic or pure mechanical systems as well as the automatic derivative-free generation of extrem nonlinear or linearized motion equations. The large equation set is available as symbolic code suitable for simplification by computer algebra systems. The ability of code for real-time simulation is proved a priori by counting the operations. After that, the entire model is performed from the topological level to the simulation platform. The model is distributed to many computing kernels that communicate by assistants and moderators. Additional objects allow the linking of hardware components. A hardware-in-the-loop test-bench (vehicle with serial hybrid drive train) is built up simultaneously to the software development. The functioning and applicability of this tool is demonstrated by design and optimization of an operation strategy.

Acknowledgment. The research presented within this paper has been funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) under the name „METRO“. The authors want to thank their colleagues from MLaP and IfM for their support and the fruitfull discussions leading to some of the solutions presented in this paper.

5 References

1. alaska 2.3 User Manual. Institute of Mechatronics, 1996.
2. Hahn, M., Object-Oriented Physical Modelling of Mechatronic Systems. In: Mathematical Modelling of Systems. Vol. 1, No. 4, 1995, pp. 286-303.
3. Hahn, M. and Meier, U., Classification in the Object-Oriented Modelling Language Objective-DSS, Exemplified by Vehicle Suspensions. In: Proc. Symposium on Computer-Aided Control Design, Dearborn, USA.
4. Honekamp, U. and Stolpe, R., Design and Application of a Distributed Simulation- and Runtime-Platform for Mechatronic Systems in the Field of Robot Control. 3rd Conference on Mechatronics and Robotics, October 4-6, Paderborn, Germany, 1995.
5. Junker, F. and Hahn, M., Systematic Modelling of Mechanical Parts in Mechatronic Systems. In: Proc. IMACS-SAS 1995, 5th Int. IMACS Symp. on System Analysis and Simulation, Berlin 1995.
6. Keil, A., Härtel, T., and Freudenberg, H, Werkzeuge der Echtzeitsimulation von Mehrkörpersystemen. Zwischenbericht, Institut für Mechatronik e.V. an der TU Chemnitz-Zwickau, 10/94.
7. Lückel, J. and Wallaschek, J., Mechanical Design and Mechatronic Modelling. Accepted for: Second International Symposium on Mathematic Modelling, February 5-7, 1997, Technical University Vienna , Austria.
8. Maißer, P.: Analytische Dynamik von Mehrkörpersystemen. In: ZAMM 68(1988) 10, pp. 463-481.
9. Maißer, P.: A Differential Geometric Approach to Multi Body System Dynamics. In: ZAMM 71(1991) 4, pp. T116-T119.
10. Wältermann, P., Modelling and Control of the Longitudinal and Lateral Dynamics of a Series Hybrid Vehicle. In: 5th IEEE Conference on Control Applications (CCA), Sept. 15-18, 1996, Dearborn, Michigan, USA.
11. Weber, B., Symbolische Programmierung in der Mehrkörperdynamik. Dissertation, Universität Karlsruhe, 1993.

INCREMENTAL CONTROL OF THREE COOPERATING ROBOTS IN LARGE-OBJECT TRANSFER OPERATIONS

C.Tzafestas$^{(*)}$, S.Tzafestas$^{(**)}$ and P. Prokopiou$^{(**)}$

$^{(*)}$ Laboratoire de Robotique de Paris
CNRS-UPMC-UVSQ
10-12 Avenue de l'Europe, 78140 Vélizy, France.
e-mail: tzaf@robot.uvsq.fr

$^{(**)}$ Intelligent Robotics and Automation Laboratory
Department of Elctrical and Computer Engineering
National Technical University of Athens, GREECE
e-mail: tzafesta@softlab.ece.ntua.gr

Abstract. An incremental control algorithm is presented for three cooperating robot arms moving a large object from an initial to a desired final position/orientation. The robots hold the object at three points that define an isosceles triangle. Two particular cases are examined, namely: (1) master-and-two-slaves and (2) three cooperating robots. The method makes use of the differential displacement of the object which is transformed into that of the end-effector of each robot arm, and then the differential displacements of the joints of the robots are computed. A numerical simulation example is provided for three STAUBLI RX-90L robots, which shows the effectiveness of the method.

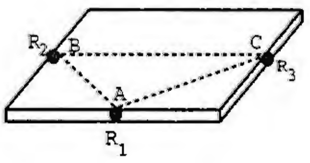
Keywords: Cooperating robots, incremental robot control, large-object robot operations

1 Introduction

Many industrial operations and tasks can be performed efficiently by a single robot. However, there are tasks which need two or more cooperating robots for satisfactory and economic performance. The case of two cooperating robots handling large objects or long flexible bars has been investigated by several researchers [1-10]. Most of these publications present theoretical investigations and only a few provide practical experimental studies (e.g.[9]). For example, in [2], feedback linearization is introduced, and the pole placement technique is used to the desired linear state-space model. In [3], each joint is controlled by a proportional type controller with the error being expressed in Cartesian space. In [4], the master-slave mode is considered, where the master arm is controlled by a position PID controller with a feedforward term and the slave moves in cooperation with the master while its force is controlled so as to balance the interactive force exerted by the master via the object. In [6], the controllers of the two arms are designed using the MIMO discrete ARX model with external inputs, where the parameters are estimated on-line recursively. Experimental real-time control results are presented in [9] for two PUMA 250 robot arms that manipulate large objects.

Although the capabilities of 2-robot-systems are substantially increased over single-robot-systems, they are still unable to handle (grasp, manipulate, transfer etc.) very large, very heavy or flexible objects. Therefore attention must be turned to the case of using three (or more) cooperating robots.
The purpose of this paper is to treat the 3-robot-arm case, by extending the technique used in [8], and to present an incremental control algorithm for moving large objects from an initial to a desired position/orientation, by holding it at three different points defining a triangle. Single robot tasks can be performed by controlling the robot's hand such as to follow a desired path, without controlling the exact time at which the hand passes through the particular points on the trajectory. The orientation of the robot's hand during the motion may also be irrelevant. This is not true in multi-robot systems, where, once the two or more hands grasp the object, their relative positions and orientations with respect to each other must remain invariant during the entire operation. Actually, in cooperating multi-robot systems each hand must pass through a particular point on its trajectory at exactly the right time, and the orientations of the hands must also be the proper ones.

In the 3-robot case (R_1, R_2, R_3):



three possible strategies can be followed:
(I) Master-Slave-Driver: Here the robot R_2 is the master, R_3 the slave, and R_1 the driver (that orients the plane ABC in space). (II) Master-and-two-Slaves: The motion of R_1 (master) follows directly from the motion planning, whereas the motion of R_2 and R_3 (the slaves) must also satisfy the constraints posed by the rigid object. (III) Cooperating robots: Here the motion planning is done by specifying the path that a fixed point of the object (e.g.the center of gravity) must follow. From this, the positions of the robot arms in the space are determined taking into consideration the constraints that are introduced by the object.

The paper presents two solutions. The first, through the homogeneous transformations by employing the "master-and-two-slaves" scheme, and the second through a path following technique using the "cooperating robots" scheme. The control of the robots is performed incrementally with the aid of the differential relations between the object and the three robot arms. The differential change of the object is transformed into that of each robot arm, and then the differential change of each joint of the three robots is derived. Numerical simulation results are provided which demonstrate the applicability and effectiveness of the proposed control algorithm.

2 3-Robot Arm Kinematics

In the following, the kinematics equations of the 3-robot arms system will be derived for the master-and-two-slaves configuration (using homogeneous transformations) and the three cooperating robots (continuous path in the space).

2.1 Master-and-two-slaves configuration

We consider the symmetric configuration shown in Fig.1.

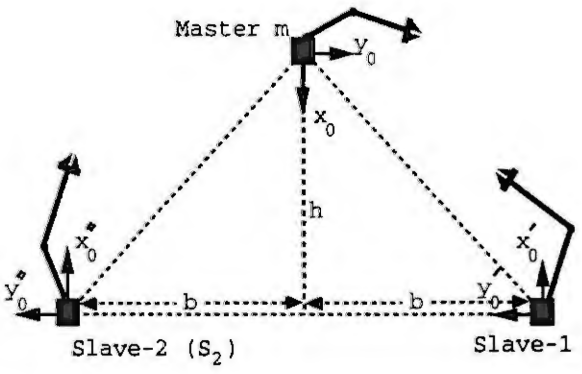


Figure 1: *Symmetric master-and-two-slaves configuration (all axes z_0, z_0', z_0'' are normal to the plane m-s_1-s_2).*

The world coordinate (w-c) system is defined to be the coordinate system $x_0y_0z_0$ of the master's base. Therefore the position and orientation of an object with respect to w-c is described by an homogeneous matrix A^m:

$$A^m = \left[\begin{array}{ccc|c} \underline{n} & \underline{o} & \underline{a} & \underline{p} \\ \hline & 0 & & 1 \end{array}\right], \qquad \underline{p} = [p_x, p_z, p_z]^T \tag{1}$$

where the vectors $\underline{n}, \underline{o}$ and $\underline{a}$ define the orientation of the object, and $\underline{p}$ its position (the position of the origin of the coordinate system $[\underline{n}, \underline{o}, \underline{a}]$). The position and orientation of the same object with respect to the coordinate systems of the bases of S_1 and S_2 is given by

$$A^{S_1} = S_1^{-1} A^m \qquad \text{and} \qquad A^{S_2} = S_2^{-1} A^m \tag{2}$$

where the matrices S_1 and S_2 define the coordinate systems of the slaves S_1 and S_2, respectively, and are given by (see Fig.1):

$$S_1 = \begin{bmatrix} -1 & 0 & 0 & h \\ 0 & -1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad S_2 = \begin{bmatrix} -1 & 0 & 0 & h \\ 0 & -1 & 0 & -b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

One can see that in this symmetric configuration: $S_i^{-1} = S_i$ (i=1,2), whereas the transformation from $x_0' y_0' z_0'$ to $x_0'' y_0'' z_0''$ is equal to:

$$S_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = S_{12}^{-1}$$

It must be remarked that in practice the distances b and h must be carefully selected and depend on the shape of the workspaces of the three robots as well as on the overall motion of the three-robot system. Usually, one can find optimum values of b and h that depend on the application at hand.

2.2 Transfer of a planar object

Here we consider a particular application where a "plane" (planar object) has to be transferred from an initial to a final position. The three robots grasp the object at three points A, B and C that define a triangle. Therefore one can either define the initial and final positions of the vertices of this triangle, or the initial and final position of the center of gravity of the triangle (fig.2).

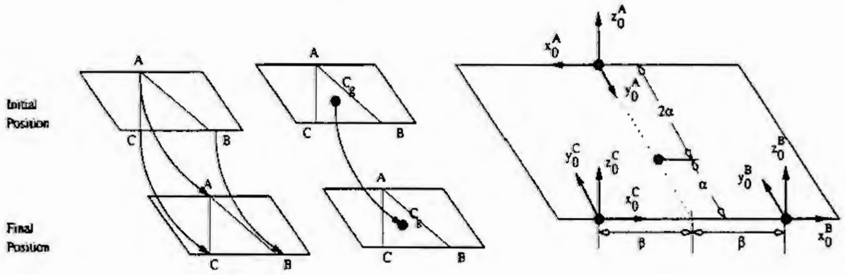


Figure 2: *Two ways of specifying the initial and final positions (C_g : center of gravity)*

The initial and final positions or the path of the object (defined in one of the two ways shown in fig.2) are used to determine the path (position and orientation) that must be followed by the end-effector of each arm. The position and orientation of an end-effector, with respect to the robot-base reference frame, is described by an homogeneous transformation (4x4 matrix) H of the type described by eq.(1).
The coordinate systems of the end-effectors and the grasping points are assumed as shown in fig.3, and therefore:

$$G^A = H^m \cdot \Theta\,, \qquad G^B = H^{S_1} \cdot \Theta\,, \qquad G^C = H^{S_2} \cdot \Theta \tag{4}$$

where G^A, G^B, G^C are the coordinate systems attached to the grasping points A, B and C of the master, slave-1 and slave-2 respectively, expressed in the corresponding robot reference frame, and:

$$\Theta = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

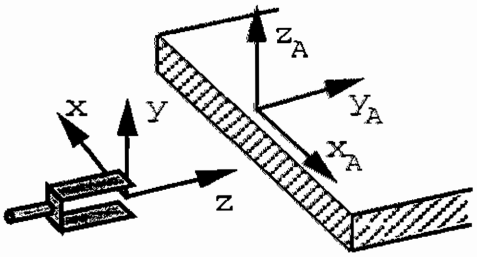


Figure 3: *Coordinate systems attached to the end-effector and grasping point*

The matrices that define the coordinate systems attached to B and C with respect to the coordinate system attached to A are:

$$K_B^* = \begin{bmatrix} -1 & 0 & 0 & -\beta \\ 0 & -1 & 0 & 3\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad K_C^* = \begin{bmatrix} -1 & 0 & 0 & \beta \\ 0 & -1 & 0 & 3\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

3 The Incremental Motion Control Algorithm

3.1 Absolute motion equations

Our purpose here is to define the motion of the planar object in space. Consider first the motion of the point A (grasped by the master arm). This motion is defined by a time-varying homogeneous transformation matrix $M(t)$ which determines the linear and angular displacements needed for the point A to go from the initial to the desired final position and orientation. The matrix $M(t)$ is given by

$$M(t) = \begin{bmatrix} r_x r_x v(\tau\phi) + c(\tau\phi) & r_y r_x v(\tau\phi) - r_z s(\tau\phi) & r_z r_x v(\tau\phi) + r_y s(\tau\phi) & \tau \cdot x \\ r_x r_y v(\tau\phi) + r_z s(\tau\phi) & r_y r_y v(\tau\phi) + c(\tau\phi) & r_z r_y v(\tau\phi) - r_x s(\tau\phi) & \tau \cdot y \\ r_x r_z v(\tau\phi) - r_y s(\tau\phi) & r_y r_z v(\tau\phi) + r_x s(\tau\phi) & r_z r_z v(\tau\phi) + c(\tau\phi) & \tau \cdot z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where $\tau = t/t_f$ is normalized time (t_f is the time in which the motion has to be completed), $s(\cdot) = \sin(\cdot)$, $c(\cdot) = \cos(\cdot)$, $v(\cdot) = 1 - \cos(\cdot)$, $\underline{p} = [x, y, z]^T$ is the position displacement vector from the initial to the final position, and the vector $\underline{r} = [r_x, r_y, r_z]^T$ defines the axis about which the initial coordinate system must rotate by an angle ϕ_* to obtain the final orientation.

Now, if $G^A(0)$ is the matrix defining the initial position/orientation of the point A, then the time-varying position/orientation of A with respect to the w-c system is given by

$$G^A(t) = G^A(0)M(t) \tag{8}$$

and the final one is given by
$$G_f^A = G^A(t_f) = G^A(0)M(t) \tag{9}$$

where

$$x = \underline{n}^T(0)\left[\underline{p}(t_f) - \underline{p}(0)\right], \qquad y = \underline{o}^T(0)\left[\underline{p}(t_f) - \underline{p}(0)\right], \qquad z = \underline{a}^T(0)\left[\underline{p}(t_f) - \underline{p}(0)\right] \tag{10a}$$

$$\phi_* = \cos^{-1}\left[\tfrac{1}{2}\underline{n}^T(0)\underline{n}(t_f) + \underline{o}^T(0)\underline{o}(t_f) + \underline{a}^T(0)\underline{a}(t_f) - 1\right] \tag{10b}$$

$$\underline{r} = \begin{bmatrix} \underline{a}^T(0)\underline{o}(t_f) - \underline{o}^T(0)\underline{a}(t_f) \\ \underline{n}^T(0)\underline{a}(t_f) - \underline{a}^T(0)\underline{n}(t_f) \\ \underline{o}^T(0)\underline{n}(t_f) - \underline{n}^T(0)\underline{o}(t_f) \end{bmatrix} \tag{10c}$$

The motion of the points B and C of the object is defined by

$$G^B(t) = S_1 G^A(0)M(t)K_B^* \qquad \text{and} \qquad G^c(t) = S_2 G^A(0)M(t)K_C^* \tag{11}$$

The motion of the end-effectors of the three arms grasping the points A, B and C is defined by the transformations $H^m(t)$, $H^{S_1}(t)$ and $H^{S_2}(t)$, which can be determined by equating the right-hand sides of (4) and(8),(11) respectively, and solving the resulting equations:

$$H^m(t) = G^A(0)M(t)\Theta \;, \qquad H^{S_1}(t) = S_1 G^A(0)M(t)K_B^*\Theta \;, \qquad H^{S_2}(t) = S_2 G^A(0)M(t)K_C^*\Theta \tag{12}$$

where the relations $S_i^{-1} = S_i$ (i=1,2) and $\Theta^{-1} = \Theta$ were used.

3.2 Differential motion equations

We now determine the differential motion equations of the three-robot arm system. Let

$$D = [d_x, d_y, d_z; d\phi_x, d\phi_y, d\phi_z]^T$$

the differential motion vector where d_x, d_y, d_z are differential linear displacements and $d\phi_x, d\phi_y, d\phi_z$ are differential angular displacements with respect to the axes x,y,z respectively.

Consider the grasping point A. The coordinate system of A at time $(t + dt)$ is given by

$$G^A(t + dt) = G^A(t) + dG^A(t) = G^A(t) \cdot [I + \Delta] \tag{13a}$$

where I is the 4x4 unit matrix, and
$$\Delta = \begin{bmatrix} 0 & -d\phi_z & d\phi_y & dx \\ d\phi_z & 0 & -d\phi_x & dy \\ -d\phi_y & d\phi_x & 0 & dz \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{13b}$$

Similarly, the differential transformations for the three arms are defined by

$$H^k(t + dt) = H^k(t)[I + \Delta^k] \tag{14a}$$

$$\Delta^k = \begin{bmatrix} 0 & -d\phi_z^k & d\phi_y^k & dx \\ d\phi_z^k & 0 & -d\phi_x^k & dy \\ -d\phi_y^k & d\phi_x^k & 0 & dz \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{14b}$$

for k=m, s_1 and s_2 respectively.
From the analysis of section 3.1:

$$H^m(t) = G^A(t)\Theta \;, \qquad S_1 H^{S_1}(t) = G^A(t)K_B^*\Theta \;, \qquad S_2 H^{S_2}(t) = G^A(t)K_C^*\Theta \tag{15}$$

Similar equations hold for the time instant $(t + dt)$.
Now using (13a,b), (14a,b) and (15), and solving for Δ^m, Δ^{S_1} and Δ^{S_2} we obtain:

$$\begin{aligned} \Delta^m &= [H^m(t)]^{-1} G^A(t)\Delta\Theta \\ \Delta^{S_1} &= [H^{S_1}(t)]^{-1} S_1 G^A(t)\Delta K_B^*\Theta \\ \Delta^{S_2} &= [H^{S_2}(t)]^{-1} S_2 G^A(t)\Delta K_C^*\Theta \end{aligned}$$

which by (15) reduce to: $$\Delta^m = \Theta\Delta\Theta, \qquad \Delta^{S_1} = \Theta K_B^*\Delta K_B^*\Theta, \qquad \Delta^{S_2} = \Theta K_C^*\Delta K_C^*\Theta \tag{16}$$

Equations (16) give the differential displacements of the three robots end-effectors in terms of the differential displacement matrix Δ of the point A. Using (13b) and the definition of Θ in (5) we get the following:

Master Arm

$$\begin{aligned} & d\phi_x^m = -d\phi_x, \; d\phi_y^m = d\phi_z \;, \; d\phi_z^m = d\phi_y \\ & dx^m = -dx, \; dy^m = dz, \; dz^m = dy \end{aligned}$$

Slave-1

$$\begin{aligned} & d\phi_x^{s_1} = d\phi_x, \; d\phi_y^{s_1} = d\phi_z, \; d\phi_z^{s_1} = -d\phi_y \\ & dx^{s_1} = dx - 3\alpha d\phi_z, \; dy^{s_1} = dz + \beta d\phi_y + 3\alpha d\phi_x, \; dz^{s_1} = -dy + \beta d\phi_z \end{aligned}$$

Slave-2

$$d\phi_x^{s_2} = d\phi_x, \; d\phi_y^{s_2} = d\phi_z, \; d\phi_z^{s_2} = -d\phi_y$$

$dx^{s_2} = dx - 3\alpha d\phi_z$, $dy^{s_2} = dz - \beta d\phi_y + 3\alpha d\phi_x$, $dz^{s_2} = -dy - \beta d\phi_z$

3.3 The incremental motion control algorithm

To develop the incremental motion control algorithm (for each robotic arm) the total linear and angular displacement of the point A ($\underline{p} = [x, y, z]^T$ and ϕ_*) given by (10a,b) is divided in a large number of small (nearly infinitesimal) displacements $\delta\underline{p}$ and $\delta\phi$. From these displacements and the above relations one can compute the corresponding displacements $\delta\underline{p}^m$, $\delta\phi^m$; $\delta\underline{p}^{s_1}$, $\delta\phi^{s_1}$; $\delta\underline{p}^{s_2}$, $\delta\phi^{s_2}$ of the three arms.

Let q_i (i=1,2,...,6) be the displacement of each joint, and dq_i the corresponding differential displacement. Then we can write:

$$H^m \cdot [dx^m, dy^m, dz^m; d\phi_x^m, d\phi_y^m, d\phi_z^m]^T = J^m(q_1, \ldots, q_6) \cdot [dq_1^m, dq_2^m, \ldots, dq_6^m]^T \tag{17}$$

where J^m is the Jacobian matrix of the master arm. Similar equations hold also for the slave arms.

Given the displacements $\delta x^m, \ldots, \delta\phi_z^m$ (determined as discussed previously) one can find the corresponding δq_i^m (i=1,...,6), by solving equation (17). On the basis of the above analysis the incremental motion control algorithm is as follows.

- Initialization: Determine the initial position (the q_i's) of each robotic arm, and the final position/orientation of the master arm. Also specify the desired time t_f for the task completion.

- Step 1: Compute the linear displacement vector $\underline{p} = [x, y, z]^T$, the axis of rotation $\underline{r} = [r_x, r_y, r_z]$ and the total rotation angle ϕ_* from equations(10a,b,c). Determine the number of elementary segments to which the motion from the initial to the final position/orientation will be splitted, and compute the corresponding $d\underline{p}$ and $d\phi$ of each one of them.

- Step 2: Set $\delta q_i = 0$ $(i = 1, 2, \ldots, 6)$

- Step 3: At each time t compute $\delta\underline{p}^m$, $\delta\phi^m$; $\delta\underline{p}^{s_1}$, $\delta\phi^{s_1}$; $\delta\underline{p}^{s_2}$, $\delta\phi^{s_2}$; using (16).

- Step 4: Using the $\delta\underline{p}^j$ and $\delta\phi^j$ $(j = m, s_1, s_2)$, found in step 3, compute the δq_i^j $(j = m, s_1, s_2; i = 1, \ldots, 6)$ by solving the Jacobian equation (17).

- Step 5: Update the q_i^j's as : $\quad q_{i,new}^j = q_{old}^j + \delta q_i^j$

  and repeat from step 3, until the final time t_f is reached. Here of course $q_{i,new}^j$ is used as initial value of q_i^j at the next time instant t+δt.

4 Simulation Example

In this study, three Staubli RX-90L robots are used. RX-90 has a kinematic structure similar to that of a PUMA 700 robot with 6 rotational degrees of freedom and a spherical workspace of a radius around 120cm.

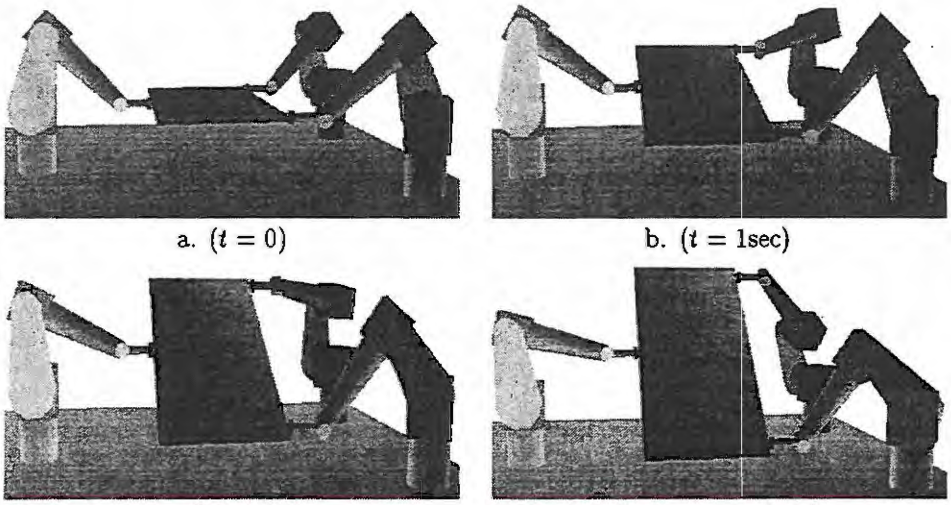
A series of numerical simulations has been performed to verify the applicability of the proposed method. The modelled task consists of picking up a horizontal plate and performing a vertical translation of 30 cm as well as a rotation of 40 degrees about an axis parallel to the x-axis of the master-robot coordinate frame. The dimensions of the plate are taken to be (180x80x4)cm.

Initial and final configurations (as well as two intermediate ones) are shown in fig.4. The motion of each robot is planned by making small incremental, linear and angular displacements, as discussed in section 3. In order to test the efficiency of the method, we varied the number N of increments. To evaluate quantitavely the performance of the algorithm we used a "relative-positionning error" measure ϵ_p, defined as

$$\epsilon_p = \sqrt{e^2_{p_{s1,m}} + e^2_{p_{s2,m}} + e^2_{p_{s2,s1}}}$$

where $\qquad e^2_{p_{j,i}} = \left|\underline{p}_{j,i}^{(i)} - \underline{d}_{j,i}^{(i)}\right|^2 \;(\, i = m, s_1 \;, \quad j = s_1, s_2 \quad \text{with } i \neq j)$

$\underline{p}_{j,i}^{(i)}$: the relative position of the j-robot end-effector, with respect to the i-robot endpoint, expressed



a. $(t = 0)$

b. $(t = 1\text{sec})$

c. $(t = 2\text{sec})$

d. $(t = t_f = 3\text{sec})$

Figure 4: *Graphical Animation of the simulated 3-robot coordination task. A sequence of configurations : initial (t=0), intermediate (t=1, 2sec) and final configuration (t=3sec).*

in the i_{th} robot local tool frame.

$\underline{d}_{j,i}^{(i)}$: the desired (reference), relative-position vectors from the i- to the j-robot end-effector, expressed in the local i_{th} tool frame. These reference position vectors are imposed by the geometry of the manipulated object and the choice of the grasping points. In our case:
$\underline{d}_{s_1,m}^{(m)} = [\beta, 0, 3\alpha]^T, \underline{d}_{s_2,m}^{(m)} = [-\beta, 0, 3\alpha]^T, \underline{d}_{s_2,s_1}^{(s_1)} = [2\beta, 0, 0]^T$.

This error gives a measure of the magnitude of the "internal forces" that may appear during execution of the task. Fig.5 shows the results obtained for three different numbers N of differential increments (N=40, 400, 1000) and $t_f = 3$ sec. The presence of cumulative errors is practically eliminated (inferior to 1mm) if sufficient number of steps (N=400, 1000) is used, which corresponds to a differential linear displacement of less than 1mm and a differential angular displacement of 0.1 degrees or less. Satisfying these conditions, the obtained results show that the proposed method can be easily implemented and efficient for the case of three-robots coordinated task.

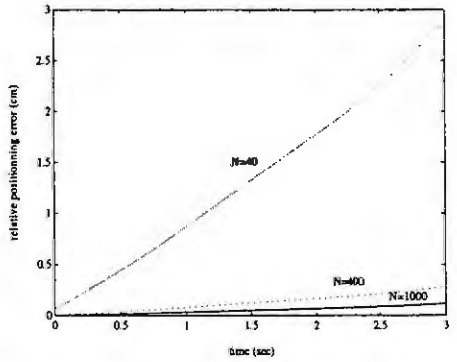


Figure 5: *Relative-positionning error for the robots' end-effectors*

5 Conclusion

A path planning method for the trajectory control of three cooperating robots is presented in this paper. The proposed algorithm consists of performing incremental, linear and angular displacements which are computed, using homogeneous transformations, from the desired motion of the manipulated object.

Numerical simulations show the applicability and the effectiveness of the proposed method, under certain conditions regarding the magnitude of the differential displacements, which is related to the number of increments used. Nevertheless, complete elimination of cumulative errors may require the use of the inverse geometric model of the robots in a periodic way, in order to reinitialize the undesirable resulting relative positionning errors.

References

[1] Alford,C.O. and Belyeu,S.M. "Coordinated control of two robot arms", *Proc. IEEE Intern. Conference on Robotics and Automation*, Atlanta, GA, March 1984, 468-473.

[2] Freund,E. "On the design of multirobot systems", *Proc. IEEE Intern. Conf. on Robotics and Automation*, Atlanta, GA, March 1984, 477-490.

[3] Fujii,S. and Kurono,S. "Coordinated computer control of a pair of manipulators", *Proc. 4th IFTOMM World Congress*, Univ.Newkastle, U.K., Sept.1975, 411-417.

[4] Ishida,T. "Force control in coordination of two arms", *Proc. 5th Int. Conf. on Artificial Intelligence*, August 1977, 411-415.

[5] Kim,K.I. and Zheng,Y.F., "Two strategies of position and force control for two industrial robots handling a single object", *Robotics and Autonomous systems*, 5 (1989), 395-403.

[6] Koivo,A.J. "Adaptive position-velocity-force control of two manipulators", *Proc. 24th IEEE Conf. on Decision and Control*, Ft.Lauderdale, Fl., Dec.1985, 334-337.

[7] Koivo,A.J. and Bekey,G.A., "Report of workshop on coordinated multiple robot manipulators: Planning, control and applications", *IEEE Journal Robotics and Automation*, 4(1), 1988, 91-93.

[8] Lim,J. and Ghyung,D.H., "On a control scheme for two cooperating robot arms", *Proc. 24th IEEE Conf. on Decision and Control*, Ft.Lauderdale, Fl., Dec.1985.

[9] Paljug,E. and Yun,X., "Experimental study of two robot arms manipulating large objects", *IEEE Trans. Control Syst. Tech.*, 3(2) (1995), 177-188.

[10] Zheng,Y.F. and Luh,J.Y.S. "Control of two coordinated robots in motion", *Proc. IEEE Conf. on Decision and Control*, Ft.Lauderdale, Fl., Dec.1985, 1761-1764.

A CASE OF INTELLIGENT AUTONOMOUS ROBOT MANAGING UNCERTAINTY BY MEANS OF MULTISENSOR FUSION

F. Matía and A. Jiménez
DISAM - Univ. Politécnica de Madrid
José Gutiérrez Abascal 2, E-28006 Madrid (SPAIN)
e-mail: matia@disam.upm.es

Abstract. A case of conventional Mobile Robot with the features of autonomy and intelligence is introduced. The topics of multisensor fusion, observation integration and sensor coordination are widely used along the article. The final goal is to demonstrate the validity of both mathematical and artificial intelligence techniques to guarantee the vehicle survivement in a dynamic environment, while the robot carries out an specific task. We review conventional techniques for the management of uncertainty while we describe an implementation of mobile robot which combines on-line disparate sensors in its navigation and location tasks.

1 Introduction

A mobile robot may be considered an Intelligent Autonomous System (IAS) in the sense that: i) the complete navigation system resides on an on-board computer, and the vehicle is completely wireless (autonomy); and ii) the robot has some kind of reasoning capability which allows it to make its own decisions, and to appropriately select, fuse and integrate heterogeneous sensor data (intelligence).

The main task of our robot is to reach a goal following a path. But the intelligence of such an IAS could be reduced to planning and control if there were not uncertainty present in sensors nor in the environment. If we restrict planning to decompose a mission into elemental tasks, or to plan a path between two points, given a map of the environment, non uncertainty is taken into account. On the other hand, the control system calculates the appropiate velocity commands to follow the reference path, and uncertainty is present in the odometric and navigation sensors.

Without precise sensor data, control objetives will never be reached. To cope with uncertainty modeling in Mobile Robotics, well-known state estimation (location) and navigation algorithms exist (kalman filters, probability theory, Dempster-Shafer, decision theory, fuzzy logic, ...). The contribution of this work is the fusion of disparate sensors in a concrete mobile robot which uses all its sensorial capability to reduce uncertainty [6] while it navigates avoiding obstacles, integrates new observations to refresh environment maps, and coordinates vision systems to improve location estimation. Advanced robot control techniques are completed with Artificial Intelligence in the control module, using fuzzy logic and/or neural networks.

2 The Mobile Robot and its Environment

Figure 1 shows the case of a mobile robot. The mobile platform is a Robuter vehicle from the French company Robosoft. The perception system is composed of: i) a ring of 24 sonars used for reactive navigation and cell map generation, ii) an infrared laser diode and an infrared camera situated in front of the robot, used for location and geometric map building, and iii) a CCD color camera with pan and tilt movements situated on the top of the robot, used for location.

The environment is a 2D room with known fixed obstacles, fixed beacons with known 3D location, and possible unknown obstacles.

2.1 Environment Models

Two different environment models are used: occupancy grid model and geometric model. In the first one, the 2D environment is divided in a grid with an occupancy value: 0 for empty and 1 for occupied. A variant of this model are quadtrees, in which the initial space is recursively divided (if necessary) into four equal parts until all objects fill in a cell.

In a 2D geometric model, each object is represented by a set of segments [1]. Each line is represented with two parameters: a distance d and an angle α. The path planning module of our mobile robot uses this geometric model with two different algorithms: Voronoi diagrams (in environments with high density of obstacles) and visibility graphs (in environments with low density of obstacles).

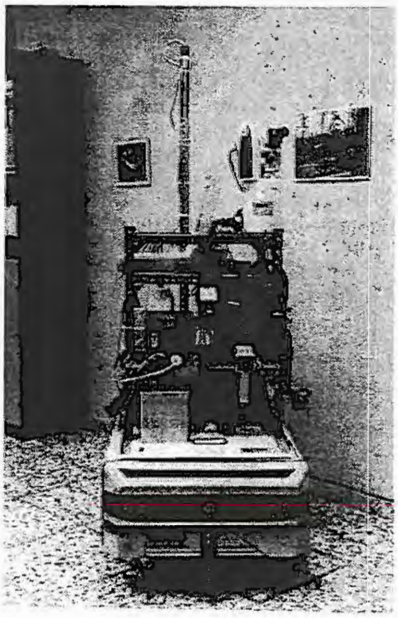


Figure 1: The Mobile Platform

2.2 Sensor Models

The information given by each sensor is modeled to compare the real measurements with the estimates. This will later allow to improve sensor information. Since all sensor model parameters are more or less estimated amount of uncertainty, a probabilistic feature must be added to the sensor model to represent the certainty of the measure [4].

The **odometric model** corresponds with the kinematic model of the mobile platform: $\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k)$, where $\mathbf{x}^T(k) = [x(k), y(k), \theta(k)]$ is the robot location (oriented position) at instant k, $\mathbf{f}$ is the cinematic model, $\mathbf{u}(k)$ is the velocity command vector, and $\mathbf{v}(k)$ is the model noise vector (uncertainty is included here). The **sonars** model follows:

$$p(z(k) = z|l) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(z-l)^2}{2\sigma^2}\right) \tag{1}$$

where p is the probability density function (uncertainty is included), $z(k)$ is the sensor measure, l is the distance of the closest object and σ is a parameter to be determined empirically. In the case of the occupancy grid model, uncertainty is refreshed using Bayes (see [5]):

$$P(s_i(k+1)|z(k+1)) = \frac{p(z(k+1)|s_i(k))P(s_i(k)|z(k))}{\sum_{s_i} p(z(k+1)|s_i(k))P(s_i(k)|z(k))} \tag{2}$$

where $s_i(k)$ is the state (occupied or free) of cell i at instant k. An example of sonar map, using a 24 sonar ring, is shown in figure 2 (ii). The set **camera-infrared laser** supplies a set of infrared ligth points in the image frame $\mathbf{x_I}(k)$, which may be transformed into the robot coordinate frame $\mathbf{x_R}(k)$. The noiseless equations follow:

$$x_R(k) = x_I(k) + D\frac{1 + \frac{y_I(k)}{(1+gr^2(k))f}tan\theta}{tan\theta - \frac{y_I(k)}{(1+gr^2(k))f}} + L_X \tag{3}$$

$$y_R(k) = y_I(k) + \frac{x_R(k)x_I(k)}{(1+gr^2(k))fcos\theta} + L_Y \tag{4}$$

with L_X and L_Y the horizontal position of the camara, f the focal distance, D its heigh, θ the camera angle, g the image distorsion factor, and $r^2(k) = x_I^2(k) + y_I^2(k)$.

The robot coordinate frame must be transformed later into the origin coordinate frame $\mathbf{x_O}(k)$. Again the noiseless equation follows:

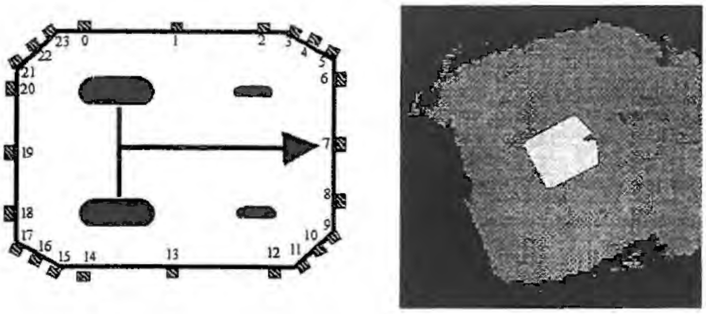


Figure 2: (i) Sonar ring; (ii) Probabilistic map

$$\mathbf{x_O}(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} cos\theta(k) & sin\theta(k) \\ -sin\theta(k) & cos\theta(k) \end{bmatrix} \mathbf{x_R}(k) \tag{5}$$

From all the light points (see figure 3), a segment extraction is carried out, so a set of measured segments (d, α) is obtained from the model $\mathbf{z}(k) = \mathbf{h}(k, \mathbf{x}(k)) + \mathbf{w}(k)$, where $\mathbf{z}^T(k) = [d(k), \alpha(k)]$ is the segment position at instant k, $\mathbf{h}$ is the sensor model, $\mathbf{x}(k)$ is the robot location, and $\mathbf{w}(k)$ is the model noise vector (measure uncertainty).

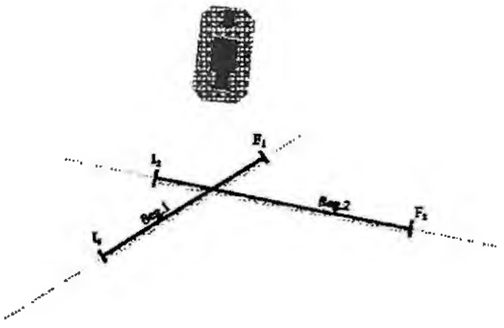


Figure 3: Camera-infrared laser

The **CCD color camera** is used to measure the 3D position $\mathbf{x_B}(k)$ of several artificial beacons. Again the beacon corrdinates are transformed into the origin coordinate frame through an equation similar to (5). The sensor model also has the form $\mathbf{z}(k) = \mathbf{h}(k, \mathbf{x}(k)) + \mathbf{w}(k)$. Special attention must be paid to the fact that the three perception systems must be calibrated previously to minimize uncertainty in model parameters. Sonars may be calibrated for large or small ranges, and camera parameters must be estimated to make proper use of the last equations.

3 MultiSensor Fusion in Navegation

An intelligent control module resides in the mobile robot for safe navigation. Its main goal is to follow a path with real time obstacles avoidance. Two control paradigms are available to the control supervisor: fuzzy logic for static obstacles and neural networks for multirobot systems. We will discuss the first case, since multirobot systems are not a topic of this article.

Our mobile robot uses the reactive architecture AFREB: Adaptive Fusion of Reactive Behaviors (see [7]). It is composed of two levels: i) the lowest one includes elemental controllers which represent primitive behaviours (see [2]) as follow a path, follow obstacle contour (left or right), and turn (left or right); and ii) a fuzzy decision module which fuses the primitive behaviors originating an emergent behavior (following a path with obstacles avoidance). Figure 4 shows the reactive control scheme.

The decision module generates a weight $a_i(k)$ for each primitive behaviour i, so they may be fused as follows:

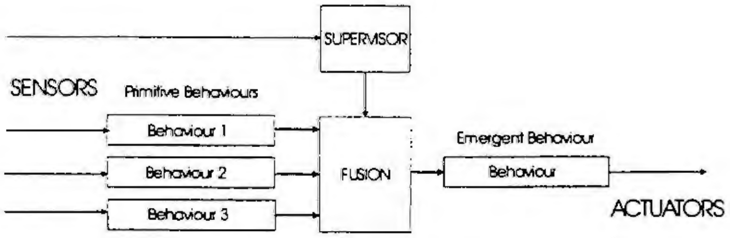


Figure 4: Reactive Control Architecture

$$\mathbf{u}(k) = \frac{\sum a_i(k)\mathbf{u}_i(k)}{\sum a_i(k)} \tag{6}$$

where $\mathbf{u}_i(k)$ is the velocity command of the primitive behavior i, and $\mathbf{u}(k)$ is the final velocity command for the mobile robot. The fusion supervisor must determine the most adequate value for the weights a_i. The fusion rules are like follows:

if *the minimum distance is medium* and *the sensor is in the right hand*
then *the weight of the behaviour follow path is medium*
and *the weight of the behaviour right contour following is medium*

In this case, 24 sonars are fused into the regions left, left front, front, right front and right. A second fuzzy decision module which implements heterogeneous sensor fusion of sonar and laser measurements is also available. The rules try to model the following reasoning:

if *the front space is wide enough* (laser)
then *center the robot* (sonars)

4 Observations Integration for State Estimation

The control module must keep an accurate estimation of the mobile robot location at each moment. This state estimation $\hat{\mathbf{x}}(k)$ is obtained using the extended Kalman filter. The algorithm works integrating all sensor readings into a more precise measure which allows to predict the robot state.

Three levels of state estimation are available: i) the odometry system, which gives an initial but unaccurate estimation of the location based on the last position and the encoders readings, ii) the set camera-infrared laser, which uses a map of the environment and compares it with the light points measures, and iii) the CCD color camera which uses the known locations of several artificial beacons and compares them with the information obtained from the actual image.

The incomming sensor readings are integrated as follows [3]. While the robot is moving following a path (or avoiding and obstacle), the incomming state from the odometry system $\hat{\mathbf{x}}(k|k)$ is improved on-line by integrating the laser measures. The extended kalman filter has four steps in each cycle:

1. **prediction** of new measures:
$$\hat{\mathbf{x}}(k+1|k) = \mathbf{f}(k, \hat{\mathbf{x}}(k|k), \mathbf{u}(k))$$
$$P(k+1|k) = \mathbf{F}_{\hat{\mathbf{x}}}(k)P(k|k)\mathbf{F}_{\hat{\mathbf{x}}}{}^T(k)$$
$$\hat{\mathbf{z}}(k+1|k) = \mathbf{h}(k+1, \hat{\mathbf{x}}(k+1|k))$$
where $\mathbf{F}_{\hat{\mathbf{x}}}$ is the jacobian matrix for $\mathbf{f}$, and P is the covariance matrix for $\hat{\mathbf{x}}$.
2. **observation** of new measures: $\mathbf{z}$(k+1)
3. **matching** of predicted and observed measures:
$$\nu(k+1) = \mathbf{z}(k+1) - \mathbf{z}(k+1|k)$$
$$\mathbf{S}(k+1) = \mathbf{H}_{\mathbf{x}}(k+1)P(k+1|k)\mathbf{H}_{\mathbf{x}}{}^T(k+1)$$
where ν is the measure innovation and $\mathbf{S}$ is its covariance matriz.

4. **State estimation:**

$$\mathbf{W}(k+1) = P(k+1|k)\mathbf{H_x}^T\mathbf{S}^{-1}(k+1)$$

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{W}(k+1)\nu(k+1)$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1)$$

In the case of the set camera-infrared laser, the estimated measures $\hat{\mathbf{z}}(k+1|k)$ are the segments from the static known map, while the real measures $\mathbf{z}(k+1)$ are the segments extracted from the real world with the laser. If the difference between both, $\nu(k+1)$, is less than a fixed value (applying Mahalanobis distance), we can use the Kalman filter to obtain the new position $\hat{\mathbf{x}}(k+1|k+1)$. If not, this means that a new object has been detected in front of the vehicle, so we may use probabilistic techniques (Bayes) to add these new segments to the map (see next section).

In the case of the color camera, artificial vision techniques are used. From time to time (when the uncertainty $\mathbf{P}(k|k)$ is high enough) the vehicle searches for landmarks in the environment, with a well known position in the environment. This beacons may be artificial or natural (e.g. a window) and are defined in the geometric map.

The color camera looks towards them, extracts from the image the desired features and measures their position $\mathbf{z}(k+1)$. As in the previous case, by comparing this position with the model $\hat{\mathbf{z}}(k+1|k)$, we can improve the estimation of the mobile robot location.

5 Map Building and Sensor Planning

In a previous section we have shown the possibility of updating propability grids by integrating sonar information. Now we are going to discuss briefly how to integrate new objects into a geometric map [8]. In the last section we said that only new meassures from laser which are inside Mahalanobis distance are to be fused into the Kalman filter.

But outside segments may be used for map building. The steps for map construction on the fly follows: i) fusion of light points into segments; ii) fusion of similar segments; iii) cut of extralarge segments; iv) dynamic refresh of segments certainty; v) elimination of segments with very low certainty; and vi) fusion of segments with very high certainty into objects.

On the other hand, examples of sensor planning and coordination may be found in the case of two mobile robots which syncronize their sonars when their position is close, to avoid interferences. But our case of mobile robot uses sensor planning for the coordination of multiple sensors (sonars, laser and cameras). The two methods for position estimation are to be coordinated: i) the use of structured light; and ii) the use of artificial vision.

While the robot is moving, a continuous localization is being carried out with the laser. When the sensor planner detects an excesive increase in the uncertainty (perhaps no segments are near in front of the robot), it stops the robot and takes advantage of the localization with the color camera. After uncertainty has decreased, the planner allows the robot to continue his previous task.

Figure 5 shows the robot following a path (the elipse represents the position uncertainty) while uses the laser for location and sonars for obstacles avoidance. From time to time the localization with the color camera makes the uncertainty to decrease. At the same time, new obstacles are being added to the initial map.

The results are very much impressive with both localization methods integrated, than alone.

6 Conclusions

We have shown a case of mobile robot which takes into account a lot of heterogeneous sensor information, and fuses it to achieve a control task. Several features follows: i) state estimation by fusion of odometric and camera observations; ii) reactive control for path following with obstacles avoidance by means of multisensor fusion; iii) dynamic building of environment geometric models, as well as integration of new objects; and iv) planning and coordination of continuous and discrete localization systems.

A demonstrator of the mobile robot completing a mission which tasks into account all the previous topics is available. With these results in mind, we conclude that the mathematical methods described above, appropiately combined with AI techniques, and the fuse of multiple sensor data, allows a conventional IAS to cope with the existent uncertainty in its sensors and the surounding environment.

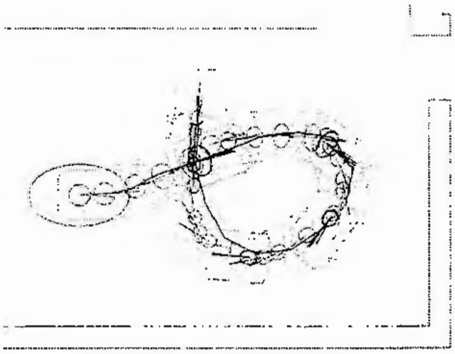


Figure 5: Sensor coordination for map building and localization

Acknowledgements. We thank Cedric Dassy and Pedro Romero for their work in preparing the demonstration. We acknowledge the funding of the Spanish government through the CICYT project TAP96-0600 EVS (*Virtual Platform for Autonomous Distributed Systems Engineering*) and of the European Community through the MOBINET project (Training and Mobility of Researchers program).

References

[1] Ayache, N. and Faugeras, O. D., Maintaining Representation of the Environment of a Mobile Robot. IEEE Trans. on Robotics and Automation, 5 (6) (1989) 804-819.

[2] Brooks, R. A., A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, 2 (1) (1986) 14-23.

[3] Crowley, J. L., Mathematical Foundations of Navigation and Perception For an Autonomous Mobile Robot. In: Reasoning with Uncertainty in Robotics, (Eds.: Dorst, L., van Lambalgen, M. and Voorbraak, F.) Springer, 1996, 9-51.

[4] Durrant-White, H. F., Integration, Coordination and Control of Multi-Sensor Robot Systems. Kluwer Academic Publishers, 1988.

[5] Elfes, A., Sonar-Based Real-World Mapping and Navigation. In: Autonomous Robot Vehicles, (Eds.: Cox, I. J. and Wilfong, G. T.) Springer Verlag, Berlin (1990).

[6] Matía, F., Moraleda, E., Jiménez, A. and Puente, E. A., Autonomous Systems and The Management of Uncertainty. In: Proc. Fuzzy Logic and the Management of Complexity, Sidney, Australia, 1996.

[7] Moreno, L., Moraleda, E., Salichs, M. A., Pimentel, J. R. and Escalera, A., Fuzzy Supervisor for Behavioral Control of Autonomous Systems. In: Proc. IECON Maui, Hawaii, USA, 1993, 258-261.

[8] Zhang, Z. and Faugeras, O., A 3D World Model Builder with a Mobile Robot. The International Journal of Robotics Research, 11 (4) (1992) 269-285.

APPLICATION OF THE MODEL REFERENCE ADAPTIVE CONTROL TO ROBOT IMPEDANCE CONTROL

Drago Matko, Roman Kamnik, Tadej Bajd
Faculty of Electrical Engineering, University of Ljubljana
Tržaška 25, Ljubljana, SLOVENIA
drago.matko@fe.uni-lj.si kamnikr@robo.fe.uni.lj.si bajd@robo.fe.uni-lj.si

Abstract. The paper deals with the application of the model reference adaptive control to robot impedance control whicb is actually a technique of steering the end-effector on a prescribed path and satisfying a prescribed dynamic relationship between the force and the end-effector position. Due to unknown parameters of the environment (stiffness, exact position) a model reference algorithm is proposed wbich differs from classical algorithms in way of excitation. The results of the proposed procedure are illustrated by the application on the ASEA IRb 6 industrial robot.

1 Introduction

Model reference adaptive control, an explicit adaptive technique, has been very attractive from the very beginning of adaptive control era. The basic idea of the method is to specify a reference model involving the desired performances of the basic loop consisting of the controlled process and a classical controller. Its main applications are in the field of the tracking control, where the plant output is supposed to track a reference trajectory. However in industrial practice, sucb as e.g. robot control, the problems are met, which do not correspond to the classical applications of model reference adaptive control. The purpose of this paper is to describe one of such applications, the adaptive control of an industrial robot with the end-effector contacting the environment.

In the basic control loop represented by *impedance controller*, the force is regulated by controlling the position and its dynamic relationship (mecbanical impedance) with the contact force. The pioneering work of impedance control was published by Hogan in [4]. In this work a second-order mass-spring-damper system is used to specify the target dynamics, however simpler models such as pure stiffness or combination of dampening and stiffness can also be used [1,11]. In this manner the basic equation of the second-order dynamic relationship between the end-effector position X and the contact force F is given by:

$$F = \boldsymbol{M}(\ddot{X}_R - \ddot{X}) + \boldsymbol{B}(\dot{X}_R - \dot{X}) + \boldsymbol{K}(X_R - X), \tag{1}$$

wbere the diagonal matrices $\boldsymbol{M}$, $\boldsymbol{B}$ and $\boldsymbol{K}$ contain the impedance parameters along Cartesian axis representing the desired inertia, damping and stiffness of the robot, respectively. The X_R is the steady state nominal equilibrium position of the end effector in the absence of any external forces. As X_R is software specified, it may during the contact with the environment reacb the positions beyond the reachable workspace or inside the environment.

On the other hand, the industrial robots are in general position controlled with independent joint controllers and kinematic software joining them into a single entity. Unavailability of an adequate robot model and, from the hardware point of view, control system with no access to direct motor current (torque) control, imply usage of force feedback to modify the reference position commands. The resulting position-based impedance control scbeme consists of an inner/outer feedback loop configuration. The inner loop represents the non-modified position robot controller, while the outer loop uses force feedback signal modifying the inputs to the position servo and in the same time satisfying the impedance dynamic equation.

Several studies were carried out analyzing the performance of position-based impedance controllers [2,7]. Volpe and Khosla [10] have made a theoretical and experimental comparison of the explicit force and impedance control methods. They showed that an impedance controller has an algebraic structure similar to the proportional gain explicit force controller with feedforward. Furthermore, this correspondence becomes exact wben the position feedback is constant what occurs when the robot is contacting stiff environment. As the industrial robot applications in most cases involve interactions with rigid objects, the introduction of a reference force signal into the impedance control scheme is possible. Thus, one of the major shortcomings of the impedance control, indirect specification of the desired contact force, can be overcome.

The paper is organized as follows: First the classical position based impedance control is introduced, in section 2 where also the necessity of adaptive control is stressed. In Section 3 the adaptive controller is presented and the differential equation describing the basic loop of the adaptive system is rewritten in the form suitable for the model reference adaptive control. This equation differs from the classical model reference approach in the fact that there no reference signal is present. The only excitation of the adaptive system is the initial condition due to zero force prior to impact. The implementation of the adaptive control system on the ASEA IRb 6 robot is presented in Section 4.

2 The basic loop - position-based impedance control

Controlling the robot mechanical impedance by generating the reference position trajectory of the existing positional controller has been recognized as a practical approach to the industrial robot impedance control. When applying the contact force, the reference position is to be chosen inside the environment and cannot be reached by the robot because of the geometrical constraints. The positional controller responds to the resulting error in position with additional actuator current and exerts force to the environment.

The class of position-based impedance controllers is usually focused on the most frequent robot contact applications, interactions with rigid objects. When the robot is contacting a stiff environment, the terms $\ddot{X}_R$ and $\dot{X}_R$ of the impedance equation (1) become approximately zero and, thus, the impedance equation is reduced to

$$-F = M\ddot{X} + B\dot{X} + K(X - X_R)\,. \tag{2}$$

In this case the term KX_R acts simply as a scaled reference position and can be directly replaced by the reference force signal [10]. Moreover, following the idea of Seraji and Colbaugh [8], who were using the reference force input for specifying the desired contact force and the reference position input for improving the controller performance, the vector of the desired position X_c (reference position for the position controller of the robot) is written as an output of a system called *impedance filter*. It is a linear second-order system with the transfer function (3) defining the dynamic relationship between the force error and the position:

$$E = F_R - F = M\ddot{X}_C + B\dot{X}_C + K(X_C - X_R)\,, \tag{3}$$

where E is the error between the reference force F_R and the measured contact force F. The parameters of the diagonal matrices $\boldsymbol{M}$, $\boldsymbol{B}$ and $\boldsymbol{K}$ define the desired robot inertia, damping and stiffness along the particular Cartesian axis.

In order to determine the behavior of the impedance controller when in contact, the environment is modelled by a linear spring with the stiffness K_E. The measured contact force can be then calculated as:

$$F = K_E(X - X_E)\,, \tag{4}$$

where X_E is the location of the environment. When deriving the control laws we assume that the robot is equipped with an ideal position control system ensuring the commanded position X_C to be reached with negligible dynamics ($X \approx X_C$). This assumption allows to consider variables of the vectors X and F independently. The elements of vectors X and F are then denoted as scalars by the lower case letters x and f. The force tracking error e is obtained as:

$$e = f_R - f = f_R - k_E(x - x_E)\,. \tag{5}$$

Assumption $x = x_C$ in equation (3) together with the equation (5) yields the equation of the force error dynamics:

$$m\ddot{e} + b\dot{e} + (k + k_E)\,e = k(f_R + k_E x_E) - k k_E x_R\,. \tag{6}$$

It can be noted that with the input signal x_R it is possible to control the force error trajectory. In the steady-state, when x_R is constant, the Laplace transform of the equation (6) defines the steady-state force tracking error:

$$e_{ss} = \lim_{s\to 0} s\,e(s) = \frac{k}{k + k_E}\,[(f_R + k_E x_E) - k_E x_R]\,. \tag{7}$$

The steady-state force tracking error will equal zero when the following reference position trajectory will be chosen:

$$x_R = \frac{f_R}{k_E} + x_E \,. \tag{8}$$

Unfortunately, in practical applications the environmental parameters, stiffness k_E and location x_E, are almost never precisely known and may change considerably during the task. For improving the force tracking characteristics an adaptive control approach should be, therefore, employed.

3 The model reference controller

In this section the adaptive control algorithm will be introduced in order to cope with the problems of unknown environmental parameters. The basic loop controller is given by:

$$x_R = f(t) + k_p(t)\, e + k_d(t)\, \dot{e} \,, \tag{9}$$

where $k_p(t)$ and $k_d(t)$ are proportional and derivative feedback gains acting on the force error $e(t)$ and the error rate $\dot{e}(t)$. Signal $f(t)$ is an auxiliary signal which compensates the steady state error. All three adjustable parameters are generated by the model reference adaptive algorithm.

Next the basic equation suitable for the application of the model reference controller will be derived. Substituting x_R in the equation of error dynamics (6) by the equation (9) yields the equation of the complete adjustable system in the frame of the model-reference adaptive control (*MRAC*):

$$\ddot{e} + \left(\frac{b + k k_E k_d(t)}{m}\right)\dot{e} + \left(\frac{k + k_E + k k_E k_p(t)}{m}\right)e = \frac{k(f_R + k_E x_E - k_E f(t))}{m} \,. \tag{10}$$

The parameters $f(t)$, $k_p(t)$ and $k_d(t)$ of the adjustable system are varied in order to minimize the difference between the actual force error $e(t)$ and the desired force error $e_m(t)$. The desired force error trajectory $e_m(t)$ is determined by the output of the reference model. The reference model is designed as a second-order linear system:

$$\ddot{e}_m + 2\zeta\omega\,\dot{e}_m + \omega^2 e_m = 0 \,, \tag{11}$$

with the output representing a response to the initial conditions. The user specified parameters ζ and ω determine the profile of the reference model output trajectory and represent the undamped natural frequency and the damping ratio, respectively.

The essential difference of the described approach to the classical approach is in the existence of the reference signal. While in classical model reference control the reference signal excites the basic loop and the reference model, there is no such signal in the described application, rather both subsystems of the adaptive control are exited by the initial conditions. When in initial contact with the environment, the force error is supposed to decay to zero by dynamics defined by the reference model. The initial condition for the reference force error e_m has the value of the reference force f_R, while the initial condition for it's rate $\dot{e}_m$ is zero. The reference model output holds initial values up to the moment of impact when it's dynamics is started according to equation (11).

It should be stressed that the reference signal is absent only in the reformulation of the problem, i.e. in the equations (10, 11) describing the model reference control system, not however in the control problem where the manipulator end-effector is supposed to follow a prescribed path while tracking a prescribed force.

The theory of the direct model reference adaptive control is providing the adaptation laws for the variable system parameters. The adaptation ensures the tendency of the system (10) response to approach to the response of the reference model (11). The adaptation laws are derived following the Lyapunov approach to the nonlinear system control [5].

Let us rewrite the equation of the adjustable system (10) in a more compact form:

$$\ddot{e} + \alpha_1 \dot{e} + \alpha_2 e = \beta_0 \,, \tag{12}$$

and define the differences between particular parameters of the above equation and the equation (11):

$$\delta_{b_0} = \beta_0 \,, \quad \delta_{a_1} = \alpha_1 - 2\zeta\omega \,, \quad \delta_{a_2} = \alpha_2 - \omega^2 \,. \tag{13}$$

By subtracting equations (12) and (11) we obtain:

$$(\ddot{e} - \ddot{e}_m) + \alpha_1 \dot{e} - 2\zeta\omega\,\dot{e}_m + \alpha_2 e - \omega^2 e_m = \beta_0 \,, \tag{14}$$

which can be transformed by the definition of the error between the actual and desired force error ($\epsilon = e - e_m$) into the form:

$$\ddot{\epsilon} = -2\zeta\omega\dot{\epsilon} - \omega^2\epsilon - \delta_{a_1}\dot{e} - \delta_{a_2}e + \delta_{b_0}. \tag{15}$$

Let us now propose a scalar Lyapunov funciion V:

$$V = (\omega^2 + 2w_p\zeta\omega - w_p{}^2)\epsilon^2 + q^2 + \frac{1}{\gamma_0}[\delta_{b_0} + \gamma_0' q]^2 + \frac{1}{\gamma_1}[-\delta_{a_1} + \gamma_1' q\dot{e}]^2 + \frac{1}{\gamma_2}[-\delta_{a_2} + \gamma_2' q e]^2, \tag{16}$$

where q defines a linear combination of ϵ and $\dot{\epsilon}$ ($q = w_p\epsilon + \dot{\epsilon}$), while the parameters γ_i and γ_i' are positive constants. The Proposed Lyapunov function is positive definite when the parameter w_p is chosen to satisfy the inequality:

$$0 \leq w_p < 2\zeta\omega. \tag{17}$$

The time derivative of V, after applying (15) and after addition and substraction of terms $2\gamma_0'(q)^2$, $2\gamma_1'(q\dot{e})^2$, $2\gamma_2'(qe)^2$, gets the form of:

$$\begin{aligned}\dot{V} = {} & 2\dot{\epsilon}^2(w_p - 2\zeta\omega) - 2w_p\omega^2\epsilon^2 - 2\gamma_0' q^2 - 2\gamma_1'(q\dot{e})^2 - 2\gamma_2'(qe)^2 + \\ & + 2(\delta_{b_0} + \gamma_0' q)\{q + \frac{1}{\gamma_0}(\dot{\delta}_{b_0} + \gamma_0' \tfrac{d}{dt}(q))\} + \\ & + 2(-\delta_{a_1} + \gamma_1' q\dot{e})\{q\dot{e} + \frac{1}{\gamma_1}(-\dot{\delta}_{a_1} + \gamma_1' \tfrac{d}{dt}(q\dot{e}))\} + \\ & + 2(-\delta_{a_2} + \gamma_2' qe)\{qe + \frac{1}{\gamma_2}(-\dot{\delta}_{a_2} + \gamma_2' \tfrac{d}{dt}(qe))\}.\end{aligned} \tag{18}$$

Now, according to the Lyapunov theory, for the response error ϵ to vanish asymptotically, $\dot{V}$ must be negative-(semi)definite. For this purpose, we set the terms { } equal to zero, while the parameter w_p is chosen according to inequality (17). Integrating the resulting equations yields the adaptive controller terms:

$$f(t) = f(0) - a_1\int_0^t q\,dt - a_2 q, \qquad k_p(t) = k_p(0) + b_1\int_0^t qe\,dt + b_2 qe, \tag{19}$$

$$k_d(t) = k_d(0) + c_1\int_0^t q\dot{e}\,dt + c_2 q\dot{e}, \qquad q = w_p(e - e_m) + (\dot{e} - \dot{e}_m). \tag{20}$$

that define the mechanism for the system adaptation. In the Figure 1 the control scheme of the impedance controller including the adaptive algorithm for the reference trajectory generation is depicted. The system adaptation is based on simple expressions (19-20) that can be on-line computed in real-time.

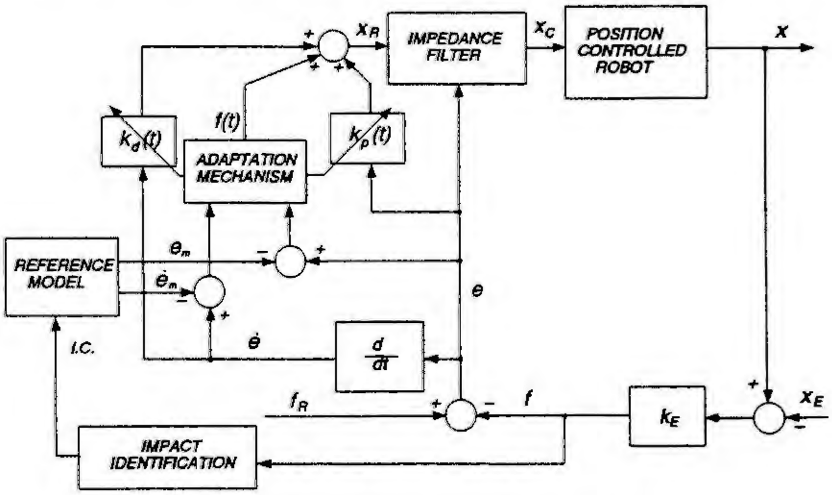


Figure 1: Adaptive impedance control scheme

In the adaptation laws, the parameter w_p represents positive weighting factor, while the parameters a_i, b_i and c_i are small positive proportional and integral adaptation gains, and $f(0)$, $k_p(0)$, $k_d(0)$ represent initial values of the adaptive parameters. While the initial proportional and integral gains are selected to be zero, the initial value $f(0)$ defines the position trajectory of the robot free-space motion. Thus, before the contact occurred the trajectory x_R had the same value as the signal $f(0)$, while after the contact the parameter $f(0)$ holds the value assessed at the instant of contact.

4 Implementation of the model reference controller

The model reference adaptive impedance control algorithm was primarily tested by the simulation in *Matlab-Simulink^{TM}*. The control scheme was realized on ideal position control of the robot i.e. under ideal circumstances corresponding to the theory described in Section 3. Thereafter an identified model of the position control subsystem of the ASEA IRb 6 robot was introduced in order to test the robustness of the algorithm. The simulation results, extensively presented in [6], proved that the system is capable to track the desired contact force trajectory determined by the reference model output and to alter the robot arm dynamic characteristics regarding the impedance parameters *m*, *b* and *k*.

Furthermore, the presented adaptive impedance controller was implemented for a real industrial ASEA IRb 6 robot. The ASEA robot is a 5 DOF robot driven by DC motors and gear transmissions. As the original ASEA controller performs only position control in the joint coordinates, a 486/66 PC computer was added providing the computational platform for both, the positional control algorithm in the world coordinates and the impedance control algorithm. Such controller enhancement would not be required for a modern industrial robot controller enabling velocity control in Cartesian coordinates.

The contact force is measured by the JR^3 four-axis force/torque wrist sensor mounted at the robot end point. Each of the sensor voltage outputs is digitized by the A/D converter and after filtering an estimate of the force error rate is obtained by digital differentiation. The impedance control algorithm calculates the commanded position trajectory defined in the Cartesian space on the basis of the force feedback and by taking into account the adaptive algorithm. Using the inverse Jacobian matrix and the joints-to-motors transformations, the reference value is transformed into the motor velocities which are at the end of the sample cycle transmitted to the servo systems through D/A converters at the sample rate of 120 *Hz*. The performance plots of the described algorithm are given in Figure 2.

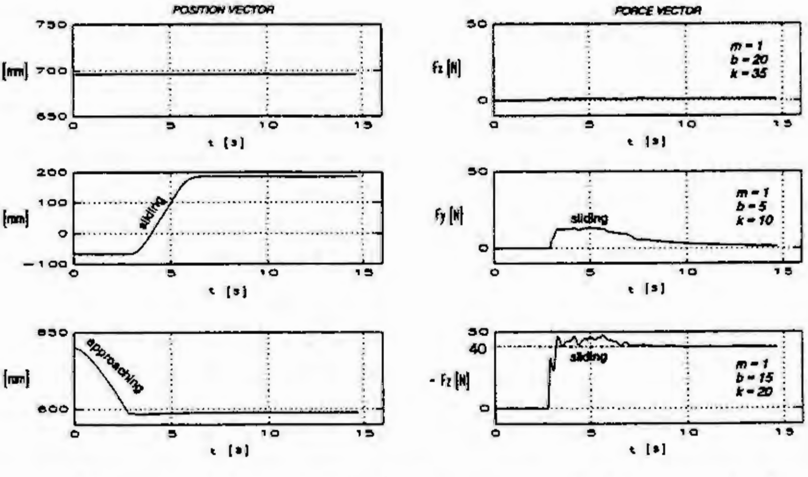


Figure 2: Position and force profiles during impact and sliding over horizontal surface

The impedance control scheme of the industrial manipulator consists of three separated adaptive controllers described in the previous sections. Each controller independently performs the impedance control along a single axis of the Cartesian coordinate system. To test the effectiveness of the impedance controller, a simple task was chosen: the robot end-effector was first approaching the constraining surface (horizontal wooden table) along a straight line in the direction normal to the surface and after the impact compliant motion in the positive *y* direction was performed along the surface, while exerting a contact force of -40 *N*. Robot was approaching the surface with the velocity 20 *mm/s* and was moving with 100 *mm/s* when sliding over the surface. The tests were performed by the following adjustments of the controller gains: $a_1 = 0.11$, $a_2 = 0.011$, $b_1 = 2\ 10^{-5}$, $b_2 = 5\ 10^{-5}$, $c_1 = 10^{-8}$, $c_2 = 10^{-8}$ and $w_p = 5$. In the left side of the figure the actual robot positions along the particular axis are presented, while the right side describes the contact forces.

It can be noted that after impact a stable contact is achieved ensuring the desired contact force in the z direction. The presence of the force in the y direction can be also observed being the consequence of the Coulomb friction due to sliding.

5 Conclusions

In the paper the application of the model reference adaptive control to the impedance control of a robot end-effector contacting the rigid environment was presented. The presented control approach employs the original industrial manipulator position control system with no demands for controller hardware reconstruction. The added force control algorithm makes use of adaptive terms for the system adaptation to unknown environmental and robot dynamic parameters. The basic difference between the classical model reference control and the given algorithm is the way of adaptive system excitation. The reference signal used in classical MRAC was replaced by the nonzero initial condition of the system.

The experimental tests on the ASEA IRb 6 robot showed that the system remains stable throughout all phases of the task, constrained and unconstrained, regardless of the high environment stiffness. Furthermore, the robot is able to exert the desired forces at the end-effector and simultaneously achieve the desired end effector impedance characteristics.

References

1. Asada, H. and Slotine, J.J.E., Robot Analysis and Control. John Wiley and Sons, New York, 1986.
2. Carignan, C.R., Manipulator Impedance Accuracy in Position-Based Impedance Control Implementations.In: Proceedings 1994 IEEE International Conference on Robotics and Automation, San Diego, 1994, 2, 1216-1221.
3. Colbaugh, R., Seraji, H. and Glass, K., Adaptive Compliant Motion Control for Dexterous Manipulators. The International Journal of Robotics Research, 14 (1995), 270-280.
4. Hogan, N., Impedance Control: An Approach to Manipulation, Parts I-III, ASME Journal of Dynamic Systems, Measurement and Control, 107 (1985), 1-24.
5. Isermann, R., Lachmann, K.H. and Matko, D., Adaptive control systems. Prentice Hall, New York, 1992.
6. Kamnik R., Matko, D. and Bajd, T., Adaptive Impedance Control of an Industrial ASEA IRb 6 Robot. In: Proceedings of 26th International Symposium on Industrial Robots, Singapore, 1995, 31-36.
7. Palletier, M. and Doyon, M., On the Implementation and Performance of Impedance Control on Position Controlled Robots. In: Proceedings 1994 IEEE International Conference on Robotics and Automation, San Diego, 1994, 2, 1228-1233.
8. Seraji, H. and Colbaugh, R., Force tracking in impedance control. In: Proceedings 1993 IEEE International Conference on Robotics and Automation, Atlanta, 1993, 2, 499-506.
9. Singh, S.K., Adaptive Control of Manipulator Interaction With Environment: Theory and Experiments. In: Proceedings 1993 IEEE International Conference on Robotics and Automation, Atlanta, 1993, 3, 1001-1006.
10. Volpe, R. and Khosla, P., The Equivalence of Second-Order Impedance Control and Proportional Gain Explicit Force Control. The International Journal of Robotics Research, 14 (1995), 574-589.
11. Whitney, D.E., Historical Perspective and State of the Art in Robot Force Control. The International Journal of Robotics Research, 6 (1987), 3-14.

ON THE APPLICABILITY OF NEURAL-NET AND FUZZY MODELS IN MODEL BASED CONTROL

D. Matko[1] K. Kavšek-Biasizzo[1], S. Milanič[1], R. Karba[1], Oliver Hecker[2]
[1] Faculty of Electrical Engineering University of Ljubljana, Slovenia
[2] Institute of Automatic Control Technical University of Darmstadt, Germany

Abstract. In the paper the applicability of neural-net and fuzzy models in model based control is critically reviewed. The models are tested on an industrial scale thermal plant. The results of the closed loop control with the Dynamic matrix controller with the matrix of impulse response recalculated from the fuzzy model illustrate the applicability of nonlinear models in model based control.

1 Introduction

Models are inherently involved in designing controllers. The model based controllers involve the process model in an explicit form. Classical model based controllers have used linear models, while recent works in this field employ some modern modelling approaches such as neural nets and fuzzy techniques, which are able to cope with nonlinearities.

Neural nets and fuzzy logic will be treated here as a mathematical tool to model nonlinear systems. Fuzzy logic provides a mathematical tool to formulate mental models in a compact mathematical form. Intuitive and heuristic nature of human mind which is actually imprecise can be incorporated in formal models which can essentially support the planning and decision making processes. Fuzzy logic has extended the classical mathematical models in the form of differential and difference equations to a broad class of models which are easy understandable.

It has been shown that fuzzy logic and neural nets are closely related, actually it has been shown that a class of fuzzy logic systems, the Learning of Fuzzy Rules (LFR) from numerical data is equal to the training of Radial basis Function (RFB) [4]. Although both approaches originate in two different forms of "intelligent" control, they can be treated as universal approximators (UA) which can approximate continuous functions to an arbitrary precision [3, 2] .

A common approach to dynamic fuzzy logic/neural net models is to use time shifted signals what results in discrete-time models. The usage of derivatives, integrals or other transfer function would results in continuous-time models. Also mixed-continuous-discrete time models are possible. In this presentation the discrete time models will be treated and tapped-line will be used to generate the time - shifted signals. If only (tapped) input signal of the model is used as input of the UA, the resulting model is nonrecursive and has finite impulse response. If (tapped) input and (tapped) output of the model are used as input of the UA the resulting model is recursive and may have an infinite impulse response.

The paper is organised as follows: The identification procedures for fuzzy and neural net models are described in Sections 2 and 3 respectively. The description of the device on which the experiments (identification and closed loop control) were performed is given next. The results of are comparatively given in Section 4. The models are used in a version of the Dynamic Matrix Controller (DMC) with the matrix of impulse response recalculated from the fuzzy model whenever the working point changes.

2 Identification by Fuzzy Models

In the recent years many different approaches for fuzzy identification have been proposed in the literature, by Sugeno [7], Pedrycz and Czogala [6], etc. In this paper the 0 order Sugeno-Takagi fuzzy model will be used. Suppose the rule base of a fuzzy system is as follows:

$$\mathbf{R}_i : \textbf{IF } x_1 \textit{ is } \mathbf{A}_i \textit{ and } x_2 \textit{ is } \mathbf{B}_i \textbf{ THEN } y = r_i \qquad i = 1, \ldots N \tag{1}$$

where x_1 and x_2 are input variables of the process, y is an output variable, A_i, B_i are fuzzy sets characterised by their membership functions and r_i are the crisp values. Such a very simplified fuzzy model can be regarded as a collection of several linear models applied locally in the fuzzy regions, defined by the rule premises. The idea behind this kind of modelling is close to well-known concept of gain scheduling.

Rule-premises are formulated as fuzzy AND relations on the Cartesian product set $X = X_1 \times X_2$, and several rules are connected by logical OR. Fuzzification of a crisp value x_1 produces a column vector

$$\mu = [\mu_{1A_1}, \mu_{1A_2}, \ldots, \mu_{1A_m}]^T \tag{2}$$

and similarly for a crisp value x_2. The degrees of fulfilment of all possible AND combinations of rule premises are calculated and written into matrix S. If the algebraic product is used as AND operator, this matrix can be directly obtained by multiplication:

$$\mathrm{S} = \mu_1 \otimes \mu_2 = \mu_1 \cdot \mu_2^T. \tag{3}$$

A crisp output value y is computed by simplified algorithm for singletons as a weighted mean value (Center of Singeltons):

$$y = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} r_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij}}. \tag{4}$$

The dimension of matrix $\mathrm{S}(m \times n)$, which actually represents the structure of the model, depends on the dimensions of input fuzzy sets $\mu_1(m \times 1)$ and $\mu_2(n \times 1)$. The fuzzy relational matrix $\mathbf{R}$ consists of elements r_{ij} which are estimated from the measurements, using standard least-squares algorithm.

In the case of more than two input variables (MISO multi-input-single-output fuzzy system), matrices S and R are no longer matrices, but both become multidimensional structures, defined in the total product space of the inputs.

The described fuzzy model actually represents a static nonlinear mapping between input fuzzy sets and output fuzzy sets, so dynamic systems can be modelled by feeding back the lagged input and output signals. In the same way the system dynamics is captured in other kinds of nonlinear models like neural network models (see the next section) or nonlinear regression models, for instance NARX model:

$$y(k+1) = f(y(k), y(k-1), \ldots, u(k), u(k-1), \ldots) \tag{5}$$

where $y(k), y(k-1), \ldots$ and $u(k), u(k-1), \ldots$ denote the lagged model output and input signals, respectively.

3 Identification by Neural Net Models

Another potential candidate for black-box nonlinear modelling are neural network models. Since, similar to fuzzy models, most of the neural networks represent a static mapping between input and output variables, the dynamics can be modelled in the way described in the Section 1. The overall nonlinear mapping depends on the network's parameters (weights and other parameters) which are optimised during the training phase in order to minimise the difference between $y(k+1)$ and $\hat{y}(k+1)$, i.e. between the measured and predicted values respectively.

Neural network models are in the last years frequently used for identification of nonlinear processes. To use them in MBPC, we tried a lot of different neural network structures and training procedures. The detailed comparative study is published elsewhere [5]. The most suitable structure seems to be multilayer network with only one hidden layer with the following training algorithms:

- The **RAWN** (Random Activation Weights Neural Network) proposed by H. te Braake and van Straten [1]. The RAWN training algorithm is a non-iterative procedure that fixes the weights in the hidden layer to random values. This results in linear-in-the-parameters estimation problem for the weights in the output layer, which can be computed by a least squares algorithm. This training procedure takes up just a tiny amount of training time in comparison to backpropagation. Resulting neural net yields excellent approximation results and good generalisation properties.

- A novel learning algorithm and the resulting OMN (Ontogenetic Least Squares Multilayer Network) structure described in [5]. All the weights are computed by a least squares algorithm. Similar to other ontogenetic approaches the OMN starts with only one neuron in the hidden layer, which is trained to mimic the network desired output. After training, the output error signal between the desired and network output is evaluated. In the next training step another neuron is added into the hidden layer. This neuron is trained to mimic the output error signal from the previous step and all the weights adherent to the newly added neuron and all the output weights are recomputed. If the output RMSE (Root Mean Squared Error) is smaller than the one a priori prescribed, the training stops, otherwise a new neuron is added into the hidden layer and the whole procedure is repeated.

  The main contribution of the OMN algorithm is that it gives acceptable accuracy with only few neurons in the hidden layer. That number is considerably smaller than the resulting number of other multilayer networks adherent training rules. This fact is of extreme importance when the network has to be incorporated in real-time predictive control, because it requires a small computational effort needed for prediction.

4 Application on a thermal plant

The plant under consideration was an industrial thermal plant, located at the Institute of Automatic Control, Technical University of Darmstadt, Germany. The heart of the thermal plant is a tubular industrial scale heat exchanger, through which steam from a steam generator continuously circulates in a counter-current flow to water circuit. A schematic diagram is shown in Fig. 1. Temperature of the saturated steam is kept constant by a local pressure control in the steam generator and the flow of the steam is controlled by position of the electrically driven steam valve (G11). After heating in the exchanger (to the temperature T31), the water passes through a pneumatic valve into the air cooler and then reenters the exchanger.

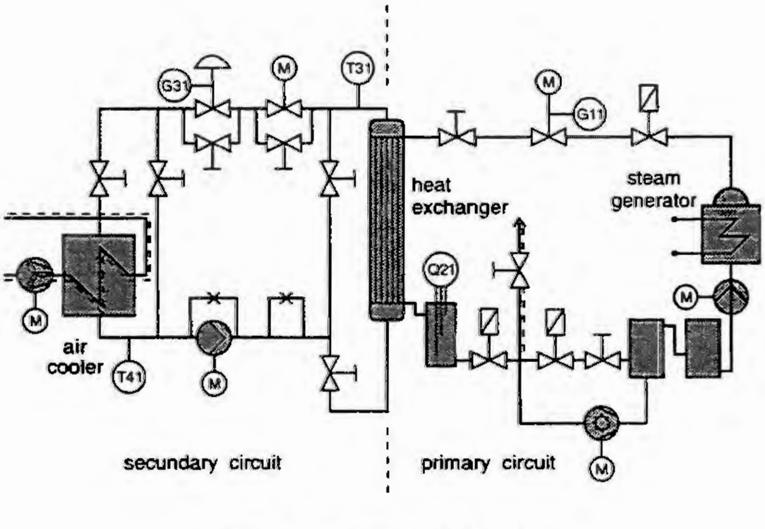


Figure 1: Thermal plant scheme

The behaviour of the plant strongly depends on operating conditions, defined by the operating point (G11, T31) and by other variables of the plant:

- temperature at the outlet of the air cooler (T41)
- pneumatic valve position (G31)

The model building: The Identification

For the thermal plant different MISO nonlinear models were identified. The aim was to build and select the simplest and the smallest nonlinear models, which were still accurate enough and valid for the wide range of operating conditions.

Both, fuzzy and neural models approximate a first-order nonlinear regression model where the new estimation of temperature T31 (denoted by $\hat{y}$) is a function of the current temperature value T31 (denoted by y), the current steam valve position G11 (denoted by u_1), the current temperature value T41 (denoted by u_2) and the current position of the pneumatic valve G31 (denoted by u_3):

$$\hat{y}(k+1) = f(y(k), u_1(k), u_2(k), u_3(k)) \tag{6}$$

Signals used for the identification are shown in Fig. 2

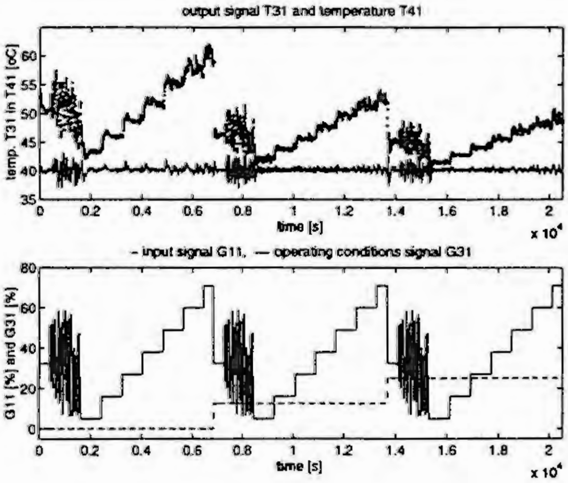


Figure 2: Signals for the identification

The verification

The models were validated by recursive simulation using another set of measurements.

Fuzzy Model: For the identification of the fuzzy model only tree equally spaced triangular shaped fuzzy sets in each fuzzy input space were chosen. In general, the estimation of the membership functions (shape, number, position) and determination of the rules can be done by several methods (genetic algorithms, neural network, clustering,..). The left part of Fig. 3 shows the measured values and the predicted values of the resulting fuzzy model with 81 identified parameters.

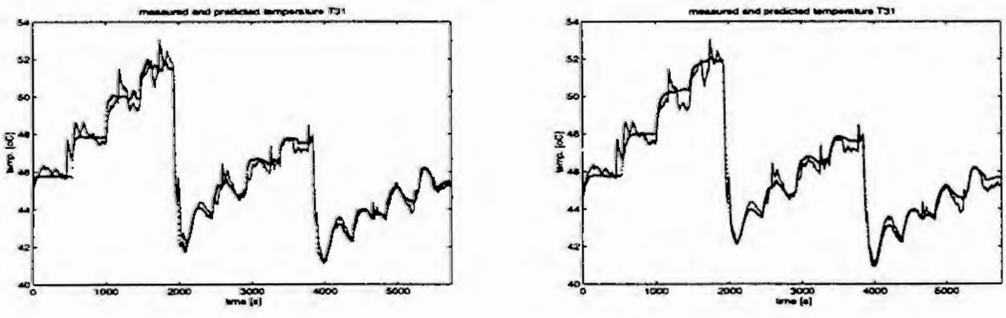


Figure 3: The validation by fuzzy (left) and RAWN (right) models.

Neural Network Models: Among different neural network models, two of them were chosen for the MBPC: RAWN and OMN Multilayer networks. Both of them yield good accuracy with small number of neurons in the hidden layer: RAWN with 18 neurons (and 19 parameters to be trained) and OMN with only 7 neurons (and 37 parameters to be identified) with almost the same accuracy. The comparison between measured output of the plant and predicted output of the RAWN neural model is shown in the right part of Fig. 3. Predicted output of the OMN neural net is almost the same as RAWN output.

Application of models in predictive control

Model Based Predictive Control techniques are based on the prediction of the process output which is obtained explicitly or implicitly according to the model of the process to be controlled. A predictive controller calculates a sequence of future control signals over a certain control horizon and calculates the control variable which brings the predicted output as close as possible to a reference trajectory. This control sequence is in general obtained by optimising a certain objective function which describes the control goals.

Here the focus will be on the Dynamic Matrix Control (DMC) technique which is based on nonparametric step response model of the process. If there are no constraints on the process input and output variables, an analytical solution of the optimisation problem exists and provides a sequence of future control increments. At current time only one (the first one) control value is applied to the process and in the next sampling time the solution is computed again according to a receding horizon strategy.

All of the conventional MBPC techniques use linear prediction models. However, many industrial processes are inherently nonlinear and a linear model may be acceptable only in a narrow range around the operating point. Our idea is to combine the theory of predictive control with fuzzy and neural network models, which could be easily identified from process input-output measurements.

Presented fuzzy and neural network models of the plant were used in the DMC control scheme to calculate the step response vector g of the model. Under the assumption that the gain and the time constants of the plant do not change in the region around one operating point the new vector g is computed only at the set-point changes and at changes of the operating conditions signal G31.

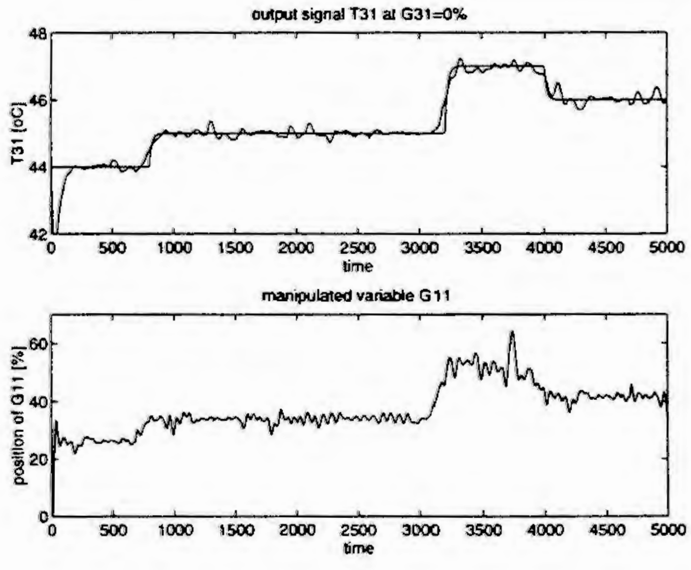


Figure 4: Real-time predictive control at G31=0%

Fig. 4 and 5 show the reference trajectory, output signals and control signals for two different operating conditions determined by the position of the pneumatic valve G31 (0% and 12.5% opened). Only the results of incorporating fuzzy model are presented. The predictive control based on both neural network models performs comparably.

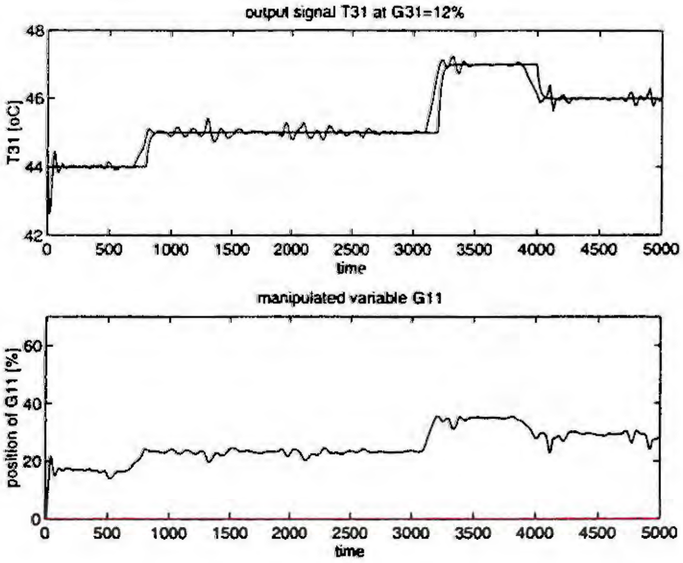


Figure 5: Real-time predictive control at G31=12%

Oscillations around the operating point are due to the oscillation of the temperature T41 and due to other disturbances. Although the gain of the process is varying for 100%, a simplified fuzzy model (or neural network models) incorporated in the DMC strategy already yield satisfactorily results. As shown in Fig. 4 and Fig. 5 the predictive control is capable of tracking different set point changes and other changes due to operating regimes.

5 Conclusion

A thermal plant was identified by fuzzy and neural net models. Both resulting models can be used in a model based controller - the DMC. The results of the closed loop control demonstrate the applicability of described models in real time control.

References

[1] te Braake H. and G. van Straten, Random Activation Weight Neural Net (RAWN) for Fast Non-Iterative Training, *Engng. Applic. Artif. Intell.*, Vol. 8, No. 1, 1995, pp. 71-80.

[2] Castro, J., Fuzzy Logic Controllers are Universal Approximators, IEEE Tr. on Systems, Man and Cybernetics, 1995, pp. 629 - 35.

[3] Girosi, F. and T. Poggio, Networks and the Best Approximation Property, C.B.I.P.Paper No. 45, MIT 1994.

[4] Kecman, V., and B. M. Pfeiffer, Learning Fuzzy Rules Equals to Radial Basis Function Neural Network Training, IEEE World Congress on Computational Intelligence, Orlando, Florida, June 1994.

[5] Milanič S., O. Hecker and R. Karba, A Comparative Study of Neural Network Models for MBPC of a Thermal Plant, Proceedings of CESA Conference, Lille, France, Vol.2, pp. 1244-49, 1996.

[6] Pedrycz W., An Identification Algorithm in Fuzzy Relational Systems, Fuzzy Sets and Systems, Vol. 15, 1984, pp. 153-167.

[7] Takagi, T. and M. Sugeno, Fuzzy Identification of Systems and its Application to Modelling and Control, IEEE Trans. on Systems, Man and Cybernetics, Vol. 15, No. 1, 1985, pp.116-132.

PSEUDOLINK-BASED CONTROL OF A FLEXIBLE MANIPULATOR

Fernández, G. [1]; Grieco, J.C. [1]; Armada, M. [2]; González de Santos, P. [2]
[1] Instituto de Ingeniería
Electrical Engineering Centre. Apdo 40200
Caracas 1040-A Venezuela
[2] Instituto de Automática Industrial. CSIC
Carretera de Campo Real, Km. 1. La Poveda, Arganda del Rey.
Madrid, Spain

Abstract. A pseudolink-based approach for a flexible one-link manipulator is presented. Modelling of the flexible link is performed using finite elements approach. The experimental set-up consists of a very flexible link attached to the motor hub moving on a plane parallel to the floor. Measurements for the tip are obtained by means of infrared emitter and acquired by using infrared cameras, that focus on sensor located at the end of the link. Hub angle is properly measured with motor encoder. Using the measurements a pseudo-link is virtually conformed, joining the tip and the motor hub. The proposed control strategy is based on tracking of the joint angle of the pseudolink to the measured hub angle. An error signal is generated with the pseudolink angle and with the hub angle and input to a PD controller; this control signal is added to the PD control signal that is coming from the encoder loop and finally this combination is delivered to the power stage of the motor controller. This strategy is evaluated and compared with other control strategies previously designed by the authors. Experimental results are presented. Achieved results are very promising. The proposed control scheme is being extended for the 2D case.

Introduction

In this paper a new approach for control of a very flexible beam is presented. In order to acquire data for tip position feedback, the system was properly adapted; using an infrared led and two infrared cameras (Selspot System by SELCOM [1]) the tip position is properly measured. This data is obtained in reference to an inertial system and converted to reference axis on the motor hub. The details of the experimental set-up can be seen at Fig. 1.

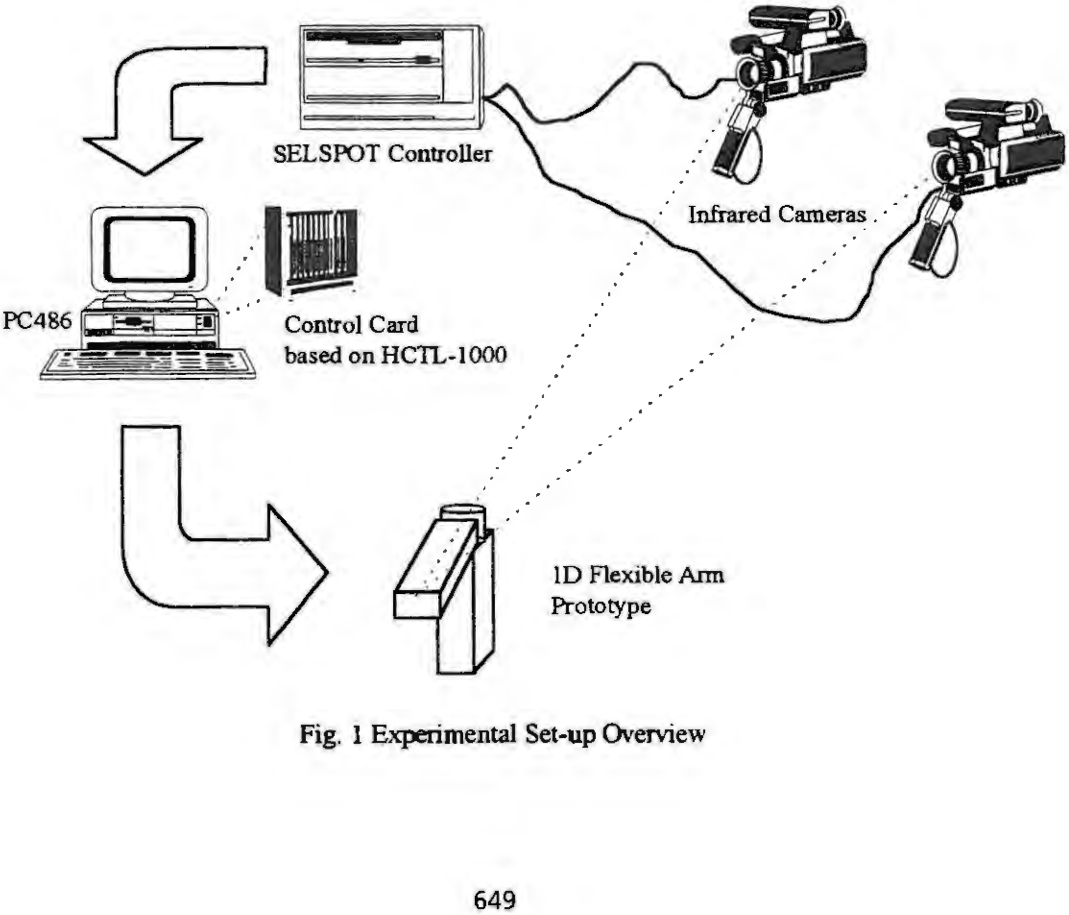


Fig. 1 Experimental Set-up Overview

In order to convert the measured tip position to the new reference system, a calibration routine must be performed before the control experiments. With the procedure developed [4], a Rotation Matrix is obtained to get on-line the tip coordinates on the plane movement. The conversion coordinates matrix employed was:

$$\mathbf{R}_0^1=\begin{bmatrix} 0.9743 & -0.0178 & -0.2246 & 108.8550 \\ 0.2250 & 0.1321 & 0.9654 & -138.8416 \\ 0.0129 & -0.9911 & 0.1326 & -940.5611 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

The resolution obtained with this experimental set-up is approximately of 0.25 mm. The control program implemented was incorporated in the software pipe provided by SELCOM, using the time control and in/out buffering synchronisation developed by the system makers (see Fig. 2).

Motor control was performed using an HCTL-1000/1100 based interface. The control and power stages were developed at *Instituto de Automática Industrial* (Madrid, Spain). The functional description blocks can be consulted in [4].

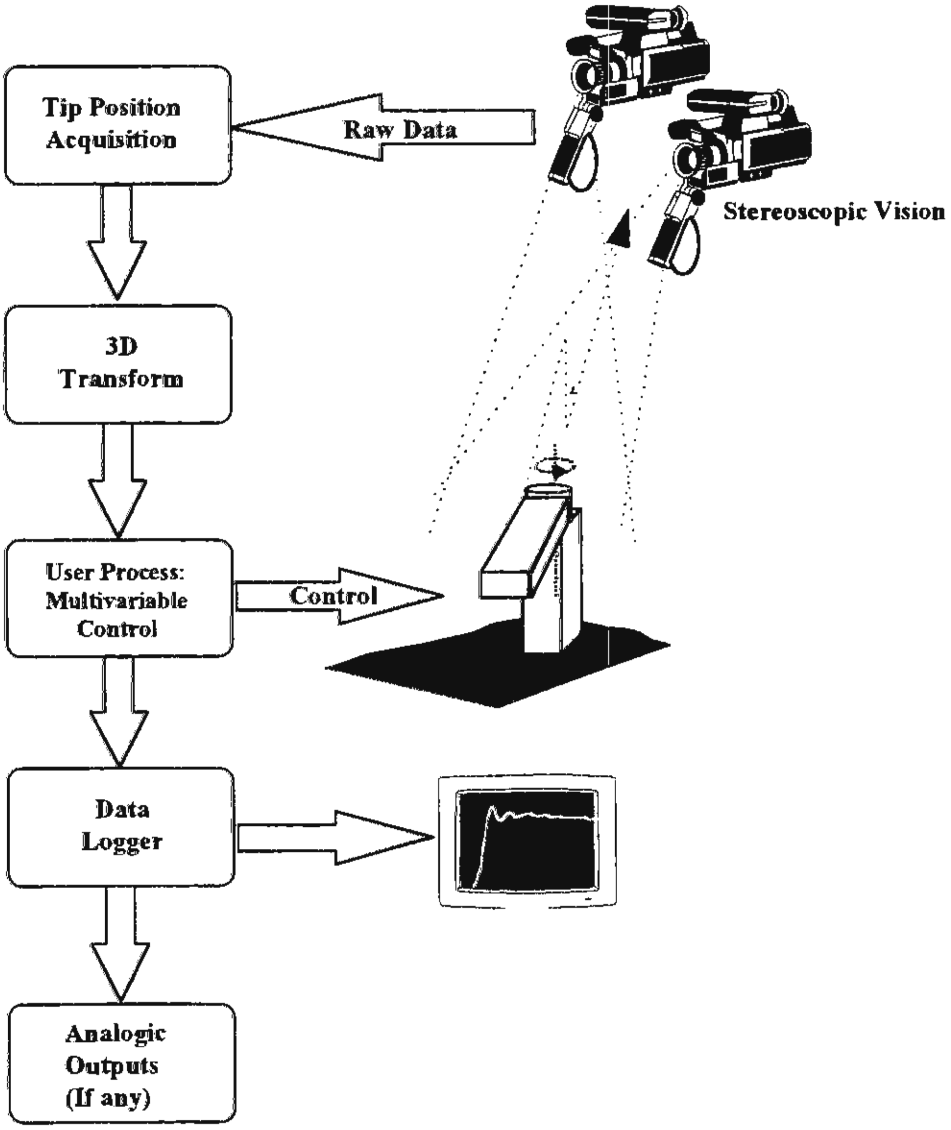


Fig. 2. Data Flow for Tip Position Control

Beam characteristics

A very flexible steel beam was employed during experiments. It was determined the Young coefficient of the material using the logarithm decrement measurement [6], obtaining an estimate close to theoretical value of the coefficient; the measured coefficient was E=2.115x10^{11} N/m^2. The inertial moment of the transverse section was of I=1.43x10^{-12} m^4. The physical dimensions are shown in Fig. 3. The natural frequency of the oscillations for this beam is 8.1348 rad/sec.

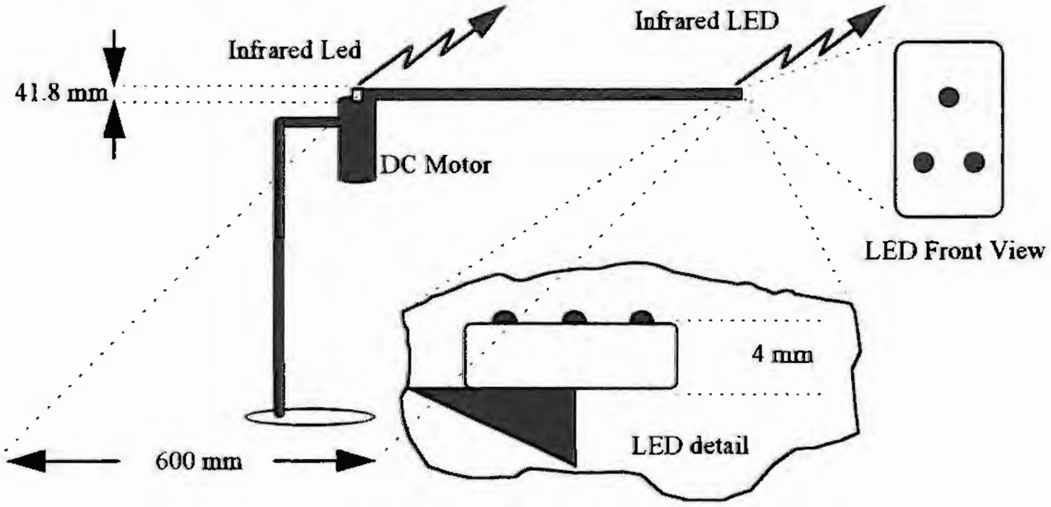


Fig. 3. Beam Physical Dimensions

Vibrations Modelling

Beam Vibrations were modelled using the Finite Element approach. Following a development presented by López-Linares [7], it is defined a *pseudolink,* and some variables related with the pseudolink rigid movement, and the beam deformation as can be observed in Fig. 4.

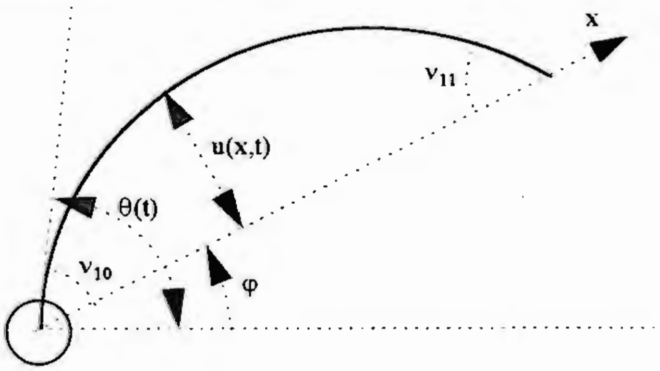


Fig. 4. Variables involved in beam modelling

The beam deformation is modelled in accordance to finite element philosophy (for details see [3]); its relationship with node variables is defined as:

$$u(x,t)=\mathbf{N}^{T}(x)v(t)=\begin{bmatrix}\phi_1(x) & \phi_2(x)\end{bmatrix}\begin{bmatrix}v_{10}(t)\\ v_{11}(t)\end{bmatrix} \qquad (2)$$

where ϕ_1 and ϕ_2 are spatial functions defined, in our case, as third order polynomials (Hermite polynomials). After some algebra manipulations the dynamic equation for the beam is obtained:

$$\mathbf{M(q)\ddot{q}+C(q,\dot{q})\dot{q}+Kq=Q} \qquad (3)$$

where the generalised variable **q** is defined by the vector:

$$\mathbf{q}=\begin{bmatrix}\varphi(t)\\ v_1(t)\\ v_2(t)\end{bmatrix}=\begin{bmatrix}\varphi(t)\\ v_{10}(t)\\ v_{11}(t)\end{bmatrix} \qquad (4)$$

Control Scheme

Based on the control of the pseudolink position, it is defined a following strategy that avoid to solve the inverse kinematic problem for the flexible manipulator, that leads, as it was demonstrated by Bayo [1], to solutions in the Fourier domain. The strategy is based on the following of the pseudolink angle to the hub angle as it is measured by the encoder; trying to vanish the beam deformation. An error position is generated between the pseudolink and the encoder measurement feeding the input of a PD controller (see Fig. 5). In this case there are two control loops; an internal loop, answering quickly, and vanishing the angular position error in the following; and an outer loop that indicates the reference for the desired hub (and tip) position. The first control loop behaves as a vibration suppresser.

For experimental test of the strategy a very quick reference signal was employed, it consists of a position profile reaching π radians in two seconds, staying at that position by two more seconds, and back again to the original position in another two seconds. This sequence was repeated by 30 seconds or more. Control results for the employed strategy can be observed in the Fig. 6. The position errors in this case can be seen in Fig. 7.

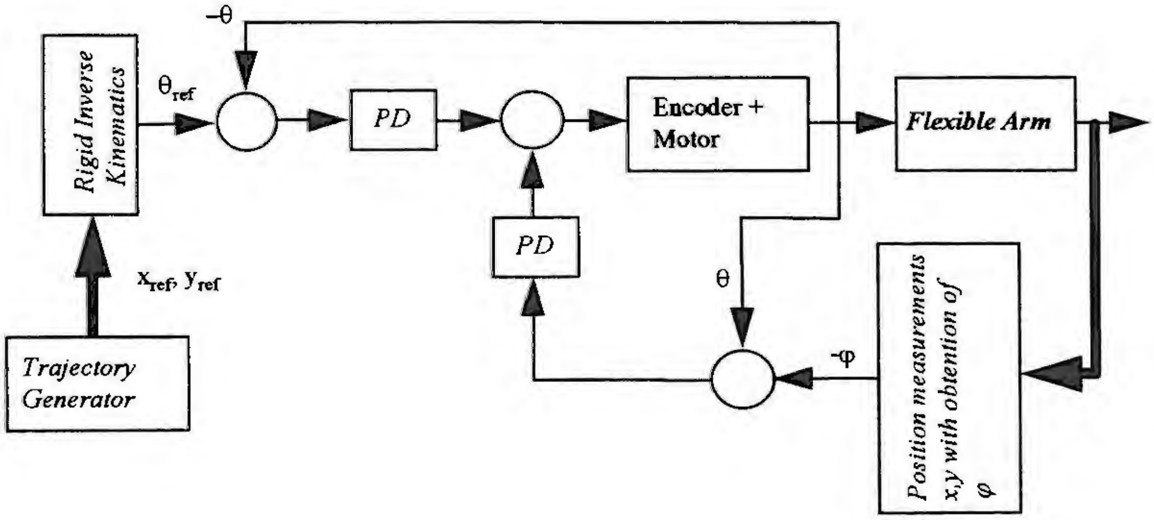


Fig. 5. Pseudolink-based Control scheme. Following strategy

The control used is a collocated control [2] avoiding the non-minimum phase behaviour that it is typical in tip control position of flexible bodies. The Inverse Kinematics block that appears in Fig. 5 it is referred to the rigid pseudolink previously defined, and the inverse problem associated is easy to solve. The overshoot presented in errors shown in Fig. 7 can be diminished with proper tuning of PD. The tuning is not easily encountered due to interactions between both loops. The beam dynamics was simulated and simulations results and experimental data was compared validating the employed model. Due to space limitations only experimental results are shown here; for model validation see [4].

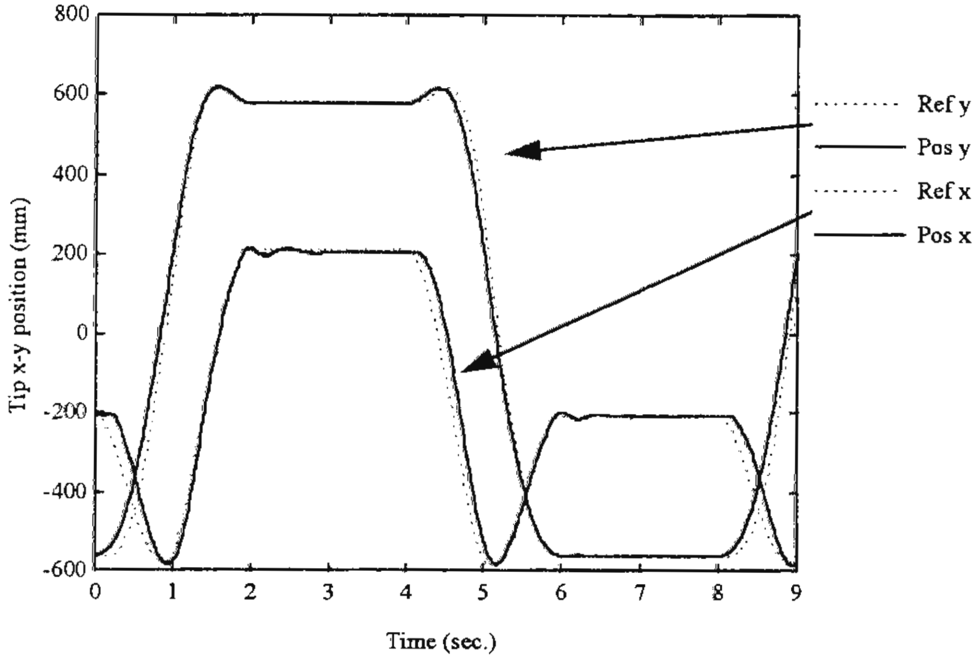


Fig. 6. Experimental measurements of the tip position under the following strategy

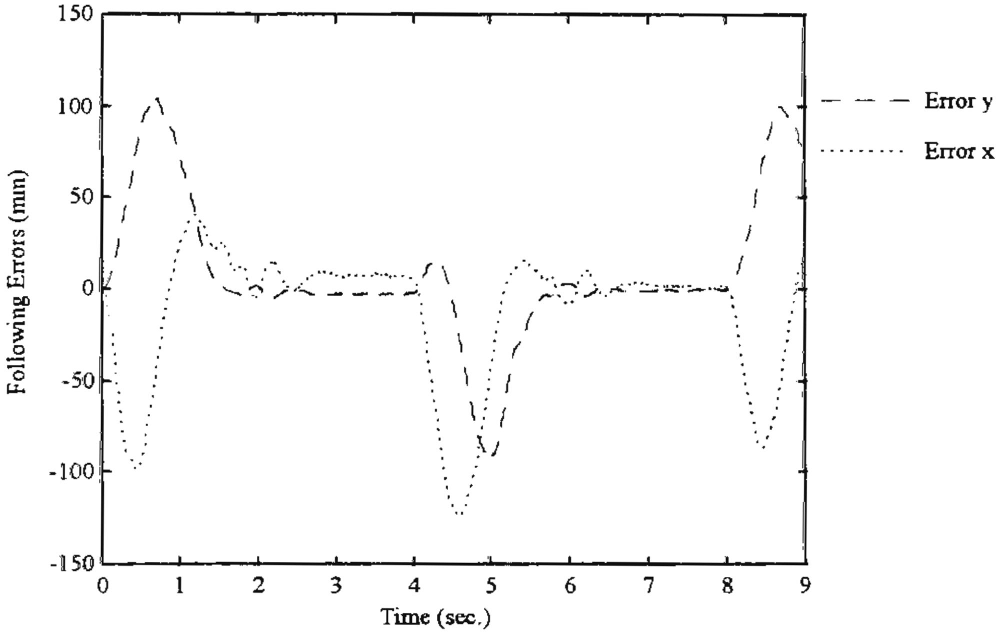


Fig. 7. Tip position following errors

Comparison with other control schemes

The results were compared with other control schemes. Here it is only presented the comparison with the Jacobian Control scheme that it has been developed by the authors [5]. The following errors due to beam oscillations, using the same input, are little bigger for Jacobian Control strategy than in Pseudolink strategy, as it can be observed in Fig. 8.

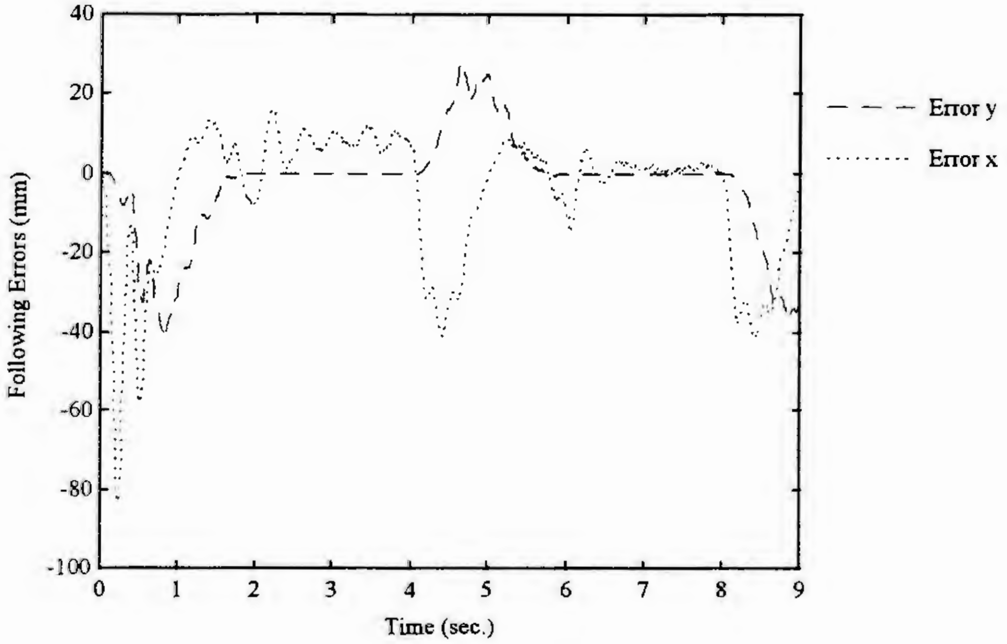


Fig. 8. Following errors in Jacobian Control [5]

Conclusions

A new Control strategy, based on Pseudolink definition, is proposed. The strategy has a very good behaviour in vibration suppression. Due to the nature of the pseudolink definition, the non-minimum phase characteristic of the beam during tip position control is avoided. Also the solution of the Inverse Kinematic problem can be obtained considering a rigid body. The strategy has been compared with other good control results obtained with Jacobian Control; in the future the scheme is going to be compared with other proposed algorithms, founded in the literature, and it will be extended for a 2D manipulator.

References

1. Bayo, E. A finite element approach to control the end-point motion of a single-link Flexible Robot. Journal of Robotic Systems. Vol. 4. N° 1. 1987.
2. Cannon, R.; Schmitz, E. Initial Experiments on the End-Point Control of a Flexible One-Link Robot. The International Journal of Robotics Research. Vol. 3. N° 3. 1984.
3. Chapra, S.; Canale, R. Numerical Methods for Engineers. McGraw-Hill Pub. Co. Second Edition. New York, 1988.
4. Fernández G. Control Multivariable en el Dominio de la Frecuencia de Robots Rígidos y Flexibles. Tesis Doctoral. Universidad de Valladolid. (To be presented).
5. Grieco, J.C.; Fernández, G.; Gamarra-Rosado, V.; Armada, M. Tip Position Control with Hub Position Corrections for a One-Link Flexible Manipulator. DARS'95 Workshop on Human-Oriented Design of Advanced Robotics Systems. Vienna, Austria. September, 1995.
6. Inman, D. Engineering Vibration. Prentice Hall. New Jersey, 1994.
7. López-Linares, S. Control de Robots Flexibles. Tesis Doctoral. Escuela Superior de Ingenieros Industriales de San Sebastián. Universidad de Navarra. Marzo, 1993.
8. Selcom Selective Electronics, Co AB. SELSPOT Hardware Users Manual and Software Users Manual. Partille, Sweden. 1994.

A DYNAMIC INTERPRETATION OF THE CLASSICAL FRICTION MODEL

G. Ferretti, G. Magnani and A. Bonsignore
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133, Milano, Italy

Abstract. The classical friction model (combination of stiction, Coulomb and viscous friction) is commonly defined by a static characteristics that, as such, does not define the friction behaviour from a dynamic point of view when changes occur in the motion status. This paper proposes a dynamic interpretation of the classical friction model and a compact form to describe friction in motion equations that is useful for motion analysis and, especially, may be exploited for numerically robust motion simulations. Simulation results are given showing that the discontinuous model can be seen as an asymptotic approximation, infinitely fast, of a recently proposed continuous, dynamic friction model.

1. Introduction

Friction between bodies in contact is often modelled by a discontinuous characteristic, that combines stiction, Coulomb and viscous friction, relating the friction force (or torque for rotating bodies) to the relative velocity between the bodies [2]. For most applications, this model is effective in explaining the macroscopical effects of the phenomenon, but its sharp discontinuity gives rise to substantial problems both for computer simulation and for theoretical analysis (i.e. stability analysis of motion control systems). A better description of the friction phenomenon for low velocities, and especially when crossing zero velocity, is given by dynamic models derived from a microscopical analysis of the contact dynamics [1,4,6]. These models describe, for instance, the spring-like behaviour during stiction, hysteretic behaviour, and Stribeck effect. Being continuous, they lend themselves very well for motion control analysis, even if, introducing microdisplacements and very fast dynamics in place of instantaneous discontinuities and abrupt transitions in the motion regime, they still involve numerical difficulties in simulation. However, the major difficulty related to using these models concerns the identification of their parameters, that is far difficult with the current instrumentation.

On the other hand, the classical friction model, as expressed by a discontinuous static characteristic, is not suitable to describe the friction behaviour for the purpose of stability analysis and simulation of transients when the velocity vanishes and the motion regime quickly changes. To this purpose, a dynamic interpretation of the classical model is needed. One possible interpretation is given in this paper. It is pointed out the need to distinguish the case in which the velocity vanishes in a single instant (e.g. during motion inversion) from the one in which it vanishes over a time interval. In the first case the value of the friction force in that instant need not to be defined, since it does not affect motion. In the second one, instead, the resultant of the active forces has to be considered to obtain the friction force.

Then the "stiction" function is introduced, which allows one to write motion equations in a compact form and to clearly discriminate a stiction condition (i.e., velocity vanishing over a time interval) from the zero velocity instantaneous crossing.

According to the reformulated model, a finite state machine has been defined that support a realistic and numerically robust simulation of motion equations in presence of friction. Furthermore, the model is suitable for the analysis of limit cycles in position control loops [3], following the theory of the differential equations with discontinuous right-hand sides [8].

The paper is organized as follows: Section 2 outlines the friction model; Section 3 proposes a robust and efficient technique to deal with discontinuous friction simulation; Section 4 compares by simulation the discontinuous friction model and the continuous LUGRE model [4]; Section 5 draws finally some conclusions.

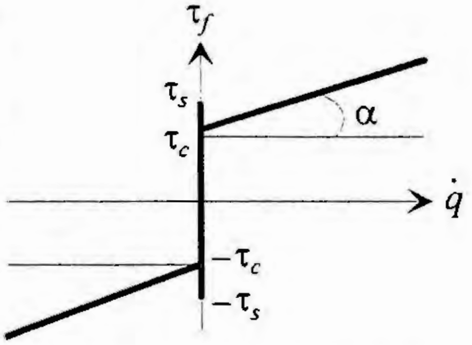


Fig. 1 Coulomb's friction model

2. Friction modelling

The classical discontinuous friction model is described by the static characteristic of Fig. 1, that gives the torque friction τ_f as a function of the relative velocity $\dot{q}$ between two bodies. This model was originated by

Coulomb's and even Leonardo da Vinci's works [1], and represents the friction phenomenon with sufficient accuracy in many applications. The characteristic of Fig. 1 however defines uniquely the friction torque only when $\dot{q} \neq 0$. In this case it is

$$\tau_f = D\dot{q} + \tau_c \operatorname{sign}(\dot{q}), \tag{1}$$

where $D\dot{q}$ ($D = \tan\alpha$) is the viscous friction term and $\tau_c \operatorname{sign}(\dot{q})$ is the Coulomb's term.

When $\dot{q} = 0$ the characteristic just establishes that $\tau_f < \tau_s$, τ_s being the "stiction" torque. To precisely determine the friction torque an additional variable has therefore to be considered, the net active torque τ_a, namely the algebraic sum of the torques acting on the mobile body (assuming for semplicity that only one body is mobile, the others being fixed) apart from friction. Thus, in rest conditions:

$$\tau_f = \tau_a. \tag{2}$$

as shown in Fig. 2, where it is also pointed out that (2) holds for $|\tau_a| \leq \tau_s$.

It is worth noting that the istantaneous value of the total torque does not influence the motion (velocity and position) of the mobile body, thus in case of istantaneous crossing of the zero velocity the friction torque at the crossing instant need not to be defined. Thus eq. (2) is significant only if the velocity vanishes over a time interval, not in a single time instant. For the same reason, to establish whether or not the motion is going to start from a rest condition (incipient motion), the inequality $|\tau_a| > \tau_s$ has to be verified over a time interval.

Consider, as an illustrative example, the motion equation of a rigid rotating shaft:

$$J\ddot{q} = \tau_a - \tau_f, \tag{3}$$

J being the total shaft inertia, assume that the shaft is rotating and, at time instant $\bar{t}$, it occurs $\dot{q} = 0$. If in a right interval of $\bar{t}$, say $I_r(\bar{t})$, it results $|\tau_a| < \tau_s$ the rotation stops. This fact may be interpreted as the instantaneous creation of junctions between the bodies in contact [1], requiring a torque over the stiction threshold to be broken. In $I_r(\bar{t})$ it will be therefore

$$J\ddot{q} = 0$$
$$\tau_f = \tau_a.$$

Conversely, if in $I_r(\bar{t})$ it results $|\tau_a| > \tau_s$ the active torque is able to break the junctions between the bodies and the rotation does not stop. According to the active torque values the rotation may reverse or not. In the first case the friction torque is subject to a step of $2\tau_c$, passing, for instance, from $+\tau_c$ to $-\tau_c$, while in the last one it does not change. In both cases $\tau_f(t)$ does not affect the rotation.

Therefore, to properly define the Leonardo-Coulomb friction model in $\dot{q} = 0$ two cases have to be distinguished, according to the fact that $\dot{q} = 0$ in a single point or in a time interval. To this purpose the following definition of *stiction* is given.

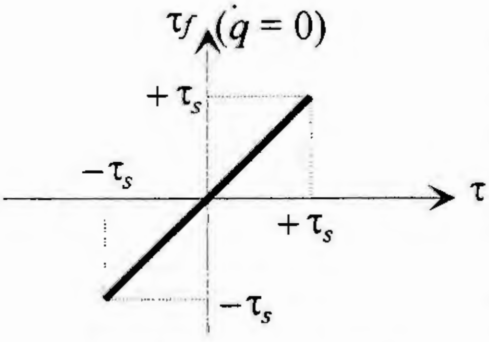


Fig. 2 Friction torque at rest

Definition 1: Stiction

With reference to the relative motion between two bodies in contact, the stiction condition holds if, for a given time instant $\bar{t}$, it results $\dot{q} = 0$ in a neighbourhood $I(\bar{t})$, $\dot{q}$ being the relative velocity. The neighbourhood can be right of $\bar{t}$, left of $\bar{t}$, centered in $\bar{t}$.

Remark 1

With reference to the case of a rotating shaft, if the stiction condition holds in a time instant $\bar{t}$, it results

$$|\tau_a| \leq \tau_s$$

in a neighbourhood $I(\bar{t})$, excluding at most single points.

Definition 2: Static friction (stiction) torque

The static friction (stiction) torque acting on a rotating shaft in stiction conditions is defined as the reaction torque that counterbalances the active torque.

The torque balance for a rigid shaft (3), can now be written with the stiction, Coulomb and viscous friction terms, using conditional equations to account for the discontinuities:

$$\begin{array}{ll} J\ddot{q} = 0\,, & \dot{q} = 0 \wedge |\tau_a| \le \tau_s \\ J\ddot{q} = -D\dot{q} - \tau_c\,\mathrm{sign}(\dot{q}) + \tau_a\,, & \text{otherwise} \end{array} \tag{4}$$

These equations can be rewritten in a compact form introducing the following stiction function.

Definition 3: Stiction function

The following function

$$s(\dot{q},\tau_s,\tau_a) = \mathrm{sign}\left[1+\dot{q}^2 + \mathrm{sign}(\tau_a^2 - \tau_s^2)\right], \tag{5}$$

where the sign function is extended for a null value of the argument as

$$\mathrm{sign}(0) = 0\,,$$

is called stiction function.

Since it results:

$$\begin{array}{ll} s(\dot{q},\tau_s,\tau_a) = 0\,, & \dot{q} = 0 \wedge |\tau_a| \le \tau_s \\ s(\dot{q},\tau_s,\tau_a) = 1\,, & \text{otherwise} \end{array}$$

eqs. (4) can be replaced as follows:

$$J\ddot{q} = s(\dot{q},\tau_s,\tau_a)\cdot[-D\dot{q} - \tau_c\,\mathrm{sign}(\dot{q}) + \tau_a]$$

or, equivalently,

$$J\ddot{q} = s(\dot{q},\tau_s,\tau_a)\cdot\tau_a - D\dot{q} - \tau_c\,\mathrm{sign}(\dot{q})\,.$$

Remark 2: Stiction and stiction function

With reference to the case of a rotating shaft, the stiction condition holds iff the stiction function vanishes in a neighbourhood $I(\bar{t})$, excluding at most $\bar{t}$.

This is consequent to Definition 1 (see also Remark 1) and Definition 3.

Remark 3: Stribeck effect

The Stribeck effect [1], modelled by a nonlinear behavior of the viscous friction term at low velocities, may be simply taken into account by substituting the term $D\dot{q}$ with a given nonlinear function $f(\dot{q})$.

3. Friction simulation

It must be first pointed out that a correct evaluation of the friction model requires an exact detection of the null value of the argument of the sign functions in (5), which cannot be obtained rigorously during numerical simulation. The classical way to cope with this problem consists in assigning a null value to the sign function over a suitable short interval around zero, as done, for example, in the well known Karnopp model [8]. Small errors could be obtained with very short intervals, but short intervals requires also small integration steps to be detected, involving long simulation times with fixed step algorithms or the need to predict a possible stick condition with variable step algorithms, in order to apply a suitable step reduction.

In this paper, a more robust and efficient technique is proposed to compute the stiction function $s(\dot{q},\tau_s,\tau_a)$ when explicit integration methods are adopted. The technique is based on the finite state machine shown in Fig. 3, where the states A, B and C denote respectively stiction, motion and incipient motion (transitions among states hold at the end of an integration step). The values assumed by the stick function s are associated to the arcs of the finite state machine.

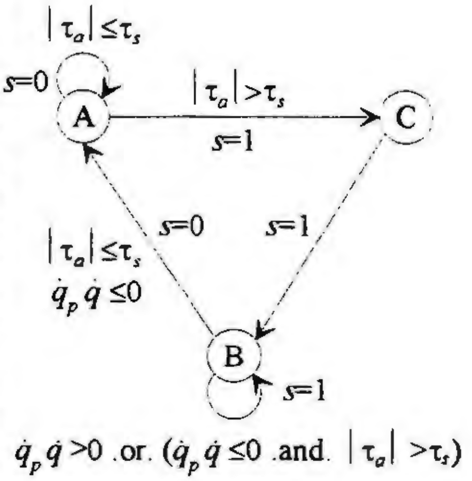


Fig. 3 Finite state machine

The stiction condition obviously holds until $|\tau_a| \le \tau_s$ while for $|\tau_a| > \tau_s$ the transition A → C is fired. In the latter case the motion equation is the following:

$$J\ddot{q} = -\tau_c\,\mathrm{sign}(\tau_{ap}) + \tau_a\,,$$

where the active torque τ_{ap} computed in the past integration step is used in place of the velocity $\dot{q}$ (which is exactly zero during stiction) as the argument of the sign function. The transition C → B is fired unconditionally.

The motion state is maintained when no velocity reversal is computed in the current integration step, thus when $\dot{q}_p \dot{q} > 0$, $\dot{q}_p$ being the velocity computed in the previous integration step, or when a velocity reversal can actually take place since $\dot{q}_p \dot{q} \leq 0$ and $|\tau_a| > \tau_s$. Otherwise, if $\dot{q}_p \dot{q} \leq 0$ and $|\tau_a| \leq \tau_s$ the transition B → A is fired.

The robustness of the technique is greatly improved if a reduction of the integration step is forced whenever a transition is predicted, on the basis of the values of the variables in the current and in the past integration step.

4. Comparing the discontinuous friction model and the LUGRE model

The discontinuous friction model can be actually interpreted as an asymptotic model, where a very fast dynamics is indeed considered as *infinitely* fast. To understand this fast dynamics it is convenient to visualize the contact between two rigid bodies as realized through elastic "bristles" [9], deflecting under the action of a tangential force and generating the friction force. The continuous friction models are based on the average behavior of the bristles, as in the case of the recently proposed LUGRE model [4,5], defined by the following equations:

$$\frac{dz}{dt} = \dot{q} - \frac{|\dot{q}|}{g(\dot{q})} z$$

$$\sigma_0 g(v) = \tau_c + (\tau_s - \tau_c) e^{-(v/v_s)^2}$$

$$\tau_f = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 \dot{q}$$

where z represent just the average deflection of the bristles. In [4], this model has been proved to be effective in modelling some effects not accounted by the classical model, such as the rising of presliding displacements during stiction, the frictional lag and the varying break-away force.

On the other hand the model depends on parameters, in particular σ_0 and σ_1, which can be hardly estimated in practical cases and whose net effect on the relative motion is questionable, provided that sufficiently high values are chosen, implying very fast dynamics for the average deflection of the bristles. On the contrary, the discontinuous model depends only on directly measurable quantities such as the stiction and the Coulomb friction torque.

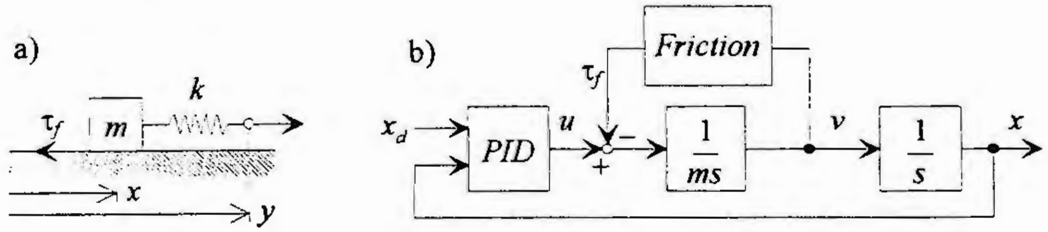


Fig. 4 Simulation experiments

The discontinuous model and the LUGRE model are now compared, repeating the same simulation experiments reported in [4]. In a first experiment (Fig. 4.a), aimed to show the stick-slip motion, a unit mass is attached to a spring with stiffness $k = 2$ N/m, and the end of the spring is pulled with constant velocity $v = 0.1$ m/s. The friction force τ_f is defined by the parameters reported in Table 1.

σ_0	σ_1	σ_2	τ_c	τ_s	v_s
10^5 N/m	$\sqrt{10^5}$ Ns/m	0.4 Ns/m	1 N	1.5 N	0.001 m/s

Table 1. LUGRE model parameters

As it is clear from Figs. 5.a, 5.b, 5.c, which report respectively the position x of the mass, the velocity v and the friction torque τ_f computed with the two friction models (solid line: LUGRE model, dashed line: classical model), the differences between the two model becomes appreciable only after some time. This is mainly due to

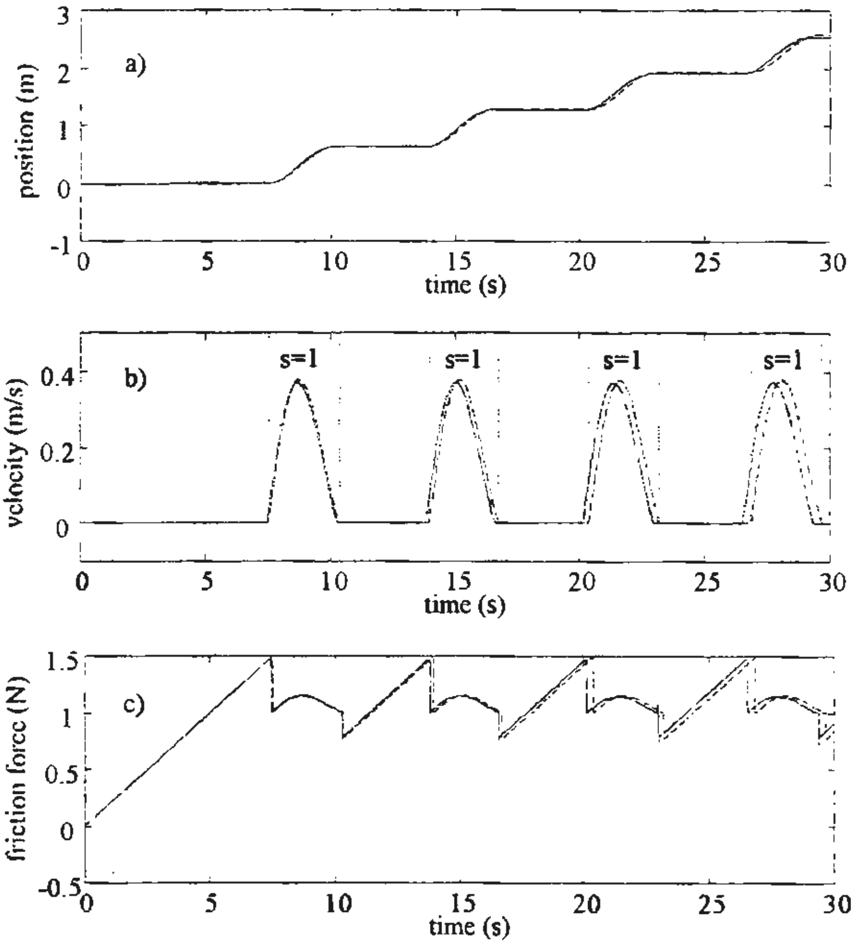


Fig. 5 First simulation experiment (open loop)

the fact that the discontinuous model considers as instantaneous phenomena which in fact take some time to evolve with the LUGRE model.

Another simulation experiment (Fig. 4.b) is aimed to show the (stick-slip) limit cycles arising in closed loop control, when an integral action is adopted. The position of a unit mass is controlled through a PID controller, thus the motor torque is given by:

$$\tau_m = -K_v\dot{q} - K_p(q-q_d) - K_i\int(q-q_d)\,dt\,,$$

where $K_v = 6$, $K_p = 3$ and $K_i = 4$, and a unit step variation to the position set point q_d is imposed. The results, shown in Figs. 6.a, 6.b, 6.c, are very similar to the previous case.

5. Conclusions

For the dynamic analysis and simulation of mechanical servos, the well known classic friction model has to be extended to define the friction behaviour nearby motion status changes, namely when crossing zero velocity, when motion stops, and when motion restarts. In the paper a dynamic interpretation of the classic friction model has been proposed, from which a compact form to describe friction in motion equations, alternative to conditional equations, has been derived. The compact form is more suitable for motion analysis and may be exploited for numerically robust motion simulations. Simulation results have been given, that compare the behaviour of the proposed model with that of the continuous dynamic LUGRE model. Apparently, the macroscopic behaviour of both models are quite similar, so that the discontinuous model can be seen as an asymptotic approximation (infinitely fast) of the continuous one. In addition, the continuous model is able to predict some very fast dynamics, whose parameters, however, seem very hard to get.

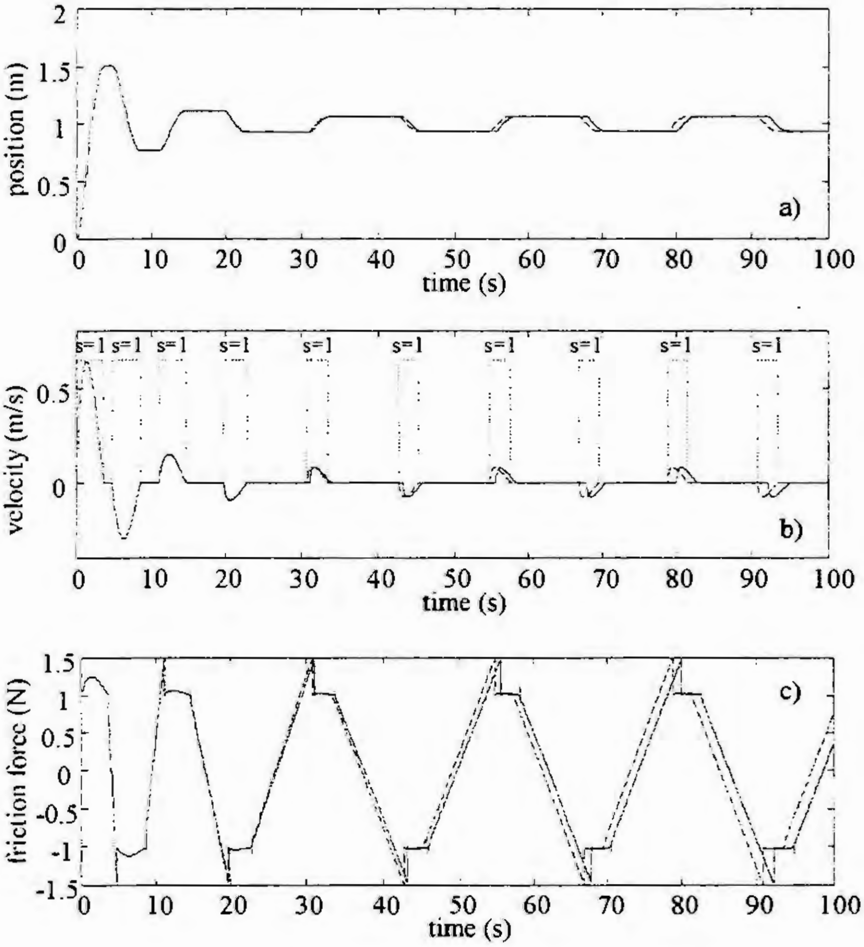


Fig. 6 Second simulation experiment (closed loop)

References

1. Armstrong-Hélouvry, B., Control of machines with friction. Kluwer, Boston, MA, 1991.
2. Armstrong-Hélouvry, B., Dupont, P. and Canudas de Wit, C., A survey of models, analysis tools and compensation methods for the control of machines with friction. Automatica, 30, 7 (1995), 1083-1138.
3. Bonsignore, A. and Cecchetti C., Analisi simulazione e controllo di giunti robotici affetti da elasticità torsionale e attrito, Laurea thesis, Politecnico di Milano, 1994/95
4. Canudas de Wit, C., Olsson, H., Åström, K. J. and Lischinsky P., A new model for control of systems with friction. IEEE Transactions on Automatic Control, Vol. 40, 3 (1995), 419-425.
5. Canudas de Wit, C., Lischinsky, P., Adaptive friction compensation with dynamic friction model, IFAC World Congress, San Francisco, 1996.
6. Bliman, P. A., Mathematical study of the Dahl's friction model. European J. Mechanics. A/Solids, Vol. 11, 6 (1992), 835-848.
7. Filippov, A. F., Differential equations with discontinuous righthand sides, Kluwer Academic Publisher, 1988.
8. Karnopp, D., Computer simulation of stick-slip friction in mechanical dynamic systems. ASME Journal of Dynamic Systems, Measurements and Control, 107, 1 (1985), 100-103.
9. Haessig, D.A. and Friedland, B., On the modeling of simulation of friction. ASME Journal of Dynamic Systems, Measurements and Control, 113, 3 (1991), 354-362.

Model Order Reduction by means of a Continuous Friction Law for a CVT Chain Drive Simulation

J. Srnik and F. Pfeiffer
Technische Universität München
Arcisstr. 21, D-80290 München

Abstract: As a central component of a new hybrid concept of an automotive drive train system, a continuously variable transmission (CVT) is considered. To meet reliable predictions about the dynamic behavior of the CVT a simulation of the vibrations of the chain drive is performed. The chain drive transmits the power exclusively through frictional forces between chain and pulleys and it contains excitation mechanisms known as polygonal action. Hence a discrete model of the chain and the contacts between chain links and pulleys is necessary. A planar description with emphasis on the 3-dimensional contact mechanics is introduced, which contains the major dynamical effects. The large number of bodies and above all the modeling of the contacts by Coulombs friction law lead to time consuming simulations. A continuous approximation of Coulomb results in a reduction of the system's order and hence in lower calculation times, without loosing information as shown in the simulation results.

Introduction

Vibrations of chain drives are subject of many scientific works. Most of them investigate only components like the chain spans [7, 2], or the global behavior, neglecting the polygonal excitation and approximating the contact mechanics between chain and pulleys [6]. Only a view studies take into account the structure of the chain. Nakanishi and Shabana [3] developed a detailed model for tracked vehicles, describing chain links, sprockets and pulleys as single bodies. Fritz and Pfeiffer present a similar extensive model of a roller chain drive [1].

In contrast to roller chain drives, CVT chain drives (Figure 1) transmit power exclusively through frictional forces in the contact zones between the bolts of the chain and the cone sheaves of the pulleys. Every contact has two possible states, sticking or slipping, depending on the relative velocity. The possible transitions between them lead to a mechanical system with variant structure. Solution methods for such systems are suggested by Pfeiffer and Glocker [4] and are applied on the model of the CVT-chain drive by Srnik and Pfeiffer [5].

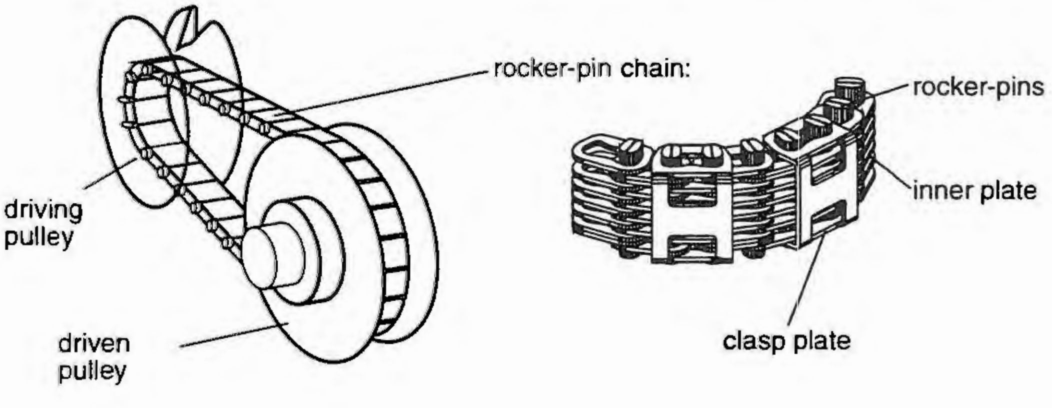


Figure 1: CVT chain drive

The discrete structure of the chain causes excitations and determines the vibrational behavior of the entire system. Therefore it is necessary to model every chain link and the two pulleys as separate bodies.

Containing all the major chain effects, a planar description of the chain drive is introduced. Only the contacts between chain links and pulleys require a three dimensional model.

Mechanical model of the bodies

Every chain link and pulley is modeled as a single body with three degrees of freedom $q^T = (x, y, \alpha)$ (Figure 2). The pulley is connected with the inertial environment through a force element, representing the bearing elasticities and the bending of the shaft. It is loaded with an external torque M and the frictional and normal contact forces acting between the pulley's cone sheaves and the bolts of the link.

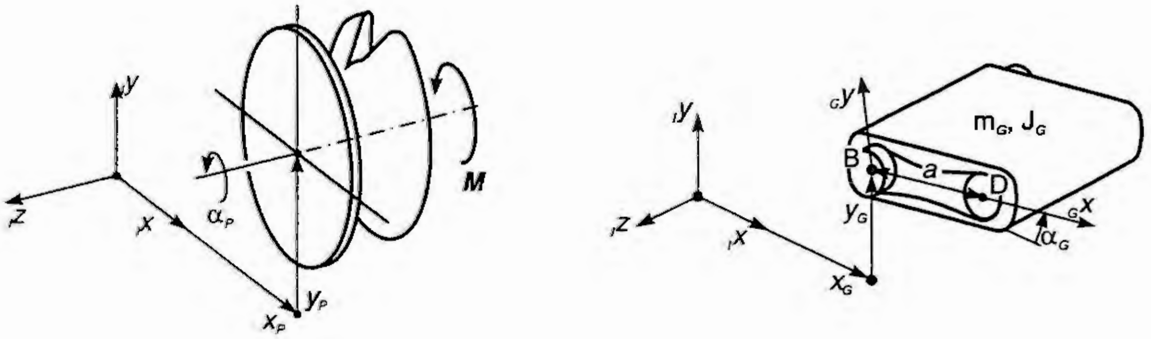


Figure 2: Model of a pulley (left) and a chain link (right)

A chain link is connected with its neighbors by force elements taking into account the elasticity and damping of the link and the joint. When it is in contact with a pulley the frictional and normal contact forces act on the bolts of the chain link and therefore on the link.

Mechanical model of the bolts

The pair of rocker pins (Figure 1) can be modeled as one single, massless spring acting exclusively perpendicular to the model plane. Figure 3 shows the bolt's model and the forces acting in the contact plane. The contact planes of the bolts are the spring's surfaces. They are parallel to the conical inner surfaces of the pulley. The normal contact force F_N acts perpendicular to the contact plane, whereby the frictional forces F_{Rr} and F_{Rt} are parallel to this plane. Therefore we have to deal with a three-dimensional contact.

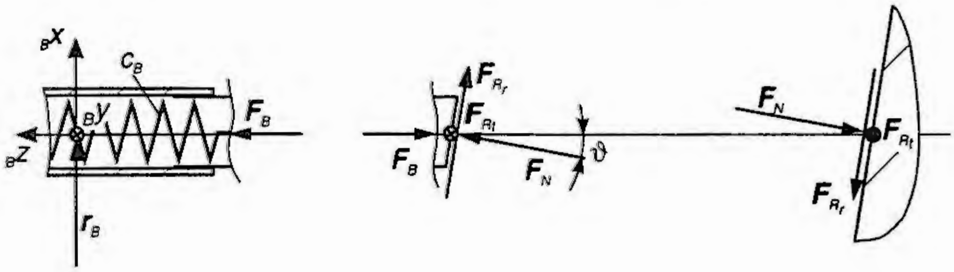


Figure 3: Model of a bolt

For the derivation of the contact forces it is necessary to quantify the bolt's spring force. It depends on the bolt's length l_B and stiffness c_B as well as on the local distance of the pulley's surfaces s (Figure

4) :

$$F_B = \begin{cases} c_B\,(l_B - s) = c_B\,\Delta l_B & \wedge \quad s \leq l_B \\ 0 & \wedge \quad s > l_B \end{cases} \tag{1}$$

The static equilibrium of forces perpendicular to the model plane provides a conditional equation for the normal force, which depends on the minimal coordinates of the according chain link and the contact force in radial direction:

$$F_N = F_{R_r}\tan\vartheta + \frac{F_B}{\cos\vartheta} = F_{R_r}\tan\vartheta + \frac{c_B\,(l_B - s)}{\cos\vartheta} \tag{2}$$

ϑ is the cone angle of the sheaves of the pulleys. To determine the remaining frictional forces in dependency of the normal force, Coulomb's friction law is used:

$$\begin{aligned} \text{slipping:}\quad \boldsymbol{F}_R &= -\mu_0\,F_N\,\frac{\dot{\boldsymbol{g}}}{\dot g}; \quad \dot g = |\dot{\boldsymbol{g}}| \\ \text{sticking:}\quad F_R &= \sqrt{F_{R_r}^2 + F_{R_t}^2} \leq \mu_0\,F_N; \quad F_R = |\boldsymbol{F}_R| \end{aligned} \tag{3}$$

In the case of slipping a uniform dependency for the calculation of the frictional forces exists and all contact forces can be calculated. In the case of vanishing relative velocity $\dot g$ a transition to sticking occurs.

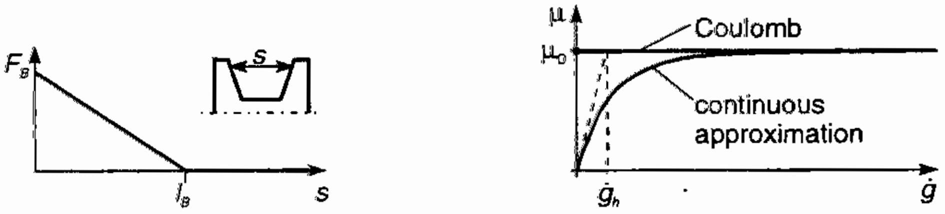


Figure 4: Characteristic of the bolts force (left), friction characteristic (right)

The second part of the friction law (3) gives an upper limit for the sticking force. Therefore additional conditions have to be taken into account to calculate the sticking force. The kinematic condition for sticking is a vanishing relative velocity at the contact points $\dot g = 0$, which for a sticking as well as a breaking loose contact is equal to the condition of vanishing relative acceleration $\ddot g = 0$. With this equations all unknown frictional forces can be determined by Coulomb's friction law.

During the numerical simulation transitions between sticking and slipping occur. The corresponding switch points have to be determined, which is one reason for extraordinary high calculation times. The other main reason for this is the the high number of contacts and the connected determination of the correct configuration of sticking and slipping states in each of them. For this purpose the additional conditions have to be solved, forming a large time consuming system of equations.

To reduce the contact's calculation costs, two possibilities exist. The first uses a continuous chain model. Hence only two contacts, one for each pulley, have to be taken into account. This also neglects the polygonal effect and the conjoined excitation mechanisms, which are a major subject of the current studies.

The second possibility for reducing the system's order is the application of a continuous approximation of Coulomb's friction law (Figure 4 right):

$$\boldsymbol{F}_R = -\mu\,F_N\,\frac{\dot{\boldsymbol{g}}}{\dot g}; \qquad \mu = \mu_0\,(1 - \exp(-\frac{\dot g}{\dot g_h})) \tag{4}$$

The factor $\dot g_h$ defines the gradient of the curve for $\dot g = 0$. $\dot g_h$ is a measure for the deviation of the approximation from the exact solution. By this friction law the frictional forces are uniquely determined by the normal force for any relative velocity. Therefore the time consuming determination of the switching points and contact configurations is no longer necessary.

Mathematical model

The equations of motion follow from the equations of momentum and the equations of moment of momentum of each single body. They are transformed into the space of the minimal coordinates by the corresponding Jacobian-matrices. Calculating the frictional forces by Coulombs friction law (3) the system's equations of motion with the additional conditions for sticking contacts can be written as follows:

$$\begin{aligned} &\boldsymbol{M}\,\ddot{\boldsymbol{q}} = \boldsymbol{h} + \sum \boldsymbol{W}_i\,\boldsymbol{\lambda}_i \\ &\ddot{g}_i = 0 \wedge \lambda_{0_i} \geq 0\,; \quad \ddot{g}_i \geq 0 \wedge \lambda_{0_i} = 0\,; \quad \ddot{g}_i\lambda_{0_i} = 0\,; \quad \ddot{g}_i = |\ddot{\boldsymbol{g}}_i| \end{aligned} \tag{5}$$

The matrix $\boldsymbol{M}$ represents the mass matrix, the vector $\boldsymbol{q}$ contains all minimal coordinates and the vector $\boldsymbol{h}$ all active forces. $\boldsymbol{\lambda}_i$ is the vector of sticking forces of the $i-$th sticking contact. It is transposed into the space of the minimal coordinates by the constraint matrix $\boldsymbol{W}_i$. To evaluate the sticking forces $\boldsymbol{\lambda}_i$ additional complementary conditions are taken into account, where $\lambda_{0_i} = \mu_0\, F_{N_i} - |\boldsymbol{\lambda}_i| \geq 0$ is the frictional saturation.

Due to the transitions from sticking to slipping and vice versa and the corresponding changes of the number of degrees of freedom, the differential equations (5) are time variant. The according switch points have to be determined, which leads to high calculation times.

Using the continuous friction law (4) the system has a constant number of degrees of freedom. The vector $\hat{\boldsymbol{h}}$ contains all contact forces which are always active in this case:

$$\boldsymbol{M}\,\ddot{\boldsymbol{q}} = \hat{\boldsymbol{h}} \tag{6}$$

The tangential constraints include no sticking, because the frictional force is equal zero for vanishing relative velocity (Figure 4 right). Hence a certain slippage exists depending above all from the factor $\dot{g}_h$. Because the system contains only continuous force characteristics, no additional conditions have to be taken into account. Therefore the model's order has declined and the calculation times have been decreased by a significant factor.

Simulation Results

The results presented in the following are computed for a stationary working order with constant driving

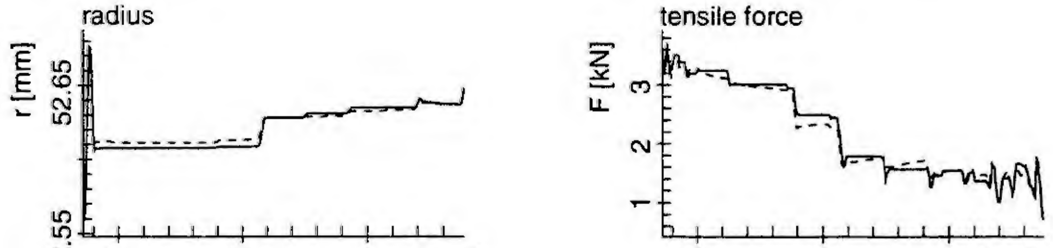


Figure 5: Radius (left) and tensile force (right) of a chain link in the driving pulley; Coulombs friction law: —— , continuous friction characteristic: - - - -

speed and output torque on geared level. Computing time for this case amounts about 80 hours in the case of Coulombs friction law and 3 hours for the continuous friction characteristic on a SUN-SPARC-10 workstation.

The comparison of the two models by means of the time-dependent radius and tensile force in a chain link during its surrounding of the driving pulley shows the differences of the two models most obviously (Figure 5). Generally both models provide results, which are close together. However for the steady model the sticking phases, which are recognized in the discontinuous model by exact constant values, are only approximately constant. On the other side the calculation time decreases by at least one order of magnitude. Thereby larger values of the coefficient $\dot{g}_h$ in equation (4) lead to higher calculation times and to a more exact approximation of the unsteady case.

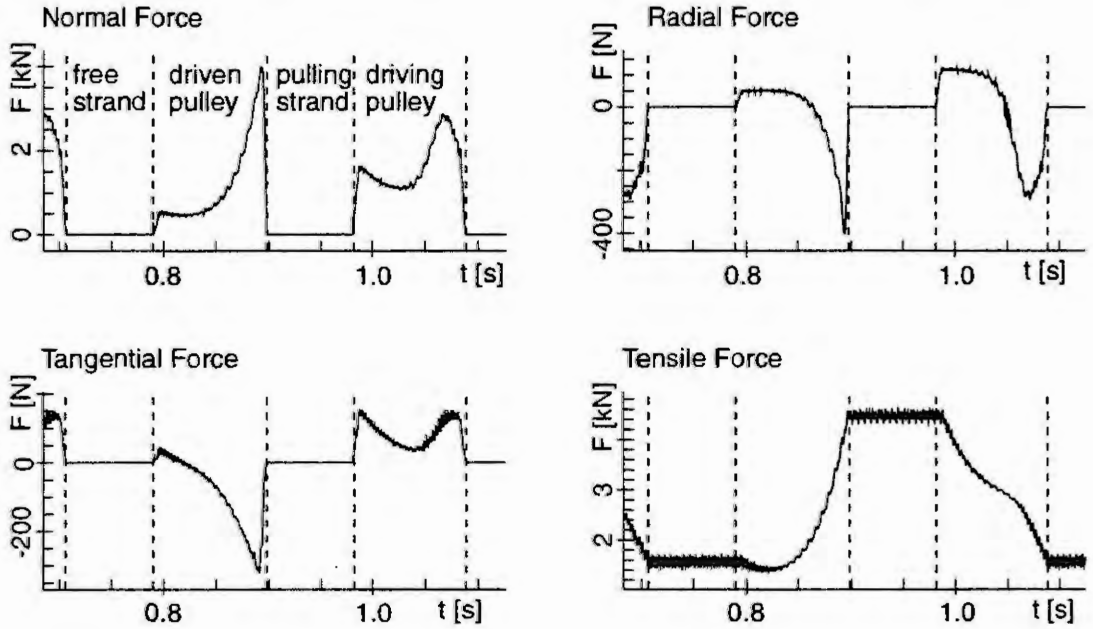


Figure 6: forces acting on the rocker pins and in a chain link

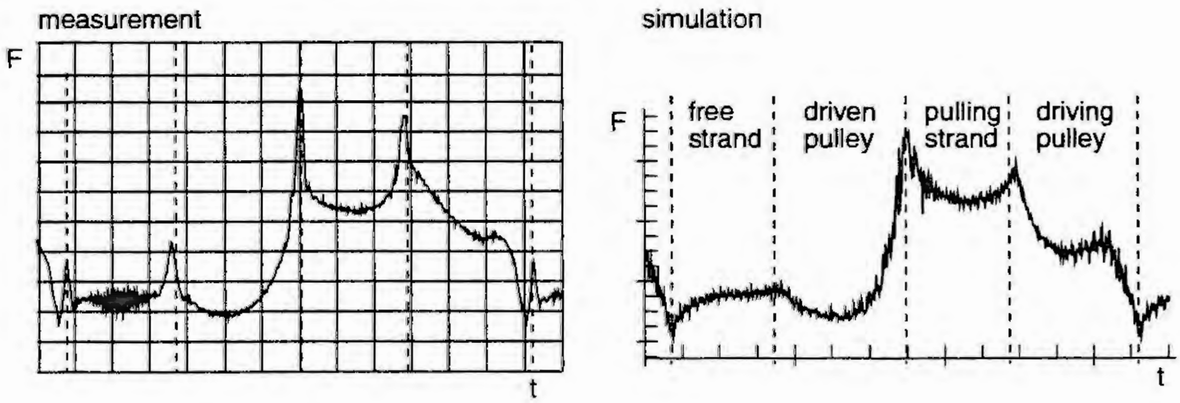


Figure 7: tensile force in the clasp plate[1]

Since the accordance of both models is proven the faster simulations are used for further investigations. Figure 6 shows the contact forces acting on a pair of rocker pins during one revolution as well as the tensile force in the related chain link. As long as the chain link is part of a strand no contact forces work on its rocker pins. When it comes into contact with one of the pulleys the pins are pressed between the two sheaves and hence the normal force increases. Its amplitude depends on the geometry of the sheaves and the transmitted power. The frictional force is a function of this normal force and the relative velocity between the pulley and the pins. It is split into one radial and one tangential component. The radial contact force coincides with a radial movement of the chain link which equals a power dissipation. In contrast to this the tangential contact force causes the changes of the tensile force in the corresponding chain link, leading to different tensile force levels in the two strands which agree with the transmitted torque.

Due to the mechanical model the simulation provides an integrated value of the tensile force in the plates of a chain link, whereas measurement was performed for the tensile force in a clasp plate[1] (Figure 1). Therefore it is necessary to determine the distribution of the tensile force on the plates of the chain links. Using the results of the dynamic simulation shown in Figure 6 and modeling the pair of rocker pins as bending beams as well as the plates as linear springs we get the graph of Figure 7 right for the clasp plate. The comparison of simulation and measurement confirms the mechanical model.

Acknowledgement

The research work presented in this paper is partially supported by a contract with the DFG (Deutsche Forschungsgemeinschaft, SFB 365).

References

[1] P. Fritz and F. Pfeiffer, *Dynamics of High Speed Roller Chain Drives*, to appear in Proceedings of the 15th Biennial Conf. on Mechanical Vibrations and Noise, Boston, Sept. 17-21 1995, ASME.

[2] A. Fritzer, *Nichtlineare Dynamik von Steuertrieben*, no. 176 in 11, VDI Fortschritt-Berichte, VDI-Verlag, Düsseldorf, 1992.

[3] T. Nakanishi and A. A. Shabana, *Contact Forces in the Non-Linear Analysis of Tracked Vehicles*, Int. Journal for Numerical Methods in Engineering, 37 (1994), pp. 1251–1275.

[4] F. Pfeiffer and C. Glocker, *Multibody Dynamics with Unilateral Contacts*, Wiley & Sons, 1996.

[5] J. Srnik and F. Pfeiffer, *Dynamik von CVT-Kettengetrieben: Modellbildung und -verifikation*, no. 1285 in VDI Berichte, 1996, pp. 441–455.

[6] K. W. Wang, *On the Stability of Chain Drive Systems under Periodic Sprocket Oszillations*, ASME Design Technical Conf. - 13th Biennial Conf. on Mechanical Vibration and Noise, Miami, FA, vol. DE-36, ASME, Sept. 22-25 1991, pp. 41–50.

[7] G. Yue and L. Eng, *Belt Vibration Considering Moving Contact between Belt and Pulley*, JSME Int. Conf. on Motion and Powertransmissions, JSME, Nov. 23-26 1991, pp. 411–416.

[1] Measurements performed by G. Sauer and K. Th. Renius, Lehrstuhl für Landmaschinen, Technical University of Munich

EXPERIMENTAL ESTIMATION OF DYNAMIC PARAMETERS FOR AN INDUSTRIAL MANIPULATOR

G. Antonelli, F. Caccavale and P. Chiacchio
Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21, 80125 Napoli, Italy

Abstract. This work deals with the problem of estimation of dynamic parameters for a conventional industrial manipulator. By exploiting the property of linearity in the parameters of the dynamic model, a procedure based on a least-squares algorithm is set up to estimate the dynamic parameters. The set of data for the estimation is collected along a suitable trajectory, planned in order to achieve an optimal degree of excitation for the manipulator dynamics. The experiments have been performed on a conventional industrial manipulator. In spite of the nonideal conditions in which the experiments have been performed, the results show that a good modeling accuracy has been achieved. In order to reduce the computational burden of the identified model, a simple model reduction procedure is experimentaly tested.

Introduction

The availabilty of an accurate model of the manipulator is of the utmost importance both for model-based control and simulation in robotics. On the other hand, the request for reliable simulation tools for robotic systems pushes toward the accurate modeling of the robotic systems.

Manipulator dynamic equations are coupled and highly nonlinear and their structure can be derived from Lagrange or Newton-Euler formulation [2]. In order to achieve an accurate modeling of the system, the knowledge of the dynamic parameters is required as well.

Algorithms have been proposed in the literature for the identification of the dynamic parameters of robot manipulators. The set of data for the estimation are obtained by executing suitably exciting trajectories and measuring the resulting motion of the structure together with the commanded joint torques. The off-line estimation can performed by resorting to least-squares techniques [2,3].

In order to improve the quality of the estimation, the experimental data should be collected along an optimally exciting trajectory [4,8]. The choice of such a trajectory leads to the solution of an optimization problem whose dimension increases with the number of robot's degrees-of-freedom.

The focus here is on a conventional industrial manipulator, the SMART-3 S manufactured by COMAU. The dynamic model is derived in terms of a minimum set of dynamic parameters [5]; the estimation is then performed via the algorithm proposed in [2] by using the data collected during the execution of the manipulator's motion. The procedure is validated by comparing the obtained model outputs with direct measurements on the manipulator collected along some test trajectories.

The conditions of the experiments are far frome the ideal ones required by the theory: unmodeled dynamics are present, constraints on the choice of the identification trajectories are imposed by the mechanical structure. In spite of the above factors, the results of the identification in terms of modeling accuracy are more than acceptable.

Finally, a model reduction technique is presented, aimed at eliminating the parameters which give negligible contributions to the joint torques. This leads to an approximated model whose computational burden decreases while ensuring still a good accuracy.

Identification algorithm

The algorithm proposed in [2] is based on the existence of a linear relationship between dynamic parameters and joint torques.

Let us consider a manipulator with n joints, then the set of dynamic parameters to be considered for each link i is given by: the mass m_i, the three components of the first-order moment $m_i \boldsymbol{r}^i_{i,c}$ (hereafter denoted by $\boldsymbol{mc}_i$), six components of the symmetric inertia tensor $\boldsymbol{I}_i$, the actuator inertia $I_{i,mot}$, the friction coefficients $b_{s,i}$ and $b_{v,i}$ (2 parametrs). The total is $13n$ parameters.

Regarding the friction torque $\tau_{f,i}$ at each joint, several models have been proposed in the literature (see [6] for a thorough review). The following model has been chosen for the identification:

$$\tau_{f,i} = b_{s,i}\mathrm{sgn}(\dot{q}_i) + b_{v,i}\dot{q}_i \qquad |\dot{q}_i| \geq \dot{q}_{i,s}, \tag{1}$$

where $b_{s,i}$ and $b_{v,i}$ are the stiction and friction coefficients of the joint i, respectively, and $\dot{q}_{i,s}$ is a treshold value. The case $|\dot{q}_i| < \dot{q}_{i,s}$ has not been considered since it is not easy to derive a correct model for stiction.

The dynamic model of the manipulator can be always written in a form linear in the dynamic parameters [1]

$$\tau = P(q, \dot{q}, \ddot{q})\gamma = P(\chi)\gamma, \qquad \chi = \begin{bmatrix} q \\ \dot{q} \\ \ddot{q} \end{bmatrix}, \tag{2}$$

where γ is the $(13n \times 1)$ vector of parameters, P is an upper triangular matrix of dimension $(n \times 13n)$, q is the $(n \times 1)$ vector of joint variables, τ is the $(n \times 1)$ vector of joint torques and χ is a $(3n \times 1)$ vector collecting the joint variables, velocities and accelerations at each time instant.

By measuring joint positions, velocities, accelerations and torques during a trajectory at N time istants $t_1, t_2, \ldots, t_N$ $(nN \geq 13n)$, it is possible to write the following relation:

$$\bar{\tau} = \begin{bmatrix} \tau(t_1) \\ \tau(t_2) \\ \vdots \\ \tau(t_N) \end{bmatrix} = \begin{bmatrix} P(\chi(t_1)) \\ P(\chi(t_2)) \\ \vdots \\ P(\chi(t_N)) \end{bmatrix} \gamma = \bar{P}\gamma, \tag{3}$$

where each vector $\chi(t_i)$ is to be considered as a measurement point in $\mathcal{R}^{3n}$. Equation (3) represents an overdetermined system of equations which has no exact solution and matrix $\bar{P}$ is in general not full rank. In the inversion of (3) this problem could be overcome by using the damped least-squares problem solution [7]

However, it is worth noticing that the dynamic parameters can be regrouped in three categories: completely identifiable, unidentifiable and identifiable in linear combinations. Unidentifiability of some parameters is due to the fact there is a restricted degree of relative motion between the links and/or actuators can apply torque only around joint axes; these parameters are not necessary in the model and can be eliminated, thus reducing the dimension of matrix γ. Identifiability in linear combination also reduces the dimension of γ and is not restrictive since we are interested in computing the dynamic model, not the single parameters. In order to identify linear combinations of parameters a method is based on the singular value decomposition of the matrix $\bar{P}$ has been used [5].

Thus, it is always possible to rewrite the dynamic model in terms of a minimum set of p $(p < 13n)$ dynamic parameters so that the matrix $\bar{P}$ in (3) is full rank. In this case the least-squares solution of (3) is

$$\gamma_e = (\bar{P}^{\mathrm{T}}\bar{P})^{-1}\bar{P}^{\mathrm{T}}\bar{\tau}, \tag{4}$$

where γ_e is the vector of estimated parameters. If an *a priori* knowledge about the values of the dynamic parameters is available, a weighted pseudo-inverse of $\bar{P}$ can be used in (4) to improve the efficiency of the estimation on each parameter [8].

Optimal trajectory planning for identification

In order to improve the efficiency of the least-squares algorithm the persistent excitation matrix $\bar{\Pi} = \bar{P}^T\bar{P}$ [4], or simply the matrix $\bar{P}$ [8], should have a small condition number and an high minimum singular value. Thus, a small sensitivity of the estimation algorithm to measurement noise and unmodeled dynamics can be obtained if the following cost function is minimized to find the N measurement points $\chi(t_i)$ $(i = 1, N)$ [8]

$$f(\chi(t_1)\ldots\chi(t_N)) = \lambda_1\mathrm{cond}(\bar{P}) + \lambda_2\frac{1}{\sigma_m(\bar{P})}, \tag{5}$$

where $\sigma_m(\bar{P})$ is the minimum singular value of $\bar{P}$ and λ_1, λ_2 are two weighting scalars.

If p is the number of parameters (i.e., the dimension of γ is p), then the minimum number of measures N required to build the matrix $\bar{P}$ in (4) is such that

$$nN \geq p, \tag{6}$$

and thus the minimum number of variables for the optimization problem is $3nN \geq 3p$.

In order to reduce the computational complexity of the optimization problem, the number of the measurement points of the optimal set can be chosen as the minimum needed to fulfill (6). Then, after the optimal set of points has been determined, they can be interpolated (e.g., using fifth order polynomials) to obtain a smooth joint trajectory to be executed from the manipulator. A strategy to obtain a good set of measures is to interpolate the same set of optimal points several times, changing the order in which the points are taken. In this way the same set of optimal points is reached with a different history in the intermediate time intervals (i.e., the time interval between two consecutive optimal samples); this helps in counteracting the effects on nonzero mean error sources (e.g., unmodeled dynamics).

However, experimental results show that the increase of the number of measures results in a better conditioned $\bar{P}$ and helps to counteract the effects of noise and errors on the measures. Thus, in order to obtain a larger number of measures additional samples between the optimal points are taken along the trajectories at sampling rate of T'; experimental tests show that the resulting $\bar{P}$ is better conditioned with respect to that built by using only the optimal set of measurement points.

Experimental setup

This work focuses the attention on a conventional industrial manipulator, the SMART-3 S manufactured by COMAU. The manipulator has six joints and is mounted on a sliding track providing an extra degree of freedom. Each joint is actuated by brushless motors via gear trains; shaft absolute resolvers provide motor position measurements, and thus joint velocities and accelerations have to be reconstructed via numerical filtering. The original controller of the SMART-3 S is the C3G 9000, a VME-based system. The manipulator is equipped with an 'open' version of the C3G 9000; a bus-to-bus communication link is estabilished with a standard Pentium/133 personal computer, on which the user can implement control algorithms as C modules.

During each sampling interval, the PC receives the motor position measurements from the relsovers and the joint references from the trajectory generation module; then, the torque references are computed on the basis of a standard PID control. The overall computation allows the control to be run at 1 ms sampling time. The reference currents are passed to the servos through the communication link.

For estimation purposes the structure consisting of the first three joints and the sliding track has been considered ($n = 4$).

The full dynamic model in terms of the parameters has been derived in symbolic form by using the software package Mathematica. The matrix $\bar{P}$ in (3) has been computed at random points χ. The rank of the matrix is resulted to be 27 that is the dimension of the minimum set of dynamic parameters ($p = 27$). By means of a singular value decomposition of the same $\bar{P}$ [5], a set of independent dynamic parameters has been found:

$$\begin{array}{lll}
\gamma_1 = m_4 + I_{4,mot} + m_1 + m_2 + m_3 & \gamma_{10} = b_{s,4} & \gamma_{19} = I_{3,xy} \\
\gamma_2 = I_{1,yy} + I_{1,mot} + I_{2,yy} + I3, yy + l_1^2(m_2 + m_3) + l_2^2 m_3 & \gamma_{11} = b_{v,4} & \gamma_{20} = I_{3,xz} \\
\gamma_3 = mc_{1,x} + l_1(m_2 + m_3) & \gamma_{12} = b_{s,1} & \gamma_{21} = I_{3,yz} \\
\gamma_4 = mc_{1,z} + mc_{2,z} + mc_{3,z} & \gamma_{13} = b_{v,1} & \gamma_{22} = I_{3,zz} \\
\gamma_5 = I_{2,xx} - I_{2,yy} - l_2^2 m_3 & \gamma_{14} = I_{2,xy} & \gamma_{23} = mc_{3,x} \; , \\
\gamma_6 = I_{2,xz} - l_2 mc_{3,z} & \gamma_{15} = I_{2,yz} & \gamma_{24} = mc_{3,y} \\
\gamma_7 = I_{2,zz} + I_{2,mot} + l_2^2 m_3 & \gamma_{16} = mc_{2,y} & \gamma_{25} = I_{3,mot} \\
\gamma_8 = mc_{2,x} + l_2 m_3 & \gamma_{17} = b_{s,2} & \gamma_{26} = b_{s,3} \\
\gamma_9 = I_{3,xx} - I_{3,yy} & \gamma_{18} = b_{v,2} & \gamma_{27} = b_{v,3}
\end{array}$$

where l_i is the length of the link i, and the link 4 is the sliding track.

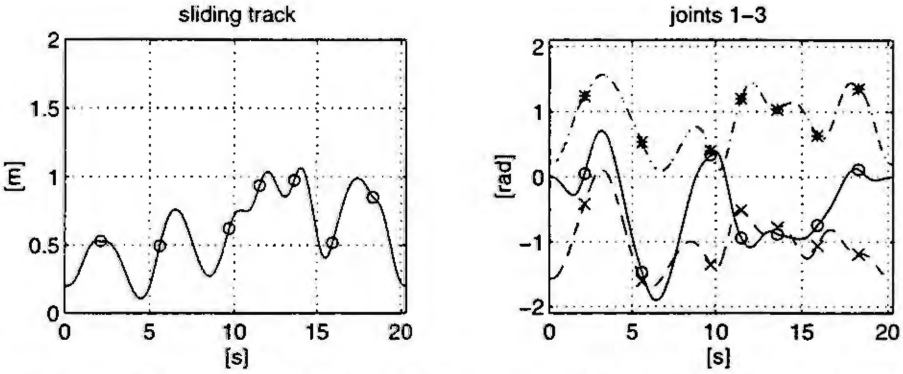


Figure 1. One of the interpolations of the optimal points (positions); *solid*—joint 1, *dashed*—joint 2, *dashdotted*—joint 3; the optimal points are marked with the symbols 'o', 'x' and '$\star$' respectively.

Parameters estimation

The optimization problem (5) has been solved numerically using the procedures in the Optimization Toolbox of Matlab. The minimum number of mesurement points which fulfils (6) is $N = 7$; this results in $3Nn = 84$ variables for the optimization problem. The obtained condition number of $\bar{P}$ is 220, while its minimum singular value is 0.16.

Then, five different trajectories are obtained by interpolating the set of optimal points taken in different orders. The joint positions of one of these trajectories is showed in Fig. 1.

In order to obtain a feasible trajectory both the optimization problem and the interpolation procedure must take into account the constraints on maximum and minimum position, velocity and acceleration at each joint.

The five trajectories have been executed by the manipulator and the data, motor positions and currents, have been collected at a sampling rate $T = 10$ ms. The joint velocities and accelerations needed in (3) are computed by using a non-causal filter from the joint angle histories which is the only information available from the C3G controller.

The resulting $\bar{P}$ has been built taking the samples along the trajectory every $T' = 50$ ms; it has a condition number of 78.2 and a minimum singular value of 7.36. The estimates of the parameters, obtained through (4) are:

$\gamma_1 =$ 7.4519e+02	$\gamma_{10} =$ 1.8267e+02	$\gamma_{19} =$ 5.5895e+00
$\gamma_2 =$ 1.1289e+02	$\gamma_{11} =$ 9.6956e+02	$\gamma_{20} =$ 2.0113e+00
$\gamma_3 =$ 5.1899e+01	$\gamma_{12} =$ 2.5341e+01	$\gamma_{21} =$ 3.2987e−02
$\gamma_4 =$ 8.5828e+00	$\gamma_{13} =$ 9.9625e+01	$\gamma_{22} =$ 1.5677e+01
$\gamma_5 =$−6.3285e+01	$\gamma_{14} =$ 1.8929e+00	$\gamma_{23} =$−6.4657e+00
$\gamma_6 =$−9.8991e−01	$\gamma_{15} =$ 3.8038e+00	$\gamma_{24} =$ 4.4833e+00
$\gamma_7 =$ 8.1287e+01	$\gamma_{16} =$−1.5370e+00	$\gamma_{25} =$ 5.7040e+00
$\gamma_8 =$−1.7049e+01	$\gamma_{17} =$ 3.7963e+01	$\gamma_{26} =$ 5.9335e+01
$\gamma_9 =$−9.9698e+00	$\gamma_{18} =$ 1.2218e+02	$\gamma_{27} =$ 3.7179e+01

Notice that the values $\gamma_1, \ldots, \gamma_{27}$ are not to be regarded as the actual values of the dynamic parameters since they come from a least squares problem solution.

To test the correctness of the identification procedure different trajectories have been executed and the measured joint torques have been compared with those obtained by computing the dynamic model with the identified parameters on the same trajectory. The time histories of the joints for one of these trajectories are plotted in Fig. 2. In Fig. 3 the measured and computed torques are reported showing that a good estimation has been performed. The residual error is mainly due to unmodeled effects and to the non-ideal conditions in which the identification has been performed.

Model reduction

The set $\gamma \in \mathcal{R}^p$ of estimated parameters may contain also some parameters whose contribution to the joint torques can be considered negligible. In this case, the model can be further reduced by eliminating these parameters in the model together with their corresponding columns in $\boldsymbol{P}$. Of course, the related degree of approximation must be evaluated. To the purpose, consider the following expression for $\bar{\tau}$ in (3)

$$\bar{\tau} = \bar{\boldsymbol{P}}\gamma = \sum_{i=1}^{p} \bar{\boldsymbol{p}}_i \gamma_i, \tag{7}$$

where $\bar{\boldsymbol{p}}_i$ is the ith column of $\bar{\boldsymbol{P}}$. Thus, the parameters and the associated columns in $\boldsymbol{P}$ to be eliminated can be selected on the basis of the following test

$$\text{if} \quad \|\bar{\boldsymbol{p}}_i \gamma_i\| \le \bar{p}_\gamma, \quad \text{then discard the parameter} \quad \gamma_i, \tag{8}$$

where $\bar{p}_\gamma$ is a properly set treshold. Notice that the above test cannot be based only on the evaluation of $\|\bar{\boldsymbol{p}}_i\|$, because it does not take into account the numerical value of the corresponding parameter.

Using the data collected along the test trajectory and setting the treshold $\bar{p}_\gamma = 250.0$ in (8), we found that the parameters $\gamma_4, \gamma_6, \gamma_{14}, \gamma_{15}, \gamma_{16}, \gamma_{19}, \gamma_{20}, \gamma_{21}$ and γ_{25} can be eliminated from the model. Thus, the columns of $\boldsymbol{P}$ can be reduced to 17. Notice that the remaining parameters are principal inertias and friction coefficients at each joint. In order to test the reduced model in different conditions, a torque reconstruction is performed on a second test trajectory (not shown here); the results reported in Fig. 4 show that the reduced model is still able to track the system's behaviour.

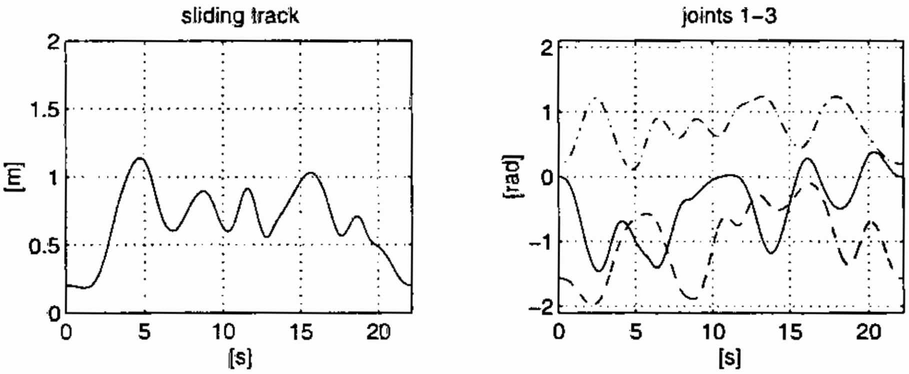


Figure 2. The first test trajectory (positions); *solid*—joint 1, *dashed*—joint 2, *dashdotted*—joint 3.

Conclusions

In this paper the problem of estimation of dynamic parameters for a conventional industrial manipulator has been addressed. By exploiting the property of linearity in the parameters of the dynamic model, a procedure based on a least-squares algorithm is set up to estimate the dynamic parameters. The set of data for the estimation is collected along a trajectory which ensures an optimal degree of excitation for the manipulator dynamics. The experiments have been performed on an industrial manipulator. The conditions of the experiments are far from the ideal ones required by the theory: unmodeled dynamics are present, constraints on the choice of the identification trajectories are imposed by the mechanical structure. In spite of the above factors, the results show a god accuracy of the obtained model. Finally, a simple model reduction technique is presented and experimentally validated.

Acknowledgments This work was supported by *Ministero dell'Università e della Ricerca Scientifica e Tecnologica* within the 40% and 60% funding framework.

References

1. Sciavicco, L. and Siciliano, B., Modeling and control of robot manipulators. McGraw-Hill, New York, NY, 1996.
2. Atkeson, C.G., An, C.H. and Hollerbach, J.M., Estimation of the inertial parameters of manipulator loads and links. In: International Journal of Robotics Research, 5(3), pp. 101-119, 1986.
3. Gautier, M. and Khalil, W., On the identification of the inertial parameters of robots. In: Proc. of the 27th IEEE Conference on Decision and Control, Austin, TX, pp. 2264-2269.

4. Armstrong, B., On finding 'exciting' trajectories for identification experiments involving systems with non-linear dynamics. In: Proc. of the 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, pp. 1131-1138, 1987.

5. Gautier, M., Numerical calculation of the base inertial parameters of robots. In: Proc. of the 1990 IEEE International Conference on Robotics and Automation, Cincinnati, OH, pp. 1020-1025, 1990.

6. Armstrong-Hélouvry, B., Dupont, P. and Canudas De Wit, C., A survey of models, analysis tools and compensation methods for the control of machines with friction. In: Automatica, (30)7, pp. 1083-1138, 1994.

7. Golub, G.H. and Van Loan, C.F., Matrix computations. John Hopkins University Press, Baltimore, MA, 1983.

8. Presse, C. and Gautier, M., New criteria of exciting trajectories for robot identification. In: Proc. of the 1993 IEEE International Conference on Robotics and Automation, Atlanta, GA, pp. 907–912, 1993.

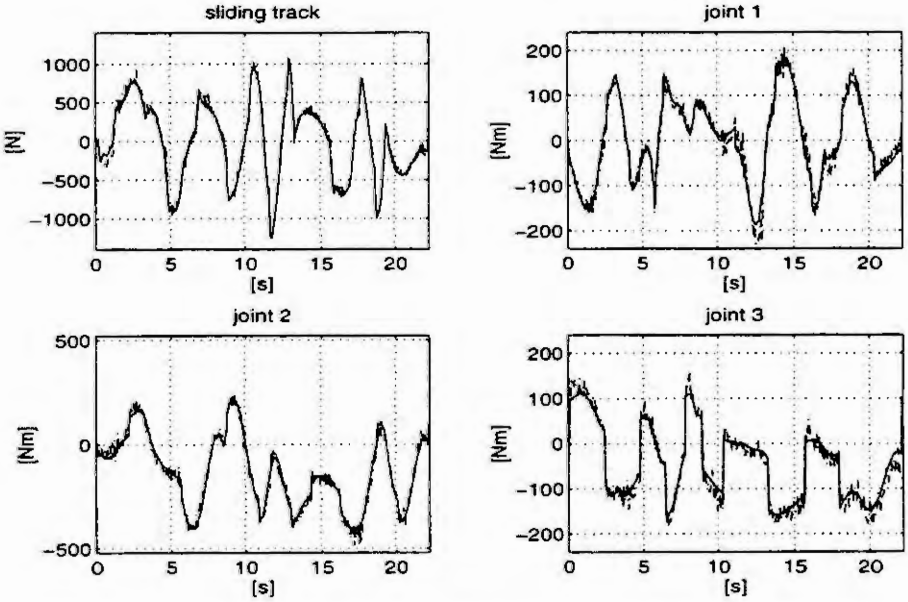


Figure 3. Torque reconstruction along the first test trajectory; *solid*—computed, *dashed*—measured.

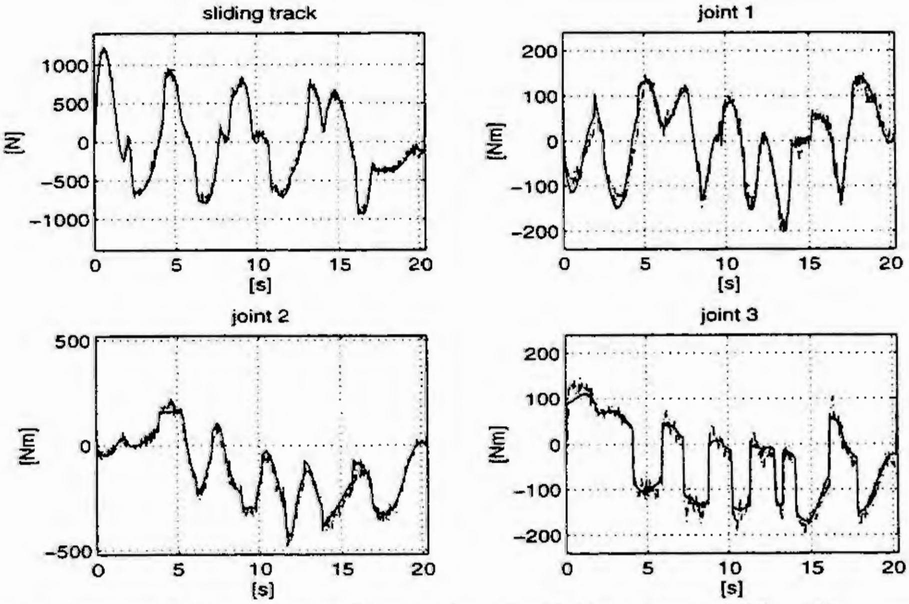


Figure 4. Torque reconstruction along the second test trajectory with the reduced model; *solid*—computed, *dashed*—measured.

SINGULAR PERTURBATION MODEL OF ROBOTS WITH ELASTIC JOINTS AND ELASTIC LINKS CONSTRAINED BY RIGID ENVIRONMENT

P. Rocco
Politecnico di Milano
Dipartimento di Elettronica e Informazione
Piazza Leonardo da Vinci, 32, 20133 Milano - Italy
rocco@elet.polimi.it

Abstract. A robot with distributed flexibility in the links and lumped flexibility in the joints is considered in this paper. First the model of the system in free motion is formulated as a set of ordinary differential equations, adopting a finite number of modes of the link deformation. Then algebraic constraint equations on the generalized coordinates of the system are added, to account for the loss of degrees of freedom due to the contact with rigid environment. A reduced order model, expressed in the residual degrees of freedom is then derived, based on a coordinate partitioning procedure. The singularly perturbed model of the system is finally computed, and the expression of the fast subsystem is given. The special cases of a robot with rigid joints or rigid links are also addressed.

Introduction

Every robot is affected, to some extent, by distributed flexibility of the links [2]. Robots actuated by transmission systems (non direct drive robots) are also affected by the elasticity of the joints connecting the motors to the links. While the elasticity can be neglected in many practical cases (distributed elasticity of the links is usually inessential in massive industrial robots), some applications exist where the resonant dynamics associated with the flexibility can be of prominent importance. Lightweight long-reach manipulators, used in space robotics and in some special applications such as nuclear waste retrieval, are inherently flexible and vibrate when they undergo fast transients; industrial robots equipped with Harmonic Drive reductors exhibit oscillatory behavior usually ascribed to the torsional flexibility of the transmissions. Since the adoption of electrical motors and geared transmission systems is a standard choice in robotics, even outside the industrial applications, it can be of some interest to study the model of a robot affected by flexibility both in the links and in the joints.

Moreover, most of the robotic applications involve a contact of the tip of the manipulator with a usually rigid external environment. Hence, models which explicitly account for the constraints imposed to the robot motion by the environment deserve some attention. This is particularly true in view of the formulation of force/position control algorithms [12], i.e. control systems which can track a desired position of the tip of the robot and simultaneously regulate the forces arising at the contact with the environment.

In any case, the flexibility is revealed by transients which evolve in a much faster time scale than those associated to the nominal operating conditions of the robot. Singular perturbation theory [4] lends itself to the description of systems whose dynamics evolve in two time scales. The goal of a singular perturbation approach is the identification of two reduced order subsystems from the original system: the slow subsystem (sometimes called quasi steady-state system), whose dynamics coincide with the dynamics of the rigid structure, and the fast system, which is a dynamic system parametrized by the variables of the slow one.

Singular perturbation models of elastic joint robots [11], as well as flexible link robots [10] in free motion have long been known in the literature, while only recently correct singular perturbation models for elastic joint robots [8], and flexible link robots [9], constrained by the environment, have been proposed.

In the present paper, a comprehensive singular perturbation model of a constrained robot with flexible joints and links will be presented. The formulation of the model will step through the following points:

-Modelling of the robot in free motion as a set of ordinary differential equations;
-Modelling of the constraint as a set of algebraic equations on the generalized coordinates;
-Expression of the reaction forces as Lagrange multipliers;
-Reduction of the order of the model by way of the coordinate partitioning method;
-Application of the singular perturbation method to the reduced order model;
-Formulation of the slow and fast systems dynamics.

The singularly perturbed models of robots with rigid links or rigid joints will be obtained as particular cases of the general model presented in this work.

Modelling in free motion

The system considered in this study is a robot with revolute single d.o.f. joints whose links are affected by elasticity. We will assume that that the rigid motion and the flexible deformation of the links occur in the same plane. The actuation of the motion is performed by means of motors connected to the robot joints through elastic transmissions. While the elasticity of the transmission can be conveniently lumped by considering a fictitious torsional spring between the motor and the link [11], the elastic link is actually a distributed parameter system, whose exact modelling is accomplished using partial differential equations.

A finite dimensional model of the robot can nevertheless be obtained by truncating the modal expansion of the deflection to a finite number of assumed modes [1], [3], [7] under the assumption of small deformation:

$$w_i(x,t)=\sum_{j=1}^{m_i} q_{fij}(t)\psi_{ij}(x) \tag{1}$$

where w_i is the deflection of link i at time t, computed at a distance x from the origin of a suitable reference frame attached to the link, ψ_{ij} is the shape assumed for the j-th mode of link i, while q_{fij} is its time-varying amplitude. The number of modes retained from the asymptotic expansion is denoted by m_i.

Lagrange's equations of motion of the overall system (the flexible robot, the elastic transmissions and the motors) in free motion can be obtained considering as a set of generalized coordinates, the rigid joint coordinates $\boldsymbol{q}_r \in \Re^n$, the flexible variables $\boldsymbol{q}_f = (q_{f11}, \ldots, q_{f1m_1}; q_{f21}, \ldots, q_{f2m_2}; q_{fn1}, \ldots, q_{fnm_n})^T \in \Re^{N-2n}$ and the motor coordinates $\boldsymbol{q}_m \in \Re^n$:

$$\boldsymbol{M}(\boldsymbol{q}_r,\boldsymbol{q}_f)\begin{bmatrix}\ddot{\boldsymbol{q}}_r\\ \ddot{\boldsymbol{q}}_f\end{bmatrix}+\begin{bmatrix}\boldsymbol{h}_r(\boldsymbol{q}_r,\dot{\boldsymbol{q}}_r,\boldsymbol{q}_f,\dot{\boldsymbol{q}}_f)\\ \boldsymbol{h}_f(\boldsymbol{q}_r,\dot{\boldsymbol{q}}_r,\boldsymbol{q}_f,\dot{\boldsymbol{q}}_f)\end{bmatrix}+\begin{bmatrix}\boldsymbol{g}_r(\boldsymbol{q}_r,\boldsymbol{q}_f)\\ \boldsymbol{g}_f(\boldsymbol{q}_r,\boldsymbol{q}_f)\end{bmatrix}+\begin{bmatrix}\boldsymbol{K}_J(\boldsymbol{q}_m-\boldsymbol{q}_r)\\ \boldsymbol{K}_f\boldsymbol{q}_f\end{bmatrix}=\boldsymbol{0} \tag{2}$$

$$\boldsymbol{J}_m\ddot{\boldsymbol{q}}_m+\boldsymbol{K}_J(\boldsymbol{q}_m-\boldsymbol{q}_r)=\boldsymbol{u}$$

where:

$$\boldsymbol{M}=\begin{bmatrix}\boldsymbol{M}_{rr} & \boldsymbol{M}_{rf}\\ \boldsymbol{M}_{rf}^T & \boldsymbol{M}_{ff}\end{bmatrix}$$

is the symmetric positive definite inertia matrix of the robot, conveniently partitioned into the matrices $\boldsymbol{M}_{rr} \in \Re^{n\times n}$, $\boldsymbol{M}_{rf} \in \Re^{n\times(N-2n)}$, $\boldsymbol{M}_{ff} \in \Re^{(N-2n)\times(N-2n)}$ and $\boldsymbol{M}_{rf}{}^T \in \Re^{(N-2n)\times n}$; $\boldsymbol{h}_r$ and $\boldsymbol{h}_f$ are the vectors of Coriolis and centrifugal terms, while $\boldsymbol{g}_r$ and $\boldsymbol{g}_f$ the vector of gravitational terms (for the rigid and the flexible parts, respectively); $\boldsymbol{K}_J$ and $\boldsymbol{K}_f$ are the matrices (diagonal and positive definite) of the stiffness constants of the joints and of the links, respectively; $\boldsymbol{J}_m$ is the matrix (diagonal and positive definite) of the moments of inertia of the motors; $\boldsymbol{u} \in \Re^n$ is the vector of the control inputs (assuming as many control inputs as rigid d.o.f.).

Constraints

Assume now that the tip of the robot makes contact with a very stiff environment. Algebraic constraint equations, in the same number as the number of d.o.f. inhibited by the interaction with the environment, are introduced in the model. While the constraint equations are more easily written in terms of the Cartesian coordinates of the tip of the robot, we will assume that, with proper use of the direct kinematics of the robot, the constraint equations are written in terms of the above defined rigid joint coordinates $\boldsymbol{q}_r$ and flexible variables $\boldsymbol{q}_f$:

$$\boldsymbol{\Phi}(\boldsymbol{q}_r,\boldsymbol{q}_f)=\boldsymbol{0}, \tag{3}$$

where $\boldsymbol{\Phi}$: $(\Re^n \times \Re^{N-2n}) \rightarrow \Re^m$, m being the number of constraints ($m\leq n$).

Defining now the two Jacobian matrices:

$$\boldsymbol{A}_r=\frac{\partial\boldsymbol{\Phi}}{\partial\boldsymbol{q}_r}, \qquad \boldsymbol{A}_f=\frac{\partial\boldsymbol{\Phi}}{\partial\boldsymbol{q}_f},$$

where $\boldsymbol{A}_r \in \Re^{m\times n}$, $\boldsymbol{A}_f \in \Re^{m\times(N-2n)}$, and recalling that the constraint forces act along the normals to the constraint surfaces, eq.(2) can be rewritten, in case of constrained motion, as:

$$\boldsymbol{M}(\boldsymbol{q}_r,\boldsymbol{q}_f)\begin{bmatrix}\ddot{\boldsymbol{q}}_r\\ \ddot{\boldsymbol{q}}_f\end{bmatrix}+\begin{bmatrix}\boldsymbol{h}_r(\boldsymbol{q}_r,\dot{\boldsymbol{q}}_r,\boldsymbol{q}_f,\dot{\boldsymbol{q}}_f)\\ \boldsymbol{h}_f(\boldsymbol{q}_r,\dot{\boldsymbol{q}}_r,\boldsymbol{q}_f,\dot{\boldsymbol{q}}_f)\end{bmatrix}+\begin{bmatrix}\boldsymbol{g}_r(\boldsymbol{q}_r,\boldsymbol{q}_f)\\ \boldsymbol{g}_f(\boldsymbol{q}_r,\boldsymbol{q}_f)\end{bmatrix}+\begin{bmatrix}\boldsymbol{K}_J(\boldsymbol{q}_m-\boldsymbol{q}_r)\\ \boldsymbol{K}_f\boldsymbol{q}_f\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}_r^T(\boldsymbol{q}_r,\boldsymbol{q}_f)\\ \boldsymbol{A}_f^T(\boldsymbol{q}_r,\boldsymbol{q}_f)\end{bmatrix}\boldsymbol{\lambda} \tag{4}$$

$$\boldsymbol{J}_m\ddot{\boldsymbol{q}}_m+\boldsymbol{K}_J(\boldsymbol{q}_m-\boldsymbol{q}_r)=\boldsymbol{u}$$

where $\boldsymbol{\lambda} \in \Re^m$ is a vector of Lagrange multipliers.

Introducing the inverse of the inertia matrix:

$$\boldsymbol{H} = \boldsymbol{M}^{-1} = \begin{bmatrix} \boldsymbol{H}_{rr} & \boldsymbol{H}_{rf} \\ \boldsymbol{H}_{fr} & \boldsymbol{H}_{ff} \end{bmatrix}$$

where $\boldsymbol{H}_{rr}$, $\boldsymbol{H}_{rf}$, $\boldsymbol{H}_{ff}$ and $\boldsymbol{H}_{fr} = \boldsymbol{H}_{rf}^T$ have the same dimensions as $\boldsymbol{M}_{rr}$, $\boldsymbol{M}_{rf}$, $\boldsymbol{M}_{ff}$ and $\boldsymbol{M}_{rf}^T$, respectively, the equations of the motors are expressed in terms of the new variables:

$$\boldsymbol{q}_J = \boldsymbol{q}_m - \boldsymbol{q}_r$$

as:

$$\ddot{\boldsymbol{q}}_J = -\left(\boldsymbol{J}_m^{-1} + \boldsymbol{H}_{rr}\right)\boldsymbol{K}_J \boldsymbol{q}_J + \boldsymbol{H}_{rr}\left(\boldsymbol{h}_r + \boldsymbol{g}_r\right) + \boldsymbol{H}_{rf}\left(\boldsymbol{h}_f + \boldsymbol{g}_f + \boldsymbol{K}_f \boldsymbol{q}_f\right) + \boldsymbol{J}_m^{-1}\boldsymbol{u} - \left(\boldsymbol{H}_{rr}\boldsymbol{A}_r^T + \boldsymbol{H}_{rf}\boldsymbol{A}_f^T\right)\boldsymbol{\lambda} \qquad (5)$$

Model reduction

The mathematical model (4) is made up by N second-order differential equations, for a system that actually presents $N-m$ d.o.f., once the constraints (3) are active. It is however possible to reduce the number of differential equations, by resorting to a coordinate partitioning procedure [5], [6], [13]. Consider the following partition of the vector $\boldsymbol{q}_r$:

$$\boldsymbol{q}_r = \begin{bmatrix} \boldsymbol{q}_{r1} \\ \boldsymbol{q}_{r2} \end{bmatrix}, \qquad (6)$$

where $\boldsymbol{q}_{r1} \in \Re^m$, $\boldsymbol{q}_{r2} \in \Re^{n-m}$, and assume that there exist a continuous, twice differentiable function Ω: $(\Theta_{r2} \times \Theta_f) \to \Re^m$, where Θ_{r2} and Θ_f are two open sets ($\Theta_{r2} \subset \Re^{n-m}$, $\Theta_f \subset \Re^{N-2n}$), such that the constraints (3) can be expressed as:

$$\boldsymbol{q}_{r1} = \Omega\left(\boldsymbol{q}_{r2}, \boldsymbol{q}_f\right). \qquad (7)$$

A reordering of the rigid variables $\boldsymbol{q}_r$ could be necessary to express the constraints as in (7). Also, note that the dependent variables $\boldsymbol{q}_{r1}$ have been chosen only among the rigid ones, thus implicitly excluding the presence of a constraint acting only on the flexible variables.

Differentiating (6) with respect to time, we have:

$$\dot{\boldsymbol{q}}_r = \boldsymbol{T}_{rr}\dot{\boldsymbol{q}}_{r2} + \boldsymbol{T}_{rf}\dot{\boldsymbol{q}}_f \qquad (8)$$

where:

$$\boldsymbol{T}_{rr} = \begin{bmatrix} \dfrac{\partial\Omega}{\partial \boldsymbol{q}_{r2}} \\ \boldsymbol{I}_{n-m} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{A}_{r1}^{-1}\boldsymbol{A}_{r2} \\ \boldsymbol{I}_{n-m} \end{bmatrix}, \quad \boldsymbol{T}_{rf} = \begin{bmatrix} \dfrac{\partial\Omega}{\partial \boldsymbol{q}_f} \\ \boldsymbol{0}_{n-m,N-2n} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{A}_{r1}^{-1}\boldsymbol{A}_f \\ \boldsymbol{0}_{n-m,N-2n} \end{bmatrix}$$

and the Jacobian matrix $\boldsymbol{A}_r$ has been partitioned as:

$$\boldsymbol{A}_r = \begin{bmatrix} \boldsymbol{A}_{r1} & \boldsymbol{A}_{r2} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial\Phi}{\partial \boldsymbol{q}_{r1}} & \dfrac{\partial\Phi}{\partial \boldsymbol{q}_{r2}} \end{bmatrix}.$$

Introducing matrix $\boldsymbol{T}$:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{rr} & \boldsymbol{T}_{rf} \\ \boldsymbol{0}_{N-2n,n-m} & \boldsymbol{I}_{N-2n} \end{bmatrix},$$

which has the property that:

$$\begin{bmatrix} \boldsymbol{A}_r & \boldsymbol{A}_f \end{bmatrix}\boldsymbol{T} \equiv \boldsymbol{0}, \qquad (9)$$

it is possible to eliminate the Lagrange multipliers $\boldsymbol{\lambda}$ from the dynamic equations (4) by premultiplying the said equations by matrix $\boldsymbol{T}^T$. Exploiting eq. (8) and its derivative, we finally arrive at the expression of the constrained dynamic system in terms of the independent variables $\boldsymbol{q}_{r2}$ and $\boldsymbol{q}_f$:

$$\boldsymbol{M}_c\left(\boldsymbol{q}_{r2}, \boldsymbol{q}_f\right)\begin{bmatrix} \ddot{\boldsymbol{q}}_{r2} \\ \ddot{\boldsymbol{q}}_f \end{bmatrix} + \begin{bmatrix} \boldsymbol{h}_{cr}\left(\boldsymbol{q}_{r2}, \dot{\boldsymbol{q}}_{r2}, \boldsymbol{q}_f, \dot{\boldsymbol{q}}_f\right) \\ \boldsymbol{h}_{cf}\left(\boldsymbol{q}_{r2}, \dot{\boldsymbol{q}}_{r2}, \boldsymbol{q}_f, \dot{\boldsymbol{q}}_f\right) \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}_{cr}\left(\boldsymbol{q}_{r2}, \boldsymbol{q}_f\right) \\ \boldsymbol{g}_{cf}\left(\boldsymbol{q}_{r2}, \boldsymbol{q}_f\right) \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{n-m,1} \\ \boldsymbol{K}_f \boldsymbol{q}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_{rr}^T\left(\boldsymbol{q}_{r2}, \boldsymbol{q}_f\right)\boldsymbol{K}_J \boldsymbol{q}_J \\ \boldsymbol{T}_{rf}^T\left(\boldsymbol{q}_{r2}, \boldsymbol{q}_f\right)\boldsymbol{K}_J \boldsymbol{q}_J \end{bmatrix} \qquad (10)$$

where:

$$\boldsymbol{M}_c = \begin{bmatrix} \boldsymbol{M}_{crr} & \boldsymbol{M}_{crf} \\ \boldsymbol{M}_{crf}^T & \boldsymbol{M}_{cff} \end{bmatrix}$$

and

$$M_{crr} = T_{rr}^T M_{rr} T_{rr}, \quad M_{crf} = T_{rr}^T M_{rr} T_{rf} + T_{rr}^T M_{rf}, \quad M_{cff} = M_{ff} + T_{rf}^T M_{rr} T_{rf} + T_{rf}^T M_{rf} + M_{rf}^T T_{rf} \ ,$$

$$h_{cr} = T_{rr}^T h_r + T_{rr}^T M_{rr} \dot{T}_{rr} \dot{q}_{r2} + T_{rr}^T M_{rr} \dot{T}_{rf} \dot{q}_f, \quad h_{cf} = T_{rf}^T h_r + h_f + T_{rf}^T M_{rr} \dot{T}_{rr} \dot{q}_{r2} + M_{rf} \dot{T}_{rr} \dot{q}_{r2} + T_{rf}^T M_{rr} \dot{T}_{rf} \dot{q}_f$$

$$g_{cr} = T_{rr}^T g_r, \quad g_{cf} = T_{rf}^T g_r + g_f$$

An expression for the Lagrange multipliers λ in terms of the state variables can be obtained by twice differentiating the constraint equations (3) [5]:

$$A_r \ddot{q}_r + \dot{A}_r \dot{q}_r + A_f \ddot{q}_f + \dot{A}_f \dot{q}_f = 0 \ ,$$

and eliminating the vector of the acceleration from eq. (4). The result is:

$$\lambda = \left(AM^{-1}A^T\right)^{-1}\left(-\dot{A}\begin{bmatrix} T_{rr}\dot{q}_{r2} + T_{rf}\dot{q}_f \\ \dot{q}_f \end{bmatrix} + AM^{-1}\left(\begin{bmatrix} h_r \\ h_f \end{bmatrix} + \begin{bmatrix} g_r \\ g_f \end{bmatrix} + \begin{bmatrix} 0_{n.1} \\ K_f q_f \end{bmatrix} - \begin{bmatrix} K_J q_J \\ 0_{N-2n.1} \end{bmatrix}\right)\right) \ , \tag{11}$$

where $A=[A_r \ \ A_f]$. Plugging this expression into (5) yields:

$$\begin{gathered} \ddot{q}_J = -\left(J_m^{-1} + H_{rr}\right)K_J q_J + H_{rr}(h_r + g_r) + H_{rf}\left(h_f + g_f + K_f q_f\right) + J_m^{-1}u - \left(H_{rr}A_r^T + H_{rf}A_f^T\right)\left(AM^{-1}A^T\right)^{-1} \\ \left(-\dot{A}_r\left(T_{rr}\dot{q}_{r2} + T_{rf}\dot{q}_f\right) - \dot{A}_f\dot{q}_f + \left(A_r H_{rr} + A_f H_{fr}\right)\left(h_r + g_r - K_J q_J\right) + \left(A_r H_{rf} + A_f H_{ff}\right)\left(h_f + g_f + K_f q_f\right)\right) \end{gathered} \tag{12}$$

which, together with (10), completes the model of the constrained robot.

A singularly perturbed version of the model

The singular perturbation parameter [4] is introduced as μ = $1/k$, where k is the minimum between k_f and k_J, which in turn are common factors among the stiffness constants of the arm (elements of matrix K_f), and among the stiffness constants of the joints (elements of matrix K_J), respectively. New variables are then defined as:

$$\zeta_f = K_f q_f = k_f \hat{K}_f q_f = \alpha_f k \hat{K}_f q_f$$

$$\zeta_J = K_J q_J = k_J \hat{K}_J q_J = \alpha_J k \hat{K}_J q_J$$

with $\alpha_f = k_f/k$ and $\alpha_J = k_J/k$. Defining the inverse of the inertia matrix of the constrained system as:

$$H_c = M_c^{-1} = \begin{bmatrix} H_{crr} & H_{crf} \\ H_{cfr} & H_{cff} \end{bmatrix} \ ,$$

where H_{crr}, H_{crf}, H_{cff} and $H_{cfr} = H_{crf}^T$ have the same dimensions as M_{crr}, M_{crf}, M_{cff} and M_{crf}^T respectively, it is possible to rewrite system (10), (12) in the following singularly perturbed form:

$$\ddot{q}_{r2} = -H_{crr}\left[h_{cr} + g_{cr}\right] - H_{crf}\left[h_{cf} + g_{cf} + \zeta_f\right] + \left[H_{crr}T_{rr}^T + H_{crf}T_{rf}^T\right]\zeta_J \tag{13}$$

$$\frac{\mu\ddot{\zeta}_f}{\alpha_f} = -\hat{H}_{cfr}\left[h_{cr} + g_{cr}\right] - \hat{H}_{cff}\left[h_{cf} + g_{cf} + \zeta_f\right] + \left[\hat{H}_{cfr}T_{rr}^T + \hat{H}_{cff}T_{rf}^T\right]\zeta_J \tag{14}$$

$$\begin{gathered} \frac{\mu\ddot{\zeta}_J}{\alpha_J} = -\left(\hat{J}_m^{-1} + \hat{H}_{rr}\right)\zeta_J + \hat{H}_{rr}(h_r + g_r) + \hat{H}_{rf}\left(h_f + g_f + \zeta_f\right) + \hat{J}_m^{-1}u - \left(\hat{H}_{rr}A_r^T + \hat{H}_{rf}A_f^T\right)\left(AM^{-1}A^T\right)^{-1} \\ \left(-\dot{A}_r\left(T_{rr}\dot{q}_{r2} + T_{rf}\mu\left(\alpha_f\hat{K}_f\right)^{-1}\dot{\zeta}_f\right) - \dot{A}_f\mu\left(\alpha_f\hat{K}_f\right)^{-1}\dot{\zeta}_f + \right. \\ \left. \left(A_r H_{rr} + A_f H_{fr}\right)\left(h_r + g_r - \zeta_J\right) + \left(A_r H_{rf} + A_f H_{ff}\right)\left(h_f + g_f + \zeta_f\right)\right) \end{gathered} \tag{15}$$

where $\hat{H}_{cfr} = \hat{K}_f H_{cfr}$, $\hat{H}_{cff} = \hat{K}_f H_{cff}$, $\hat{H}_{rr} = \hat{K}_J H_{rr}$, $\hat{H}_{rf} = \hat{K}_J H_{rf}$, $\hat{J}_m^{-1} = \hat{K}_J J_m^{-1}$.

In the limit as μ→0, eqs. (14) and (15) collapse to the following algebraic equations:

$$0 = -\bar{\hat{H}}_{cfr}\left[\bar{h}_{cr} + \bar{g}_{cr}\right] - \bar{\hat{H}}_{cff}\left[\bar{h}_{cf} + \bar{g}_{cf} + \bar{\zeta}_f\right] + \left[\bar{\hat{H}}_{cfr}\bar{T}_{rr}^T + \bar{\hat{H}}_{cff}\bar{T}_{rf}^T\right]\bar{\zeta}_J \tag{16}$$

$$\begin{gathered} 0 = -\left(\hat{J}_m^{-1} + \bar{\hat{H}}_{rr}\right)\bar{\zeta}_J + \bar{\hat{H}}_{rr}\left(\bar{h}_r + \bar{g}_r\right) + \bar{\hat{H}}_{rf}\left(\bar{h}_f + \bar{g}_f + \bar{\zeta}_f\right) + \hat{J}_m^{-1}\bar{u} - \left(\bar{\hat{H}}_{rr}\bar{A}_r^T + \bar{\hat{H}}_{rf}\bar{A}_f^T\right)\left(\bar{A}\bar{M}^{-1}\bar{A}^T\right)^{-1} \\ \left(-\bar{\dot{A}}_r\bar{T}_{rr}\bar{\dot{q}}_{r2} + \left(\bar{A}_r\bar{H}_{rr} + \bar{A}_f\bar{H}_{fr}\right)\left(\bar{h}_r + \bar{g}_r - \bar{\zeta}_J\right) + \left(\bar{A}_r\bar{H}_{rf} + \bar{A}_f\bar{H}_{ff}\right)\left(\bar{h}_f + \bar{g}_f + \bar{\zeta}_f\right)\right) \end{gathered} \ . \tag{17}$$

In eqs. (16), (17), the overbars denote that the variables, or the matrices, are evaluated in the special case μ=0: for example, $\bar{h}_{cr}$ stands for $h_{cr}\left(\bar{q}_{r2}, \bar{\dot{q}}_{r2}, 0, 0\right)$. The system is linear in $\bar{\zeta}_f$ and $\bar{\zeta}_J$: by solving for these variables and plugging the result into (13), with μ=0, the equations of the rigid robot model are obtained, as it can be proven. Thus the slow dynamics of the system are identified as the dynamics of the rigid system.

To reveal the fast dynamics, we first introduce the fast variables:

$$\eta_1 = \zeta_f - \overline{\zeta}_f, \quad \eta_2 = \varepsilon\dot{\zeta}_f$$

$$\eta_3 = \zeta_J - \overline{\zeta}_J, \quad \eta_4 = \varepsilon\dot{\zeta}_J$$

where $\varepsilon = \sqrt{\mu}$, and the fast time scale $\tau = t/\varepsilon$. Rewriting the system in this time scale, and examining it for $\varepsilon=0$, it is easy to conclude that system (13) confirms that $\boldsymbol{q}_{r2}$ and $\dot{\boldsymbol{q}}_{r2}$ are constant in the fast time scale, while the expression of the fast dynamics can be obtained by combining eqs. (14), (15), (16) and (17). The result is:

$$\begin{aligned}\frac{d\eta_1}{d\tau} &= \eta_2\\ \frac{d\eta_2}{d\tau} &= \alpha_f\left(-\overline{\hat{\boldsymbol{H}}}_{cff}\eta_1 + \left(\overline{\hat{\boldsymbol{H}}}_{cfr}\overline{\boldsymbol{T}}_{rr}^T + \overline{\hat{\boldsymbol{H}}}_{cff}\overline{\boldsymbol{T}}_{rf}^T\right)\eta_3\right)\end{aligned} \tag{18}$$

$$\begin{aligned}\frac{d\eta_3}{d\tau} &= \eta_4\\ \frac{d\eta_4}{d\tau} &= \alpha_J\left(\left(\overline{\hat{\boldsymbol{H}}}_{rf} - \overline{\boldsymbol{L}}\left(\overline{\boldsymbol{A}}_r\overline{\boldsymbol{H}}_{rf} + \overline{\boldsymbol{A}}_f\overline{\boldsymbol{H}}_{ff}\right)\right)\eta_1 - \left(\hat{\boldsymbol{J}}_m^{-1} + \overline{\hat{\boldsymbol{H}}}_{rr} - \overline{\boldsymbol{L}}\left(\overline{\boldsymbol{A}}_r\overline{\boldsymbol{H}}_{rr} + \overline{\boldsymbol{A}}_f\overline{\boldsymbol{H}}_{fr}\right)\right)\eta_3 + \hat{\boldsymbol{J}}_m^{-1}(\boldsymbol{u} - \overline{\boldsymbol{u}})\right)\end{aligned} \tag{19}$$

where:

$$\overline{\boldsymbol{L}} = \left(\overline{\hat{\boldsymbol{H}}}_{rr}\overline{\boldsymbol{A}}_r^T + \overline{\hat{\boldsymbol{H}}}_{rf}\overline{\boldsymbol{A}}_f^T\right)\left(\overline{\boldsymbol{A}}\,\overline{\boldsymbol{M}}^{-1}\overline{\boldsymbol{A}}^T\right)^{-1} .$$

The fast system is linear in the state variables η_1, η_2, η_3, η_4 , parametrized in the values of the variables of the slow system.

Robot with rigid joints and elastic links

This special case corresponds to:

$$\alpha_J = \infty, \quad \alpha_f = 1 \ .$$

The second of (19) thus collapses to an algebraic equation. Solving this equation for η_3, plugging the result in the first of (18), and performing messy calculations (here omitted), the following result is obtained:

$$\begin{aligned}\frac{d\eta_1}{d\tau} &= \eta_2\\ \frac{d\eta_2}{d\tau} &= -\overline{\hat{\boldsymbol{H}}}_{cff}\eta_1 + \left(\overline{\hat{\boldsymbol{H}}}_{cfr}\overline{\boldsymbol{T}}_{rr}^T + \overline{\hat{\boldsymbol{H}}}_{cff}\overline{\boldsymbol{T}}_{rf}^T\right)(\boldsymbol{u} - \overline{\boldsymbol{u}})\end{aligned}$$

where matrices $\hat{\boldsymbol{H}}_{cff}$ and $\hat{\boldsymbol{H}}_{cfr}$ account also for the effect of the inertia matrix of the motors (which sums up to the rigid part of the inertia matrix of the robot, $\boldsymbol{M}_{rr}$).

Robot with elastic joints and rigid links

This second special case can be identified by letting:

$$\alpha_J = 1, \quad \alpha_f = \infty \ .$$

The second of (18) thus collapses to an algebraic equation. Solving this equation for η_1, plugging the result into the first of (19) and performing again messy calculations (here omitted), the following result is obtained:

$$\begin{aligned}\frac{d\eta_3}{d\tau} &= \eta_4\\ \frac{d\eta_4}{d\tau} &= -\left(\hat{\boldsymbol{J}}_m^{-1} + \overline{\hat{\boldsymbol{M}}}_{rr}^{-1} - \overline{\boldsymbol{F}}\right)\eta_3 + \hat{\boldsymbol{J}}_m^{-1}(\boldsymbol{u} - \overline{\boldsymbol{u}})\end{aligned}$$

where:

$$\overline{\boldsymbol{F}} = \overline{\boldsymbol{M}}_{rr}^{-1}\overline{\boldsymbol{A}}_r^T\left(\overline{\boldsymbol{A}}_r\overline{\boldsymbol{M}}_{rr}^{-1}\overline{\boldsymbol{A}}_r^T\right)^{-1}\overline{\boldsymbol{A}}_r\overline{\boldsymbol{M}}_{rr}^{-1}$$

Conclusions

The aim of this paper was substantially methodological. While the model of the robot with elastic joints and elastic links might be of limited interest in practical situations, the systematic approach used in this work could be of some help for readers interested in the two time scale modelling of constrained mechanical systems with flexible parts. Once the contact has been modelled as infinitely stiff and the distributed nature of the flexibility has been removed by some lumping techniques, the very general problem of the singularly perturbed formulation of a DAE (Differential Algebraic Equations) system arises. Attention must be paid to perform the singular perturbation decomposition of the system only after the model reduction, i.e. on the model free of the Lagrange multipliers.

The expression obtained for the dynamics of the fast system is rather involved and does not lend itself to an easy adoption in model based control laws. However, since the system is linear, the said expression should be useful to find bounds on the matrices of the linear system, to be adopted in robust control laws.

References

1. Book, W. J., Recursive Lagrangian dynamics of flexible manipulator arms, International Journal of Robotics Research, 3 (1984), 87-101.
2. Book, W. J., Controlled motion in an elastic world, ASME Journal of Dynamic Systems, Measurements and Control, 115 (1993), 252-261.
3. Cetinkunt, S. and Book, W. J., Symbolic modeling and dynamic simulation of robotic manipulators with compliant links and joints, Robotics and Computer-Integrated Manufacturing, 5 (1989), 301-310.
4. Kokotovic, P., Khalil, H.K. and O'Reilly, J., Singular Perturbation Methods in Control: Analysis and Design, Academic Press, 1986.
5. Jankowski, K. P. and ElMaraghy, H. A., Dynamic decoupling for hybrid control of rigid-/flexible-joint robots interacting with the environment, IEEE Transactions on Robotics and Automation, 8 (1992), 519-534.
6. McClamroch, N. H. and Wang, D., Feedback stabilization and tracking of constrained robots", IEEE Transactions on Automatic Control, 33 (1988), 419-426.
7. Meirovitch, L., Elements of Vibration Analysis, McGraw-Hill, 1986.
8. Rocco, P., "On 'Stability and control of elastic-joint robotic manipulators during constrained-motion tasks'" IEEE Transactions on Robotics and Automation, to appear (1996).
9. Rocco, P. and Book, W.J., Modelling for two-time scale force/position control of flexible robots, In: Proc. IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 1996, 1941-1946.
10. Siciliano, B. and Book, W.J, A singular perturbation approach to control of lightweight flexible manipulators, International Journal of Robotics Research, 7 (1988), 79-90.
11. Spong, M.W., Modelling and control of elastic joint robots. ASME Journal of Dynamic Systems, Measurements and Control, 109 (1987), 310-319.
12. Vukobratovich, M. and Surdilovic, D., Control of robotic systems in contact tasks: an overview, in: Tutorial on Force and Contact Control in Robotic Systems, IEEE International Conference on Robotics and Automation, 13-32, (1993).
13. Wehage, R. A. and Haug, E. J., Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems, ASME Journal of Mechanical Design, 104 (1982), 247-255.

PATH PLANNING FOR ROBOTS BY STOCHASTIC OPTIMIZATION METHODS

K. Marti
Federal Armed Forces University
Aero Space Engineering and Technology
D-85577 Neubiberg/München

Abstract.

In order to reduce large online measurement and correction expenses, the a priori information (given by certain moments or parameters of a probability distribution) on the random variations $p=p(\omega)$ of the vector p of model parameters are taken into account already in the planning phase. Thus, instead of solving a deterministic path planning problem with fixed estimated data, here, the optimal velocity profile along the given trajectory is determined by using a stochastic programming approach. Consequently, the Polygon V(s,p) of Constrained Motion is replaced by a more general Set V(s) of Constrained Motion, determined by chance constraints or more general expected cost constraints. The properties of V(s) are considered for several probability distributions of $p(\omega)$.

1. Optimal Path Planning for Robots

According to [1],[2],[5] the dynamic equation for robots is given by

$$\sum_{j=1}^{n} J_{ij}(q(t),p_D)\ddot{q}_j(t) + \sum_{j,k=1}^{n} C_{ijk}(q(t),p_D)\dot{q}_j(t)\dot{q}_k(t) \tag{1}$$

$$+F_{Ri}(q(t),\dot{q}(t),p_D) + G_i(q(t),p_D) = u_i(t), i=1,\dots,n,$$

where q=q(t) is the n-vector of robot (or configuration) coordinates, $J(q)=(J_{ij}(q))_{1\leq i,j\leq n}$ is the inertia matrix, $C(q)=(C_{ijk}(q))_{1\leq i,j,k\leq n}$ is the tensor of Coriolis and centripetal torques/forces, and $F_{Ri}(q,\dot{q})$; $G_i(q)$ denote the frictional, gravitational forces, respectively.

Moreover, p_D designates the vector of robot or model parameters arising in the dynamic equation (1). Having then a geometric path in work space

$$x_e=x_e(s)\ 0\leq s\leq s_e, \tag{2}$$

to be followed by the end-effector of the robot, in case of non-redundant robots, the kinematic equation

$$T(q,p_T) = x_e(s),\ 0\leq s\leq s_e, \tag{3a}$$

can be solved uniquely for the vector of robot coordinates

$$q = q_e(s,p_T),\ 0\leq s\leq s_e. \tag{3b}$$

The trajectory q=q(t) in the configuration space is then represented by

$$q(t) := q_e(s(t)),\ q_e(s)=q_e(s,p_T), \tag{4}$$

where s=s(t) is a strictly monotonous relationship between time t, $0\leq t\leq t_e$, and the path parameter s, $0\leq s\leq s_e$, to be determined in some optimal sense [1],[2],[5]. Hence, putting

$$\dot{q}(t) = q_e^{/}(s)\dot{s},\ \ddot{q}(t) = q_e^{/}(s)\ddot{s} + q_e^{//}(s)\dot{s}^2, \tag{5}$$

where $\dot{q} := \frac{dq}{dt}$, $q_e^{/} := \frac{dq_e}{ds}$, into (1), we find

$$\sum_{j=1}^{n} J_{ij}(q_e(s,p_T),p_D)(q_{ej}^{/}(s,p_K)\ddot{s} + q_{ej}^{//}(s,p_T)\dot{s}^2) \tag{6a}$$

$$+\sum_{j,k=1}^{n} C_{ijk}(q_e(s,p_T),p_D)q_{ej}^{/}(s,p_T)q_{ek}^{/}(s,p_T)\dot{s}^2 + F_{Ri}(q(s),q_e^{/}(s,p_T),p_D) + G_1(q_e(s,p_T),p_D) = u_i(t), 1 \le i \le n.$$

For $\mathrm{F_{Ri}}$ we suppose that

$$F_{Ri} = \sum_{j=1}^{n} R_{ij}(q_e(s,p_T),p_D)q_{ej}^{/}(s,p_T)\dot{s},\ or\ F_{Ri} = R_i(q_e(s,p_T),p_D)sgn(q_{ei}^{/}(s,p_T)). \tag*{(6b),(6c)}$$

Consequently, for the unknown, strictly monotonous increasing functions $s=s(t)$, $0 \le t \le t_e$, we find with $p_I := (p_T^{/}p_D^{/})$

$$a_i(s,p_I)\ddot{s} + b_i(s,p_I)\dot{s}^2 + b_{iI}(s,p_I)\dot{s} + c_i(s,p_I) = u_i(t),\ 1 \le i \le n, \tag{7a}$$

where

$$a_i(s,p_I) := \sum_{j=1}^{n} J_{ij}(q_e(s,p_T)p_D)q_{ej}^{/}(s,p_T), \tag{7b}$$

$$b_i(s,p_I) := \sum_{j=1}^{n} J_{ij}(q_e(\dot{s},p_T),p_D)q_{ej}^{//}(s,p_T) + \sum_{j=1}^{n} C_{ijk}(q_e(s,p_T),q_{ej}^{/}(s,p_T)q_{ek}^{/}(s,p_T). \tag{7c}$$

Furthermore, in case (6b) we have that

$$b_{iI}(s,p_I := \sum_{j=1}^{n} R_{ij}(q_e(s),p_T),q_{ej}^{/}(s,p_T),\ c_i(s,p_I):\ = G_i(q_e(s,p_T),p_D), \tag{7e}$$

and in case (6c) we get

$$b_{iI}(s,p_I) \equiv 0,\ c_i(s,p_I) := R_i(q_e(s,p_T),p_D)\ sgn\ (q_{ei}^{/}(s,p_T)) + G_i(q_e(s,p_T)). \tag{7f}$$

Introducing now the unknown velocity profile β(s) along the trajectory $x_e=x_e(s)$ by

$$\beta(s) := \dot{s}^2(s) = \dot{s}^2(t(s)), \tag{8}$$

where t=t(s), 0≤s≤s$_e$, denotes the inverse s=s(t), 0≤t≤t$_e$, we have $\ddot{s} = \frac{1}{2}\beta'(s) = \frac{1}{2}\frac{d\beta}{ds}(s)$.Thus, equations (7a-f) read

$$\frac{1}{2}a_i(s,p_I)\beta'(s) + b_i(s,p_I)\beta(s) + b_{ij})s,p_I)\sqrt{\beta(s)} + c_i(s,p_I) = u_i(t),\ 1 \le i \le n. \tag{9}$$

Besides the dynamic and kinematic equation (1),(3a), resp., an optimal control of a robust must satisfy still the following constraints [1],[2],[5]: i) Initial and terminal conditions

$$\beta(0) = \beta(s_e) = 0 \tag{10a}$$

ii) Control constraints

$$u_{\min}(q(t),p_C) + \sum_{j=1}^{n} u_{ij min}(q(t),p_C)\dot{q}_j(t) \leq u_i(t) \tag{10b}$$

$$\leq u_{imax}(q(t),p_C) + \sum_{j=1}^{n} u_{ijmax}(q(t),p_C)\dot{q}_i(t),\ i=1,\dots,n,$$

where p_C is a parameter vector describing the possible uncertainties in the bounds (10b).

iii) Velocity constraints

iii1) Conditions for the joint velocities

$$\dot{q}_{imin}(q(t),p_C) \leq \dot{q}(t) \leq \dot{q}_{imax}(q(t),p_C),\ i=1,\dots,n \tag{10c}$$

iii2) Conditions for the path velocity

$$\mathrm{v}_{imin}(x(t),p_C) \leq \dot{x}_i(t) \leq \mathrm{v}_{imax}(x(t),p_C),\ i=1,\dots,n. \tag{10d}$$

Acording to the representation (4), conditions (10b),(10c),(10d) read:

$$u_{imin}(q_e(s_i(p_k),p_C) + \sum_{j=1}^{n} u_{ijmin}(q_e(s,p_K),p_C)q_{ej}^{\prime}(s,p_K)\sqrt{\beta(s)} \tag{10b'}$$

$$\leq u_i(t) \leq u_{imax}(q_e(s,p_K),p_C) + \sum_{j=1}^{n} u_{ijmax}(q_e(s,p_K),p_C)q_{ej}^{\prime}(s,p_K)\sqrt{\beta(s)},$$

$$\dot{q}_{imin}(q_e(s,p_K),p_C) \leq q_{ei}^{\prime}(s,p_K)\sqrt{\beta(s)} \leq \dot{q}_{imax}(q_e(s,p_K),p_C), \tag{10c'}$$

$$\mathrm{v}_{imin}(x_e(s),p_C) \leq x_{ei}^{\prime}(s)\sqrt{\beta(s)} \leq \mathrm{v}_{imax}(x_e(s),p_C), \tag{10d'}$$

for i=1,2,...,n. Obviously, conditions (10c'),(10d') yield

$$\beta_{\min}(s,p_T,p_C) \leq\beta(s) \leq \beta_{\max}(s,p_T,p_C). \tag{11}$$

Using (9), conditions (10b') and (11) can be represented by the system of linear inequalities

$$\begin{aligned}&(A(s,p_I(\omega)) - B^{u}(s,p_T(\omega),p_C(\omega))\xi(s) \leq h^{u}(s,p(\omega))\\ &-(A(s,p_I(\omega))B^{\ell}(s,p_T(\omega),p_C(\omega))\xi(s) \leq -h^{\ell}(s,p(\omega)),\end{aligned} \tag{12}$$

for the vector

$$\xi = \xi(s) := \begin{pmatrix} \beta(s) \\ \sqrt{\beta(s)} \\ \beta^{\prime}(s) \end{pmatrix}. \tag{13}$$

where $p:= (p_I,p_C)$. (13b)

The objective of the optimal trajectory planning problem is then to determine the velocity profile $\beta=\beta(s)$, $0\leq s\leq s_0$, such that the cost function

$$J(\beta(\cdot)):= \int_0^{s_e} \frac{1}{\sqrt{\beta(s)}} f(s,\beta(s),\beta'(s))ds \tag{14}$$

is minimized subject to the constraints (10a), (12).

2. Stochastic Optimization of Robots

A basic problem in the optimal trajectory planning problem is that the component of the vector p of model parameters in (13b) are **not** given fixed quantities. Due to stochastic variations of the material, manufacturing errors, measurement(identification) errors, stochastic uncertainty of the path in work space and of the payload, p must be described by a random vector

$p=p(\omega),\ \omega\in(\Omega,A,P)$

on a certain probability space (Ω,A,P). Hence, the optimal trajectory planning problem

$$\min J(\beta(\cdot)) \tag{15a}$$

s.t.

$$\begin{pmatrix} A(s,p_I)-B^u(s,p_T,p_C) \\ -A(s,p_I)+B^l(s,p_T,p_C) \end{pmatrix} \begin{pmatrix} \beta(s) \\ \sqrt{\beta(s)} \\ \beta'(s) \end{pmatrix} \leq \begin{pmatrix} h^u(s,p) \\ -h^l(s,p) \end{pmatrix} \tag{15b}$$

$$\beta(s)\geq 0,\ \beta(0) = \beta(s_e) = 0 \qquad (15c),(15d)$$

with the random data $p=p(\omega)$ must be treated by means of **Stochastic Optimization Methods** [3],[4]: The main task is to convert now the random constraint (15b) into one or several deterministic conditions to be treated then by appropriate analytical or numerical methods.

2.1. Probabilistic constraints

Considering the probability that the inequalities in (15b) hold jointly, separately, resp., for each path point s, $0\leq s\leq s_e$, we obtain the probability functions of the following type

$$P_s(\xi):= \left(\begin{pmatrix} A(s,p_I(\omega)) \ - \ B^u(s,p_T(\omega),p_C(\omega)) \\ -A(s,p_I(\omega) \ + \ B^l(s,p_T(\omega),p_C(\omega)) \end{pmatrix} \xi \leq \begin{pmatrix} h^u(s,p(\omega)) \\ -h^l(s,p(\omega)) \end{pmatrix} \right). \tag{16}$$

The system of random linear inequalities (15b) is replaced then by the probability or chance constraints of the type

$$P_s(\xi(s)) \geq \alpha(s) \textit{ with given } \alpha(s),\ 0\leq\alpha(s)\leq 1,\ 0\leq s\leq s_e. \tag{17}$$

2.2 Cost constraints

Evaluating the violation of the constraints in (15b) by means of a certain nondecreasing cost/loss function $\gamma=\gamma(z)$ on $\Re^{2(n+1)}$ instead of the probability function (16) we obtain the expected cost function

$$\Gamma_s(\xi):=E\gamma\begin{pmatrix} (A(s,p_I(\omega))-B^u(s,p_T(\omega),p_C(\omega)))\xi-h^u(s,p(\omega)) \\ (-A(s,p_I(\omega))+B^l(s,p_T(\omega),p_C(\omega)))\xi+h^l(s,p(\omega)) \end{pmatrix}. \tag{18}$$

As a substitute for (15b), here we get then the mean cost constraints

$$\Gamma_s(\xi) \leq \Gamma_{s,\max},\ 0 \leq s \leq s_e \tag{19}$$

where $\Gamma_{s,\max}$ denotes an upper cost limit.

3. Set of constrained motion

In generalization of the polygon of constrained motion [5] the set of constrained motion $V_{prob}(s), V_{cost}(s)$, *resp.*, is defined by

$$V_{prob}(s) := \left\{ \begin{pmatrix} \beta(s) \\ \beta'(s) \end{pmatrix} : \beta(s) \geq 0,\ P_s(\xi(s)) \geq \alpha(s) \right\}, \tag{20}$$

$$V_{cost}(s) := \left\{ \begin{pmatrix} \beta(s) \\ \beta'(s) \end{pmatrix} : \beta(s) \geq 0,\ \Gamma_s(\xi)) \leq \Gamma_{\max}(s) \right\}. \tag{21}$$

Assuming - for technical reasons - that $V(s)=V_{prob}(s)$, $V(s)=V_{cost}(s)$, resp., is compact for each s, $0 \leq s \leq s_e$, the velocity profils limits

$$\beta_l(s) := \min\{\beta(s): \begin{pmatrix} \beta(s) \\ \beta' \end{pmatrix} \in V(s) \text{ for some } \beta' \in \Re\} \tag{22a}$$

$$\beta_u(s) := \max\{\beta(s): \begin{pmatrix} \beta(s) \\ \beta' \end{pmatrix} \in V(s) \text{ for some } \beta' \in \Re\} \tag{22b}$$

exist for each s, moreover, for each point $(s,\beta(s))$ such that $0 \leq s \leq s_e$ and $\beta_l(s) \leq \beta(s) \leq \beta_u(s)$, the maximum acceleration, deceleration, resp.,

$$f_{\max}(s,\beta(s)) := \max\{\beta': \begin{pmatrix} \beta(s) \\ \beta' \end{pmatrix} \in V(s)\}, \tag{23a}$$

$$f_{\min}(s,\beta(s)) := \min\{\beta': \begin{pmatrix} \beta(s) \\ \beta' \end{pmatrix} \in V(s)\}, \tag{23b}$$

exist. The optimal values in (22a,b), (23a,b) can be determined by means of **stochastic optimization methods** [3].

4. The fields of extremal trajectories

Having the maximum acceleration, deceleration $f_{\max}(s,\beta(\omega))$, $f_{\min}(s,\beta(\omega))$, resp., being robust with respect to stochastic parameter variations, two **robust** fields of **extremal trajectories** $\beta^{\max}=\beta^{\max}(s)$, $\beta^{\min}=\beta^{\min}(s)$, $0 \leq s \leq s_e$, are defined by the differential equations:

$$\beta'(s) = f_{\max}(s,\beta(s)),\ 0 \leq s \leq s_e \tag{24a}$$

$$\beta'(s) = f_{\min}(s,\beta(s)),\ 0 \leq s \leq s_e \tag{24b}$$

where

$$\beta(s) \geq 0,\ 0 \leq s \leq s_e,\ \beta(0)=\beta(s_e)=0. \tag{24c}$$

The **optimal trajectory** $\beta^*=\beta^*(s)$, $0 \leq s \leq s_e$, minimizing the objective function in (15a) subject to probabilistic or expected cost constraints under consideration can now be determined by the same methods as in optimal derterministic path planning [1],[2],[5].

References

[1] Bobrow, J.E. et al.: Time-Optimal control of Robotic Manipulators Along Specified Paths. The Int. J. of Robotics Research 4, No. 3, 3-17 (1985)

[2] Chen, Y.-C.: Solving Robot Trajectory Planning Problems with Uniform Cubic B-Splines. Optimal Control Applications and Methods 12, 247-262 (1991)

[3] Marti, K.: Approximationen stochastischer Optimierungsprobleme. Königstein/Ts.: A. Hain, 1979

[4] Marti, K., Qu, S.: Optimal Trajectory Planning for Robots under the Consideration of Stochastic Parameters and Disturbances. J. of Intelligent and Robotic Systems 15, 19-23 (1996)

[5] Pfeiffer, F., Johanni, J.: A Concept for Manipulator Trajectory Planning. IEEE J. of Robotics and Automation, Vol. RA-3, No. 2, 115-123 (1987)

MODELLING AND SIMULATION OF AN AGRICULTURAL TRACKED VEHICLE

G. Ferretti and R. Girelli
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133, Milano, Italy

Abstract. A new approach to the dynamic modelling of tracked vehicles is proposed in this paper, resulting in a full 3D, 8 d.o.f. dynamic model of an agricultural tracked vehicle. The main features of the approach are an accurate description of the track-terrain interaction, a dynamic model of the shear displacement, and the adoption of an innovative modelling and simulation environment: MOSES, based on Object-Oriented tools and techniques.

1. Introduction

Tracked vehicles modelling has been mainly dealt with in the literature with three different goals:
1) Analysis of steerability ([3,6]);
2) Analysis of ride characteristics ([1]);
3) Theoretical prediction of ground pressure distribution and tractive performance ([2,10,11,12]).

Just in the first two cases dynamic modelling is properly involved, while in the third case stationary motion conditions are assumed. Moreover, in all cases a 2D modelling is considered. In the first case the motion of the vehicle in an horizontal plane is considered, in the second case the attention is focused on the ride dynamic response (vibrations) in a vertical plane, while in the third case the actual goal of the investigation is the prediction of the ground pressure and tractive effort as a function of the track slip. On the other hand, only in the latter case an accurate modelling of the track-terrain interaction is performed, which appears as the most challenging goal in tracked vehicle modelling.

The above drawbacks are overcome with the approach proposed in this paper, which integrates an accurate description of the terramechanics in a full 3D, dynamic model of an agricultural tracked vehicle, characterized by 8 d.o.f. and having the two independently applied sprocket torques as input variables. In particular, a significant and innovative contribution of this work is the modelling of the dynamics of the shear displacement under non-stationary motion conditions, described by a partial derivatives equation. The approach is also innovative from the point of view of the modelling and simulation environment adopted: MOSES [4], a modular modelling environment designed to overcome the limits of available tools in dealing with CAE of complex technological systems. In particular, it is shown how the features of MOSES make easier the modular approach to modelling, while maintaining a high efficiency in simulation.

The paper is organized as follows. Section 2 outlines the modular scheme of the tracked vehicle model, summarizing the modelling assumptions. Section 3 is devoted to the description of the terramechanics, starting from the description of the dynamics of the shear displacement and then introducing the forces arising at the track-terrain interface: 1) normal reaction; 2) tractive effort arising from the shearing of the terrain; 3) motion

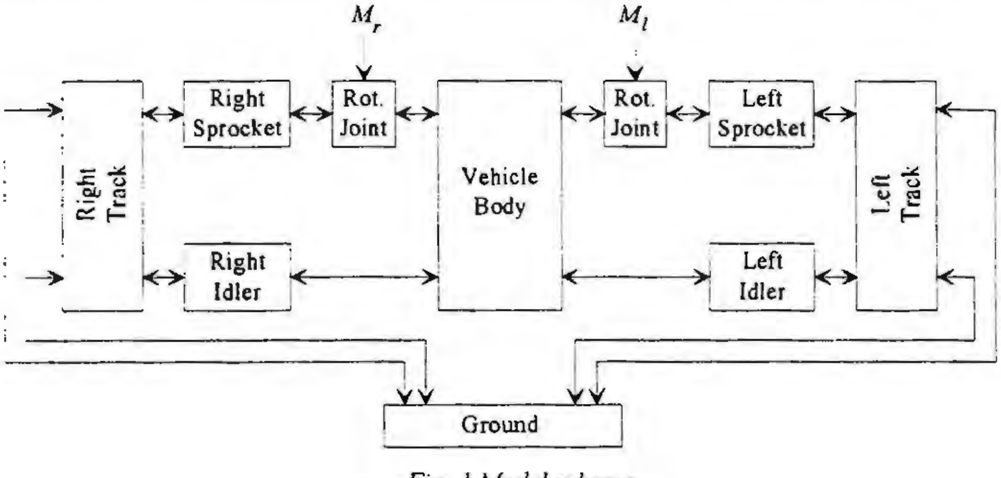


Fig. 1 Model scheme

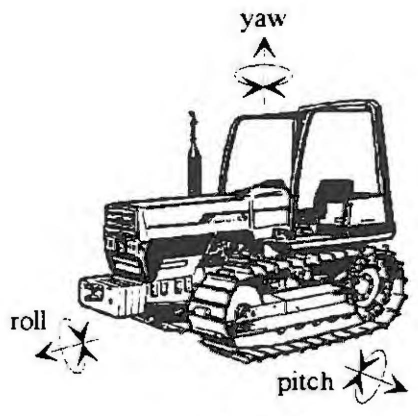


Fig. 2 Roll, Pitch, Yaw

resistance due to terrain compaction; 4) lateral resistance, giving rise to the moment of turning resistance during steering manoeuvres. Section 4 shows the features of MOSES that allows a complete and efficient modular approach to modelling. Section 5 discusses a qualitative validation of the model, based on field experiments with the agricultural tracked vehicle LANDINI TREKKER 75F/ST. Section 5 draws finally some conclusions.

2. The modular model of the tracked vehicle

The overall model of the tracked vehicle, characterized by eigth *d.o.f.*, has been decomposed into subsystems or modules, shown in Fig. 1.

Six *d.o.f.* are introduced by a classical (Newton-Euler) rigid body model, accounting for the dynamics of the vehicle body, whose position and attitude are described by the vector position of its center of mass and by the three roll, pitch and yaw angles of Fig. 2. Other two *d.o.f.* are introduced by the rotational joints connecting the vehicle body to the right and left sprockets. Through these two joints the torques M_r and M_l, which are the only exogenous variables of the model, are independently applied to the right and left sprockets respectively.

The longitudinal elasticity of the track has been neglected, and it is therefore considered as an inextensible driving belt with respect to the rolling motion of the sprockets, idlers and roadwheels. As a consequence, the angular velocities of the rolling parts are assumed as algebraically related and the rolling inertial effects of the track are entirely ascribed to the sprocket, whose moment of inertia around the rolling axis has been suitably augmented. The aerodynamic resistance can also be neglected, owing to the low operating speed of agricultural vehicles [9], as well as the effects of suspensions [2].

The track modules deal with the main modelling problem, namely the description of the vehicle-terrain interaction, and will be described in detail in the next section.

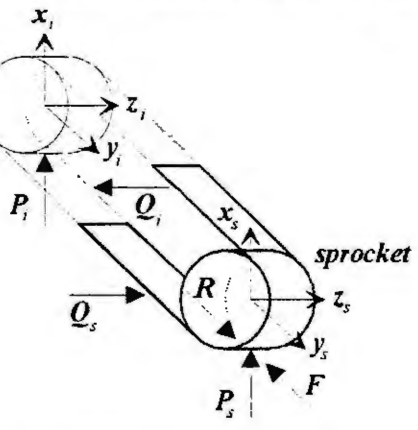


Fig. 3 Forces acting on the track

3. Track-terrain interaction model

The track has been divided in two parts (Fig. 3), the sprocket and the idler, each one located by a frame and assumed as a rigid footing[1] in order to compute the normal reactions $\boldsymbol{P_s}$ and $\boldsymbol{P_i}$ and the lateral resistances $\boldsymbol{Q_s}$ and $\boldsymbol{Q_i}$, giving rise to a moment of turning resistance during skid-steering. The force $\boldsymbol{R}$ is the motion resistance due to terrain compaction, while $\boldsymbol{F}$ is the thrust.produced by the shearing of the terrain.

All these forces depend on the terrain response, which can be characterized by the pressure-sinkage and by the shear stress-shear displacement relationships, and by the coefficient of lateral resistance [8,9,11]. The thrust in particular is a function of the shear displacement, whose computation in dynamic conditions deserves a special attention and will be dealt with first in the next subsection.

3.1 Dynamic model of the shear displacement

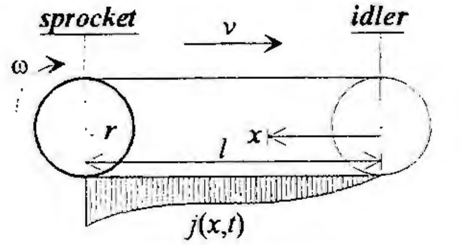


Fig. 4 Shear displacement in forward motion

During non-stationary motion the shear displacement j can be in general defined as a function of time t and of the distance x from the front of the contact area (see Fig. 4). On the other hand, since the track cannot stretch, the speed of slip of the track with reference to the ground v_j is the same for every point of the track in contact with the terrain, v_j being given by the difference between the theoretical speed $v_t = r\omega$, where ω and r are the angular velocity and the radius of the pitch circle of the sprocket, and the actual

[1] This assumption should be realistic in the case of agricultural tracked vehicles [9].

forward speed of the track v (according to the conventions of Fig. 4):

$$v_j = v_t - v = r\omega - v\,.$$

Since the speed of slip v_j is also equal to the *total* time derivative of the shear displacement:

$$v_j = \frac{dj(x,t)}{dt} = \frac{\partial j(x,t)}{\partial t} + \frac{\partial j(x,t)}{\partial x}\frac{dx}{dt}\,,$$

and the theoretical velocity v_t also defines the relative velocity of the track with respect to the vehicle body, i.e. $v_t = dx/dt$, the dynamic model of the shear displacement is defined by the following non-linear, partial derivatives equation:

$$\frac{\partial j}{\partial t} + r\omega\frac{\partial j}{\partial x} = r\omega - v\,, \tag{1.1}$$

together with the boundary conditions

$$\begin{cases} j(t,0) = 0\,, & \omega > 0 \quad (\textit{forward}\text{ motion}) \\ j(t,l) = 0\,, & \omega < 0 \quad (\textit{backward}\text{ motion}) \end{cases} \tag{1.2}$$

imposing a null shear displacement where the track engages the terrain.

For the sake of the numerical simulation an ordinary differential equation should be recovered from model (1). To this aim, integrate both members of (1.1) with respect to the spatial variable x, along the length l of the contact area, and introduce the mean value $\tilde{j}(t)$ of the shear displacement:

$$\begin{cases} l\dfrac{d\tilde{j}(t)}{dt} = -r\omega(t)j(t,l) + l[r\omega(t) - v(t)]\,, & \omega > 0 \\ l\dfrac{d\tilde{j}(t)}{dt} = r\omega(t)j(t,0) + l[r\omega(t) - v(t)]\,, & \omega < 0 \end{cases}$$

If a linear profile is assumed for the shear displacement then

$$\begin{cases} j(t,l) = 2\tilde{j}(t)\,, & \omega > 0 \\ j(t,0) = 2\tilde{j}(t)\,, & \omega < 0 \end{cases},$$

so that the following non-linear, first order differential equation is finally obtained:

$$\frac{d\tilde{j}}{dt} = -\frac{2r}{l}|\omega|\tilde{j} + r\omega - v\,.$$

3.2 Normal forces

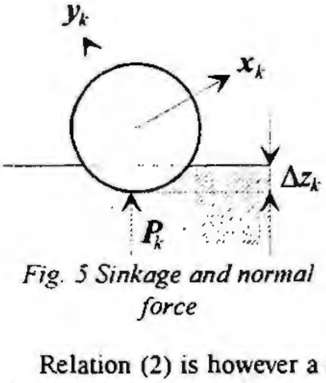


Fig. 5 Sinkage and normal force

Assuming a uniform normal pressure distribution under both segments of the track, the reaction forces $\boldsymbol{P}_k$ $(k = s,i)$ are related to the pressure p_k as:

$$\boldsymbol{P}_k = \frac{bl}{2}p_k\boldsymbol{z}_0\,,$$

where b is the width of the track and $\boldsymbol{z}_0$ is the absolute normal unit vector. In turn, the pressure p_k can be related to the sinkage of the lowest point of both sprocket and idler Δz_k (see Fig. 5) through some empirical pressure-sinkage relationship, for example, if an homogeneous terrain is considered:

$$p_k = (k_c/b + k_\phi)(\Delta z_k)^n \tag{2}$$

where k_c, k_ϕ and n are the pressure-sinkage parameters [9].

Relation (2) is however a nonlinear elastic characteristic, while the compacting of the terrain is essentially plastic. This means that the pressure-sinkage relationships during loading or unloading are different, giving rise to a significant amount of hysteresis during the loading-unloading cycle, with an associated energy dissipation. This fact has been simply taken into account by adding a viscous term in (2), thus:

$$p_k = \left(\frac{k_c}{b} + k_\phi\right)(\Delta z_k)^n + k_v\frac{d\Delta z_k}{dt}$$

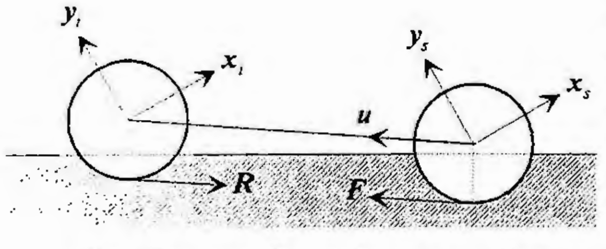


Fig. 6 Tractive effort and motion resistance

where k_v is a suitable parameter; in this way the terrain response to repetitive loading is not modelled but consistency is gained at least from an energetic point of view.

3.3 Tractive effort

The amplitude F of the total tractive effort developed by the track is given by the integral of the shear stress $\tau(t,x)$ along the contact area. In turn, the shear stress can be computed as a function of the shear displacement through shear stress-shear displacement relationships dependent on the particular terrain.

Assuming a uniform distribution of the shear stress along the width of the track, in the case of homogeneous terrain and forward motion the amplitude of the tractive effort is given by [9]:

$$F(t) = b\int_0^l \tau(t,x)\,dx = b\int_0^l \left[c + p(t,x)\tan\phi\right]\left[1 - e^{-j(t,x)/K}\right]dx$$

where p is the normal pressure and c, ϕ, K are characteristic parameters of the terrain.

If the pressure $p(t,x)$ is computed as the mean between the pressures under the sprocket and the idler, while recovering the assumption of a linear profile for the shear displacement, the tractive effort is given by:

$$\boldsymbol{F} = \mathrm{sign}(\tilde{j})F\boldsymbol{u} = \mathrm{sign}(\tilde{j})\,bl\left[c + \frac{p_s + p_i}{2}\tan\phi\right]\left[1 + \frac{K}{2|\tilde{j}|}\left(e^{-2|\tilde{j}|/K} - 1\right)\right]\boldsymbol{u}\,.$$

where $\boldsymbol{u}$ is the unit vector determined by the vector position of the idler frame with respect to the sprocket frame. The tractive effort is applied in the lowest point of the sprocket pitch circle (Fig. 6).

3.4 Motion resistance

The motion resistance is ascribed to terrain compaction and is computed by equating the work done by the towing force when pulling the track for a distance l to the work done in making a rut of lenght l [9]. If the track is assumed as a rigid footing, a uniform normal pressure distribution is considered, and the sinkage is computed as the mean between the sprocket and the idler sinkage, the motion resistance is given by:

$$\boldsymbol{R} = -\mathrm{sign}(v_{iu})R\boldsymbol{u} = -\mathrm{sign}(v_{iu})\frac{b}{n+1}\left(\frac{\Delta z_s + \Delta z_i}{2}\right)^{n+1}\left(\frac{k_c}{b} + k_\phi\right)\boldsymbol{u}$$

where v_{iu} is the component of the velocity of the origin of idler frame along the unit vector $\boldsymbol{u}$. The motion resitance is applied to the lowest point of the idler pitch circle (Fig. 6).

3.5 Lateral resistance

The lateral resistance, giving rise to the moment of turning resistance opposing to the steering of the vehicle, can be directly related to the normal load through a coefficient of lateral resistance μ_t, depending on the terrain and on the design of the track. Assuming a uniform pressure distribution under both segments of the track, the amplitude of the lateral resistance Q_k is therefore given by:

$$Q_k = \mu_t P_k\,.$$

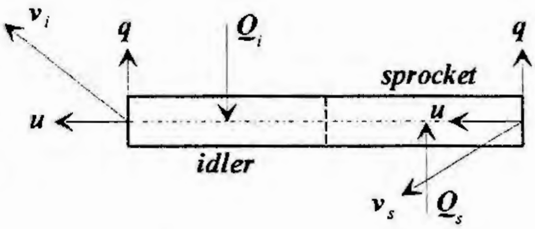


Fig. 7 Lateral resistance

The lateral resistance vector $\boldsymbol{Q}_k$ is considered applied in the middle of each segment (Fig. 7), at a distance $l/4$ from the origins of the sprocket and idler frame, in a direction normal to the unit vector $\boldsymbol{u}$, defined by the unit vector $\boldsymbol{q} = (\boldsymbol{u} \times \boldsymbol{z}_0)/|\boldsymbol{u} \times \boldsymbol{z}_0|$ and by the component v_{kq} of the velocity $\boldsymbol{v}_{kq}$ along the unit vector $\boldsymbol{q}$, thus:

$$\boldsymbol{Q}_k = -\mathrm{sign}(v_{kq})\mu_t P_k\boldsymbol{q}\,.$$

4. MOSES: An Object-Oriented modelling and simulation environment

The model has been written and simulated in MOSES (Modular Object-oriented Software Environment for Simulation) [4], a prototype developed at Politecnico di Milano to overcome the limits of available modelling and simulation tools in dealing with CAE of complex technological systems. The main characteristics of MOSES are the following:

- it is implemented in an (object-oriented) database, in order to guarantee integrity of data and security while sharing data among different users;
- it uses an Object-Oriented (OO) modelling language, which largely simplifies the definition and reuse of physical systems models;
- a symbolic formula manipulation is performed in order to simplify the system of equations defined, and to improve efficiency during numerical analisys.

The model definition language implements concepts like abstraction, encapsulation and standardization of interfaces, so achieving a high level of modularity. Models are defined in a declarative way, that is, by means of sets of variables and parameters representing physical properties, together with a set of equations stating the relations among them. Equations are not written in explicit form: the actual causality (eventually) needed by the specific simulation problem is determined by the symbolic manipulation.

To implement the abstraction mechanism, a set of interconnected (sub)models can be grouped in an aggregate model, which acts as a proxy of them: in this way, different levels of decomposition can be identified.

To enhance models' reuse, physical ports (in MOSES called *terminals* [7]) are strictly standardized: they are basic elements of the modelling language, and represent the energy exchanges involved in physical systems' interactions. Since the kinds of energy exchange are well known in every application domain, the structure of any physical terminal can be defined at language level. For example, in 3-D mechanics only one type of mechanical terminal can be used: it has associated a terminal frame which is described by the frame origin coordinates and orientation together with the transmitted force and torque, expressed in that frame. The connection between two mechanical terminals is equivalent to a perfect rigid overlapping of the two frames associated to the terminals and is the abstraction of the ideal "welding" between the (two) mechanical components realised at the frame location. It must be noted that, in order to prevent problems with system index [5], also linear and angular velocities and their derivatives have been included in the set of variables belonging to the terminal.

Given the physical meaning of models and their interfaces, it is possible to implement the so called physical modelling, that means building a global model which has the same structure and topology of the real system it represents. This feature, together with the use of abstraction hierarchies, is very useful in understanding even very complex models. For example, the elements of the model scheme shown in Fig. 1 have a one-to-one correspondence with the elements of the MOSES window showing the global model.

5. Simulation results

The model has been qualitatively validated by comparing simulations with motion experiments, performed with the tracked vehicle LANDINI TREKKER 75F/ST (Fig. 2) moving on an hard homogeneous clayey terrain. Two different motion experiments are here discussed, considered as particularly expressive in order to appreciate the reliability of the model.

First, it has been noted that by abruptly vanishing the torque applied to one sprocket, starting from a straight motion condition, the motion of the vehicle exhibits a very small deviation from the straigth trajectory. To verify this behaviour in a first simulation experiment a torque[2] of 2000 Nm is initially applied to both sprockets then, after 4 s, the left sprocket is set idle. The center of mass of the vehicle moves initially along the absolute x_0 axis, so that the displacement along the y_0 axis gives directly the deviation from the straight trajectory. Figure 8 shows the simulated trajectory of the center of mass on the horizontal plane (note the different scales adopted to emphasize the trajectory deviation)

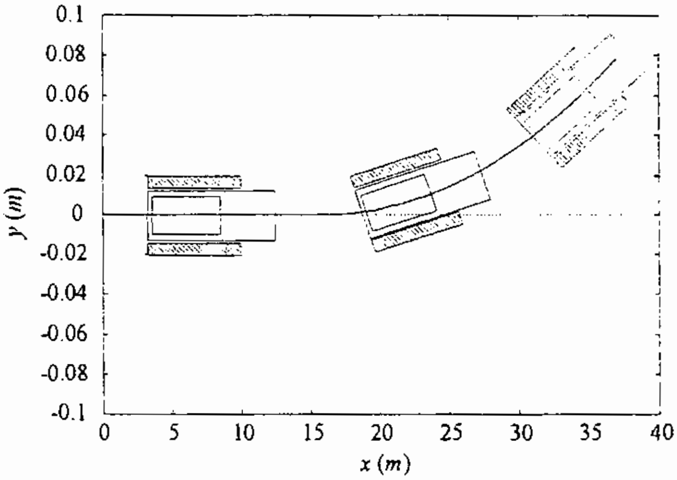


Fig. 8 Idle sprocket experiment: horizontal trajectory

[2] Corresponding to nearly 30% of the maximum torque

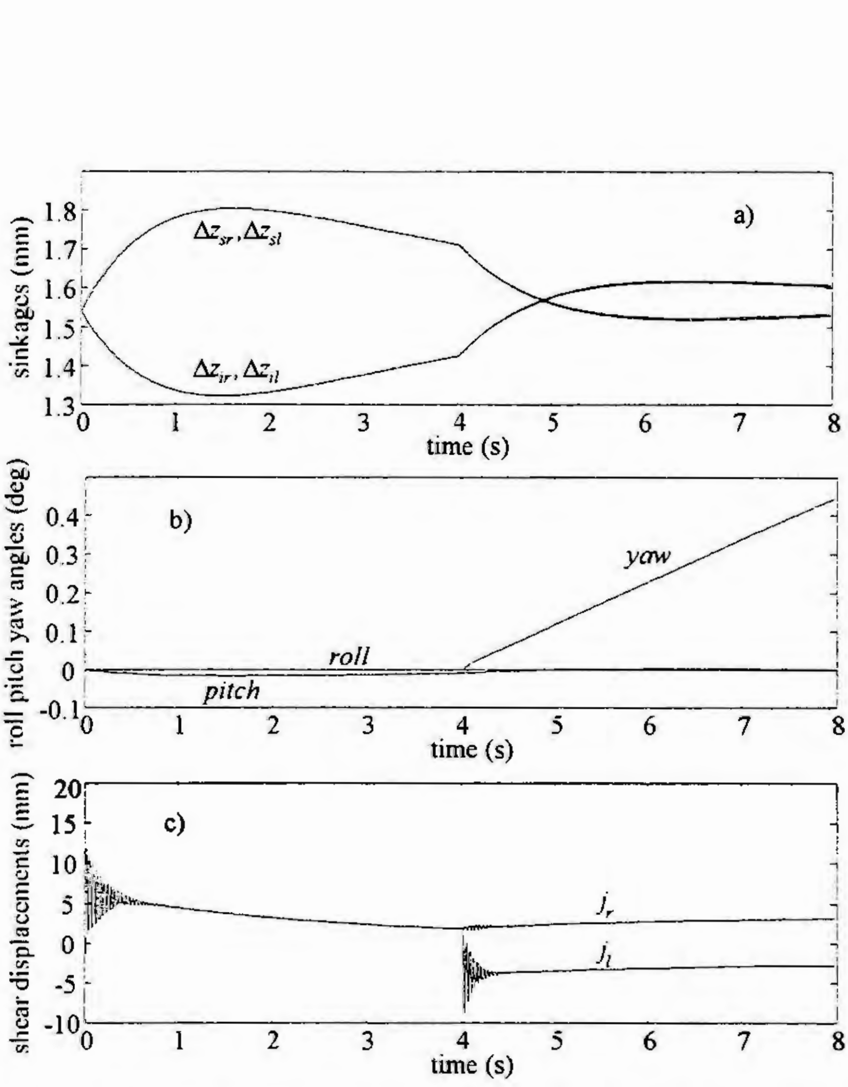


Fig. 9 Idle sprocket experiments: sinkages, angles and shear displacements

while Figs. 9.a,b,c show respectively the sinkages, the roll, pitch and yaw angles and the shear displacements.

It must be pointed out that:

- from the instant when the left sprocket is set idle the vehicle covers nearly 20 m, while the deviation from the straight trajectory is less than 8 cm (a mean forward velocity of 5 m/s is obtained, which agrees with experiments);
- the sinkages (Fig. 9.a) are in the range of 2 mm, corresponding to a hard terrain;
- the variations of the yaw angle only become appreciable (Fig. 9.b), incresing linearly from 0° to 0.4° in the time interval where the left sprocket is idle.
- the shear displacements are in the range of 5÷10 mm (Fig. 9.c), the left shear displacement becoming negative when the left sprocket is idle;
- in the first 4 s the vehicle assumes a "nose-up" attitude, then the pitch angle decreases.

It must be finally mentioned that the shear displacements are affected by damped high frequency oscillations, scarcely influencing the motion of the vehicle and mainly due to the fact that the terrain response to repetitive shearing has been neglected (together with the associated energy dissipation).

It has been also observed on the field that, without loading the vehicle, it is necessary to brake or even lock one sprocket in order to steer. In particular it has been observed that, starting form rest, if one sprocket is maintained locked the trajectory becomes circular, with the center located near the center of the locked track. The same experiment has been repeated in simulation, applying a torque of 2000 N to the left sprocket and maintaining locked the right one, and the resulting trajectory of the center of mass is reported in Fig. 10.

It can be observed that:

- the resulting trajectory is indeed circular;
- the vehicle turns more than one rev in 12 s (the yaw angle reaches 450°), in agreement with the experiment;
- the shear displacements are in the range of 5 mm, the right shear displacement remaining negative.

6. Conclusions

The dynamic model of an agricultural tracked vehicle has been outlined in this paper. The main distinctive features of the model are the following:

- a full 3D dynamic model has been developed, thus including gyroscopic effects;
- the track-terrain interaction has been accurately described, based on well known experimentally derived relationships (relating ground pressure to sinkage and shear stress to shear displacement) and on a new dynamic model of the shear displacement under non-stationary motion conditions;
- a modular approach to modelling has been adopted, based on an innovative modelling and simulation environment (MOSES).

The model has been qualitatively validated through field experiments, performed on the tracked vehicle LANDINI TREKKER 75F/ST.

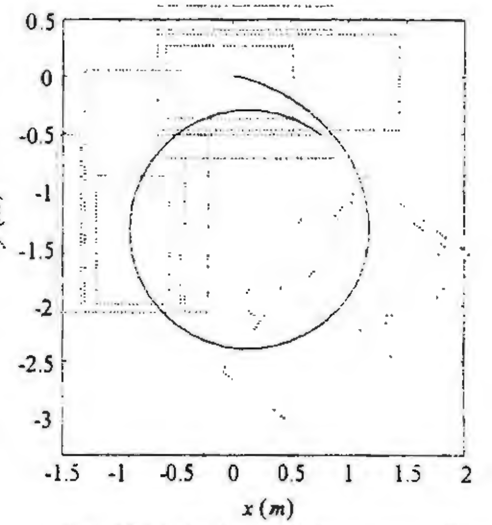


Fig. 10 Locked sprocket experiment

It is expected that the modular approach and the modelling environment adopted will allow an easy estension of the model, including the dynamic of suspensions and the models of motor, brakes and leverages.

Acknowledgement

The authors would like to thank Prof. C. Maffezzoni for many helpful discussions on the subject of this paper.

References

1. Dhir, A. and Sankar S., Analytical track models for ride dynamic simulation of tracked vehicles. Journal of Terramechanics, 31 (1994), 2, 107-138.
2. Gigler, J. K. and Ward, M., Simulation model for the prediction of the ground pressure distribution under tracked vehicles. Journal of Terramechanics, 30 (1993), 6, 461-469.
3. Kitano, M. and Kuma M., An analysis of horizontal plane motion of tracked vehicles. Journal of Terramechanics, 14 (1977), 4, 211-225.
4. Maffezzoni, C., Girelli, R. and Lluka, P., Object-oriented database support for modular modelling and simulation. Modelling and Simulation ESM 94, Barcellona, 1994, pp. 354-361.
5. Petzold, L. R., A description of DASSL: a Differential/Algebraic system solver. Scientific Computing, North-Holland, 1983.
6. Watanabe, K. and Kitano M., Study on steerability of articulated tracked vehicles - Part 1. Theoretical and experimental analysis. Journal of Terramechanics, 23 (1986), 2, 69-83.
7. Wellstead, P.E., Physical System Modelling. Academic Press, 1979.
8. Wong, J. Y., Data processing methodology in the characterization of the mechanical properties of terrain. Journal of Terramechanics, 17 (1980), 1, 13-41.
9. Wong, J. Y., Theory of ground vehicles. J. Wiley, New York, 1991.
10. Wong J. Y., Garber M. and Preston-Thomas J., Theoretical prediction and experimental substantiation of the ground pressure distribution and tractive performance of tracked vehicles, Proceedings of the Institution of Mechanical Engineers, 198 (1984), D15, 265-285.
11. Wong, J. Y. and Preston Thomas, J., On the characterization of the shear stress-displacement relationship of terrain. Journal of Terramechanics, 19 (1983), 4, 225-234.
12. Wong J. Y. and Preston-Thomas, J., Investigation into the effects of suspension characteristics and design parameters on the performance of tracked vehicles using an advanced computer simulation model, Proceedings of the Institution of Mechanical Engineers, 202 (1988), D3, 143-161.

CALCULATION OF THE JACOBIAN MATRIX FOR MECHANISMS WITH MORE THEN ONE DOF IN EACH JOINT

K. Gotlih

University of Maribor, Faculty of Mechanical Engineering
Smetanova 17, SI-2000 Maribor, Slovenia
GOTLIH@UNI-MB.SI

Abstract. The Jacobian matrix is the most used matrix in the field of kinematic and kinetic simulation (analysis and synthesis) of open kinematic chains. Such open kinematic chains are robot mechanisms and human arms and legs. This matrix is for simple non-redundant mechanisms with good known algorithms easy to obtain. For redundant mechanisms with complicated joints, with more than one degree of freedom (DOF) in each joint, is the calculation of this matrix time consuming and complicated.

The aim of this paper is to show an approach to calculate the Jacobian matrix as a block matrix with the use of Kronecker products. To illustrate the efficiency of the algorithm, the Jacobian matrix of a simplified human arm model with 6 DOFs, is shown.

Introduction

The developing of the Jacobian matrix, for mechanism simulation purposes, for redundant open kinematic chains, where the joints are combined with more than one DOF, is a tedious job. There are many ways to develop the matrix but all of them require big amount of computation. The Jacobian matrix is very important in the field of redundant open kinematic chains because most of the control algorithms for this category of mechanisms need this matrix. It will be of benefit to calculate this matrix for various structures of redundant mechanisms either in analytic or symbolic form. The time, consumed for the Jacobian matrix developing, can be so in the simulation (control algorithm) minimised.

The Jacobian matrices differ dependent on the mechanisms structures and the determination of them is different for each mechanism. The Jacobian matrix is the personal card of mechanism and contain many very useful information about the observed mechanism.

There are some well-known methods for the Jacobian matrix determination. The method of Whitney, [1], is used most frequently. This method has its base in the velocity relation transformation between the task and configuration space. The method of Paul, [2], is very useful too. In this method the Jacobian matrix is developed with use of the DH notation [3].

In many cases symbolical differentiation or numerical differentiation of the transformation function between the task and configuration space is used. These two approaches require big amount of computation for the developing.

The introduced method for the Jacobian matrix calculation is based on the use of Kronecker products and on the fact that we, for the developing, need just derivatives of the elementary transformation matrices with respect to the generalised co-ordinates. This differentiation can be done for each matrix analytically and then substituted to the whole simulation algorithm.

Development of the Jacobian matrix

The transformation between the task space $\bar{x}$ and configuration space $\bar{\varphi}$ can be written for an arbitrary chosen mechanism in the form:

$$\bar{x} = \underline{f}(\bar{\varphi}) \tag{1}$$

where $\underline{f}$ denotes the vector function of the transformation. The equation (1) can be, with respect to the notation of the relative motion between two neighboured links that are connected with a joint, written in the form:

$$\bar{x}_{n,0} = \underline{A}_1 \cdot \bar{x}_1 + \underline{A}_1 \cdot \underline{A}_2 \cdot \bar{x}_2 + \cdots + \underline{A}_1 \cdot \underline{A}_2 \cdot \underline{A}_3 \cdots \underline{A}_n \cdot \bar{x}_n \tag{2}$$

The matrix $\underline{A}_i$ is the transformation matrix between two links through a joint, and $\bar{x}_i$ is the vector placed in the relative co-ordinate frame "i" in the direction to the relative co-ordinate frame "$i+1$". This vector is constant when the joint permits relative revolute motions of the neighboured links (relative co-ordinate frames) and variable if the joint permits relative translational movements of the neighboured links (relative co-ordinate frames).

The differentiation of the equation (1) with respect to time gives:

$$\begin{aligned}\dot{\bar{x}}_{n,0} = {} & \dot{\underline{A}}_1.\bar{x}_1 + \underline{A}_1.\dot{\bar{x}}_1 + \dot{\underline{A}}_1.\underline{A}_2.\bar{x}_2 + \underline{A}_1.\dot{\underline{A}}_2.\bar{x}_2 + \underline{A}_1.\underline{A}_2.\dot{\bar{x}}_2 + \cdots + \dot{\underline{A}}_1.\underline{A}_2.\underline{A}_3 \cdots \underline{A}_n.\bar{x}_n + \\ & \underline{A}_1.\dot{\underline{A}}_2.\underline{A}_3 \cdots \underline{A}_n.\bar{x}_n + \underline{A}_1.\underline{A}_2.\dot{\underline{A}}_3 \cdots \underline{A}_n.\bar{x}_n + \cdots + \underline{A}_1.\underline{A}_2.\underline{A}_3 \cdots \dot{\underline{A}}_n.\bar{x}_n + \underline{A}_1.\underline{A}_2.\underline{A}_3 \cdots \underline{A}_n.\dot{\bar{x}}_n\end{aligned} \tag{3}$$

The matrix $\underline{A}_i$ can also be a product of rotational matrices, so that each joint can be modelled as a 3DOFs spherical joint.

$$\underline{A}_i = \underline{A}_{xi}.\underline{A}_{yi}.\underline{A}_{zi} \tag{4}$$

The differentiation of the matrix $\underline{A}_i$ with respect to time is, [5]:

$$\frac{d}{dt}\underline{A}_i = \left(\left(\frac{d\bar{\varphi}_i^{\,T}}{dt} \otimes \underline{I}_{pi}\right).\frac{\partial \underline{A}_i}{\partial \bar{\varphi}_i} + \frac{\partial \underline{A}_i}{\partial t}\right) \tag{5}$$

The dimension p of the unit matrix is the number of components of the vector $\bar{\varphi}_i$. $\bar{\varphi}_i^{\,T}$ denotes the transpose of $\bar{\varphi}_i$ and $\underline{I}_{pi}$ is the p dimensional unit matrix. The matrix $\underline{A}_i$ is dependent on vector $\bar{\varphi}_i$ from the configuration space and not on time t. Vector $\bar{\varphi}_i$ is time dependent, so that the term (5) is:

$$\frac{d}{dt}\underline{A}_i = \left(\frac{d\bar{\varphi}_i^{\,T}}{dt} \otimes \underline{I}_{pi}\right).\frac{\partial \underline{A}_i}{\partial \bar{\varphi}_i} \tag{6}$$

The sign $\otimes$ in (5) and (6) is the Kronecker product operator [4], [5], [6]. The result (6) with the substitution $\frac{d\bar{\varphi}_i^T}{dt} = \dot{\bar{\varphi}}_i^T$ is put into the equation (3).

$$\begin{aligned}\dot{\bar{x}}_{n,0} = {} & (\dot{\bar{\varphi}}_1^T \otimes \underline{I}_{p1}).\frac{\partial \underline{A}_1}{\partial \bar{\varphi}_1}.\bar{x}_1 + \underline{A}_1.\dot{\bar{x}}_1 + (\dot{\bar{\varphi}}_1^T \otimes \underline{I}_{p1}).\frac{\partial \underline{A}_1}{\partial \bar{\varphi}_1}.\underline{A}_2.\bar{x}_2 + \\ & \underline{A}_1.(\dot{\bar{\varphi}}_2^T \otimes \underline{I}_{p2}).\frac{\partial \underline{A}_2}{\partial \bar{\varphi}_2}.\bar{x}_2 + \underline{A}_{1,}.\underline{A}_2.\dot{\bar{x}}_2 + \cdots\end{aligned} \tag{7}$$

If we purpose that just revolute joints, as in robotics often appear and in human extremities completely, then vector $\bar{x}_i$ is constant and the time differential of it vanishes. The equation is modified and written in the form:

$$\dot{\bar{x}}_{n,0} = (\dot{\bar{\varphi}}_1^T \otimes \underline{I}_{p1}).\frac{\partial \underline{A}_1}{\partial \bar{\varphi}_1}.\bar{x}_1 + (\dot{\bar{\varphi}}_1^T \otimes \underline{I}_{p1}).\frac{\partial \underline{A}_1}{\partial \bar{\varphi}_1}.\underline{A}_2.\bar{x}_2 + \underline{A}_1.(\dot{\bar{\varphi}}_2^T \otimes \underline{I}_{p2}).\frac{\partial \underline{A}_2}{\partial \bar{\varphi}_2}.\bar{x}_2 + \cdots \tag{8}$$

This equation (8) can be transformed:

$$\begin{aligned}\dot{\bar{x}}_{n,0} = {} & (\dot{\bar{\varphi}}_1^T \otimes \underline{I}_{p1}).\frac{\partial \underline{A}_1}{\partial \bar{\varphi}_1}.(\bar{x}_1 + \underline{A}_2.\bar{x}_2 + \underline{A}_2.\underline{A}_3.\bar{x}_3 + \cdots) + \\ & \underline{A}_1.(\dot{\bar{\varphi}}_2^T \otimes \underline{I}_{p2}).\frac{\partial \underline{A}_2}{\partial \bar{\varphi}_2}.(\bar{x}_2 + \underline{A}_3.\bar{x}_3 + \cdots) + \underline{A}_1.\underline{A}_2.(\dot{\bar{\varphi}}_3^T \otimes \underline{I}_{p3}).\frac{\partial \underline{A}_3}{\partial \bar{\varphi}_3}.(\bar{x}_3 + \underline{A}_4.\bar{x}_4 + \cdots)\end{aligned} \tag{9}$$

The main problem is now to simplify the term:

$$\underline{U}_i.(\dot{\bar{\varphi}}_i^T \otimes \underline{I}_{pi}).\bar{r}_i \tag{10a}$$

where is:

$$\underline{U}_i = \underline{I}_{pi} \cdot \underline{A}_1 \cdot \underline{A}_2 \cdot \underline{A}_3 \cdot \cdots \cdot \underline{A}_{i-1} \tag{10b}$$

and

$$\bar{r}_i = \frac{\partial \underline{A}_i}{\partial \varphi_i} . (\bar{x}_i + \underline{A}_{i+1} . \bar{x}_{i+1} + \cdots) \tag{10c}$$

To solve the problem (10a) we initiate the *vec* operator, [6], and write:

$$\bar{r}_i = vec(\underline{R}_i) \tag{10d}$$

With respect to [6] and (10d) the term (10a) is written in the form:

$$\underline{U}_i . (\dot{\bar{\varphi}}_i^T \otimes \underline{I}_{pi}) . \bar{r}_i = \underline{U}_i . vec(\underline{R}_i . \underline{D}_i) \tag{11}$$

where $\underline{D}_i$ is one column matrix form of vector $\dot{\bar{\varphi}}_i$. The equation (7) can now be written:

$$\dot{\bar{x}}_{n,0} = \sum_{i=1}^{n} \underline{U}_i . vec(\underline{R}_i . \underline{D}_i) \tag{12}$$

The product in the round brackets in (12) of matrix $\underline{R}_i$ and one column matrix $\underline{D}_i$ is a vector. The *vec* operator, applied on the vector, has no effect, so this operator is omitted in the equation (12). The result is :

$$\dot{\bar{x}}_{n,0} = \sum_{i=1}^{n} \underline{U}_i . \underline{R}_i . \dot{\bar{\varphi}}_i \tag{13}$$

The product of matrices is $\underline{U}_i . \underline{R}_i = \underline{B}_i$. The equation (13) is tben:

$$\dot{\bar{x}}_{n,0} = \sum_{i=1}^{n} \underline{B}_i . \dot{\bar{\varphi}}_i \tag{14}$$

From (14) we get the Jacobian matrix. This matrix is a block matrix as shown in (15).

$$\underline{J} = \left[\underline{B}_1 \vdots \; \underline{B}_2 \vdots \; \underline{B}_3 \vdots \; \cdots \; \vdots \underline{B}_n \right] \tag{15}$$

At the end we can write:

$$\dot{\bar{x}}_{n,0} = \underline{J} . \dot{\bar{\varphi}} \tag{16}$$

Example

As mentioned in the introduction this calculation procedure for the Jacobian matrix determination is usable for the simulation of complicated joints as appear in human extremities (human shoulder, elbow or wrist joints in a human arm). In the example we show the Jacobian matrix determination of a simplified human arm model. The simplifications are in the wrist where all DOFs are neglected The arm is redundant, because we control just the position of arm tip. The human arm model is shown in Fig. 1.

With respect to the equation (1) there are:

$$\bar{x} = \{x \quad y \quad z\}^T$$

and

$$\bar{\varphi} = \{\varphi_{x1} \quad \varphi_{y1} \quad \varphi_{z1} \quad \varphi_{x2} \quad \varphi_{y2} \quad \varphi_{z2}\}^T .$$

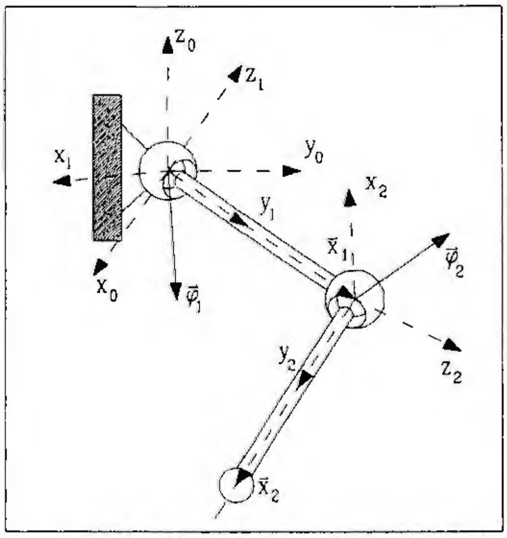


Fig. 1. Simplified human arm model

The arm model has two spherical joints, Fig 1, with each three DOFs. The first is in the shoulder and the second in the elbow. The shoulder joint variables vector is

$\bar{\varphi}_1 = \left\{ \varphi_{x1} \quad \varphi_{y1} \quad \varphi_{z1} \right\}^T$ and the elbow joint variables vector is $\bar{\varphi}_2 = \left\{ \varphi_{x2} \quad \varphi_{y2} \quad \varphi_{z2} \right\}^T$.

To develop the Jacobian matrix, matrices $\underline{U}_i$ from (10b) and the vector $\bar{r}_i$ form (10c) must be calculated. We purpose that each of the joints (the shoulder and the elbow joint) is combined as an *x*, *y* and *z* rotational joint. The transformation matrices for the *x*, *y* and *z* rotation are the well-known orthogonal rotational matrices:

$$\underline{A}_{xi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_{xi}) & -\sin(\varphi_{xi}) \\ 0 & \sin(\varphi_{xi}) & \cos(\varphi_{xi}) \end{bmatrix}_{i=1,2} \qquad \underline{A}_{yi} = \begin{bmatrix} \cos(\varphi_{yi}) & 0 & \sin(\varphi_{yi}) \\ 0 & 1 & 0 \\ -\sin(\varphi_{yi}) & 0 & \cos(\varphi_{yi}) \end{bmatrix}_{i=1,2} \qquad \underline{A}_{zi} = \begin{bmatrix} \cos(\varphi_{zi}) & -\sin(\varphi_{zi}) & 0 \\ \sin(\varphi_{zi}) & \cos(\varphi_{zi}) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{i=1,2}$$

The product of these three matrices is:

$$\underline{A}_i = \underline{A}_{xi} \cdot \underline{A}_{yi} \cdot \underline{A}_{zi} \qquad \underline{A}_i = \begin{bmatrix} cyi.czi & -(cyi.szi) & syi \\ czi.sxi.syi + cxi.szi & cxi.czi - sxi.syi.szi & -(cyi.sxi) \\ -(cxi.czi.syi) + sxi.szi & czi.sxi + cxi.syi.szi & cxi.cyi \end{bmatrix} \qquad (i=1,2)$$

Where is: $sxi = sin(\varphi_{xi})$, $syi = sin(\varphi_{yi})$, $szi = sin(\varphi_{zi})$, $cxi = cos(\varphi_{xi})$, $cyi = cos(\varphi_{yi})$ and $czi = (\varphi_{zi})$.

The partial differentiation of the matrix $\underline{A}_i$ with respect to the vector $\bar{\varphi}_i$ is:

$$\frac{\partial \underline{A}_i}{\partial \bar{\varphi}_i} = \begin{bmatrix} 0 & , & 0 & , & 0 \\ cxi.czi.syi - sxi.szi & , & -(czi.sxi) - cxi.syi.szi & , & -(cxi.cyi) \\ czi.sxi.syi + cxi.szi & , & cxi.czi - sxi.syi.szi & , & -(cyi.sxi) \\ -(czi.syi) & , & syi.szi & , & cyi \\ cyi.czi.sxi & , & -(cyi.sxi.szi) & , & sxi.syi \\ -(cxi.cyi.czi) & , & cxi.cyi.szi & , & -(cxi.syi) \\ -(cyi.szi) & , & -(cyi.czi) & , & 0 \\ cxi.czi - sxi.syi.szi & , & -(czi.sxi.syi) - cxi.szi & , & 0 \\ czi.sxi + cxi.syi.szi & , & cxi.czi.syi - sxi.szi & , & 0 \end{bmatrix} \qquad (i=1,2)$$

In the equations (2), (3), (7), (8), (9) and (10c) we need vector $\bar{x}_i$ that defines the positions of local co-ordinate frames (dimensions of the mechanism links). In this example, Fig. 1, there are two vectors with constant values: $\bar{x}_1 = \{0 \quad l_1 \quad 0\}^T$ and $\bar{x}_2 = \{0 \quad l_2 \quad 0\}^T$.

The length's l_1 and l_2 are the lengths of upper and forearm of the chosen simplified human arm model. The vector $\bar{r}_i$ $(i=1,2)$ is from equation (10c)

$$\begin{aligned}
\bar{r}_1 = \{&0,-(l_2.cy2.(cx1.cz1.sy1-sx1.sz1).sz2)-l_2.cx1.cy1.(cz2.sx2+cx2.sy2.sz2)+\\
&(-(cz1.sx1)-cx1.sy1.sz1).(l_1+l_2.(cx2.cz2-sx2.sy2.sz2)),\\
&-(l_2.cy2.(cz1.sx1.sy1+cx1.sz1).sz2)-l_2.cy1.sx1.(cz2.sx2+cx2.sy2.sz2)+\\
&(cx1.cz1-sx1.sy1.sz1).(l_1+l_2.(cx2.cz2-sx2.sy2.sz2)),\\
&l_2.cy2.cz1.sy1.sz2+l_2.cy1.(cz2.sx2+cx2.sy2.sz2)+sy1.sz1.(l_1+l_2.(cx2.cz2-sx2.sy2.sz2)),\\
&-(l_2.cy1.cy2.cz1.sx1.sz2)+l_2.sx1.sy1(cz2.sx2+cx2.sy2.sz2)-cy1.sx1.sz1.(l_1+l_2.(cx2.cz2-\\
&sx2.sy2.sz2)),l_2.cx1.cy1.cy2.cz1.sz2-l_2.cx1.sy1.(cz2.sx2+cx2.sy2.sz2)+\\
&cx1.cy1.sz1.(l_1+l_2.(cx2.cz2-sx2.sy2.sz2)),l_2.cy1.cy2.sz1.sz2-cy1.cz1.(l_1+\\
&l_2.(cx2.cz2-sx2.sy2.sz2)),-(l_2.cy2.(cx1.cz1-sx1.sy1.sz1).sz2)+(-(cz1.sx1.sy1)-cx1.sz1).(l_1+\\
&l_2.(cx2.cz2-sx2.sy2.sz2)),-(l_2.cy2.(cz1.sx1+cx1.sy1.sz1).sz2)+(cx1.cz1.sy1-sx1.sz1).(l_1+\\
&l_2.(cx2.cz2-sx2.sy2.sz2))\ \}^T
\end{aligned}$$

$$\begin{aligned}
\bar{r}_2 = \{&0,l_2.(-(cz2.sx2)-cx2.sy2.sz2),l_2.(cx2.cz2-sx2.sy2.sz2),l_2.sy2.sz2,-(l_2.cy2.sx2.sz2),\\
&l_2.cx2.cy2.sz2,-(l_2.cy2.cz2),l_2.(-(cz2.sx2.sy2)-cx2.sz2),l_2.(cx2.cz2.sy2-sx2.sz2)\ \}^T
\end{aligned}$$

The matrix form $\underline{R}_i$ of vector $\bar{r}_i$ is with respect to [6]:

$$\underline{R}_i = \begin{bmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{bmatrix}$$

The matrices $\underline{U}_i$ $(i=1,2)$ are:

$$\underline{U}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \underline{U}_2 = \begin{bmatrix} cy1.cz1 & , -(cy1.sz1) & , sy1 \\ cz1.sx1.sy1+cx1.sz1 & ,cx1.cz1-sx1.sy1.sz1, & -(cy1.sx1) \\ -(cx1.cz1.sy1)+sx1.sz1 & ,cz1.sx1+cx1.sy1.sz1, & cx1.cy1 \end{bmatrix}$$

The Jacobian matrix is then with respect to the terms (14) and (15):

$$\underline{J} = \begin{bmatrix} J_{1,1} & J_{1,2} & J_{1,3} & J_{1,4} & J_{1,5} & J_{1,6} \\ J_{2,1} & J_{2,2} & J_{2,3} & J_{2,4} & J_{2,5} & J_{2,6} \\ J_{3,1} & J_{3,2} & J_{3,3} & J_{3,4} & J_{3,5} & J_{3,6} \end{bmatrix}$$

The elements of the Jacobian matrix are:

$$\begin{aligned}
J_{1,1} &= 0\\
J_{1,2} &= l_2.cy2.cz1.sy1.sz2+l_2.cy1.(cz2.sx2+cx2.sy2.sz2)+sy1.sz1.(l_1+l_2.(cx2.cz2-sx2.sy2.sz2))\\
J_{1,3} &= l_2.cy1.cy2.sz1.sz2-cy1.cz1.(l_1+l_2.(cx2.cz2-sx2.sy2.sz2))\\
J_{2,1} &= -(l_2.cy2.(cx1.cz1.sy1-sx1.sz1).sz2)-l_2cx1.cy1.(cz2.sx2+cx2.sy2.sz2)+\\
&\quad (-(cz1.sx1)-cx1.sy1.sz1).(l_1+l_2.(cx2.cz2-sx2.sy2.sz2))\\
J_{2,2} &= -(l_2.cy1.cy2.cz1.sx1.sz2)+l_2.sx1.sy1.(cz2.sx2+cx2.sy2.sz2)-\\
&\quad cy1.sx1.sz1.(l_1+l_2.(cx2.cz2-sx2.sy2.sz2))\\
J_{2,3} &= -(l_2.cy2.(cx1.cz1-sx1.sy1.sz1).sz2)+(-(cz1.sx1.sy1)-cx1.sz1).(l_1+l_2.(cx2.cz2-sx2.sy2.sz2))
\end{aligned}$$

$$\begin{aligned} J_{3,1} = &-(l_2.cy2.(cz1.sx1.sy1+cx1.sz1).sz2)-l_2.cy1.sx1.(cz2.sx2+cx2.sy2.sz2)+ \\ &(cx1.cz1-sx1.sy1.sz1).(l_1+l_2.(cx2.cz2-sx2.sy2.sz2)) \end{aligned}$$

$$\begin{aligned} J_{3,2} = &\,l_2.cx1.cy1.cy2.cz1.sz2-l_2.cx1.sy1.(cz2.sx2+cx2.sy2.sz2)+ \\ &cx1.cy1.sz1.(l_1+l_2.(cx2.cz2-sx2.sy2.sz2)) \end{aligned}$$

$$J_{3,3} = -(l_2.cy2.(cz1.sx1+cx1.sy1.sz1).sz2)+(cx1.cz1.sy1-sx1.sz1).(l_1+l_2.(cx2.cz2-sx2.sy2.sz2))$$

$$J_{1,4} = -(l_2.cy1.sz1.(-(cz2.sx2)-cx2.sy2.sz2))+l_2.sy1.(cx2.cz2-sx2.sy2.sz2)$$

$$J_{1,5} = l_2.cx2.cy2.sy1.sz2+l_2.cy1.cz1.sy2.sz2+l_2.cy1.cy2.sx2.sz1.sz2$$

$$J_{1,6} = -(l_2.cy1.cy2.cz1.cz2)-l_2.cy1.sz1.(-(cz2.sx2.sy2)-cx2.sz2)+l_2.sy1.(cx2.cz2.sy2-sx2.sz2)$$

$$J_{2,4} = l_2.(cx1.cz1-sx1.sy1.sz1).(-(cz2.sx2)-cx2.sy2.sz2)-l_2.cy1.sx1.(cx2.cz2-sx2.sy2.sz2)$$

$$\begin{aligned} J_{2,5} = &-(l_2.cx2.cy1.cy2.sx1.sz2)+l_2.sy2.(cz1.sx1.sy1+cx1.sz1).sz2- \\ &l_2.cy2.sx2.(cx1.cz1-sx1.sy1.sz1).sz2 \end{aligned}$$

$$\begin{aligned} J_{2,6} = &-(l_2.cy2.cz2.(cz1.sx1.sy1+cx1.sz1))+l_2.(cx1.cz1-sx1.sy1.sz1).(-(cz2.sx2.sy2)-cx2.sz2)- \\ &l_2.cy1.sx1.(cx2.cz2.sy2-sx2.sz2) \end{aligned}$$

$$J_{3,4} = l_2.(cz1.sx1+cx1.sy1.sz1).(-(cz2.sx2)-cx2.sy2.sz2)+l_2.cx1.cy1.(cx2.cz2-sx2.sy2.sz2)$$

$$\begin{aligned} J_{3,5} = &\,l_2.cx1.cx2.cy1.cy2.sz2+l_2.sy2.(-(cx1.cz1.sy1)+sx1.sz1).sz2- \\ &l_2.cy2.sx2.(cz1.sx1+cx1.sy1.sz1).sz2 \end{aligned}$$

$$\begin{aligned} J_{3,6} = &-(l_2.cy2.cz2.(-(cx1.cz1.sy1)+sx1.sz1))+l_2.(cz1.sx1+cx1.sy1.sz1).(-(cz2.sx2.sy2)-cx2.sz2)+ \\ &l_2.cx1.cy1.(cx2.cz2.sy2-sx2.sz2) \end{aligned}$$

Conclusions

This paper presents a method for the Jacobian matrix determination used for the kinematic and kinetic simulation of redundant and non-redundant open chain mechanisms. The treated mechanisms have more then one DOF in each joint. The benefit of the presented method is simplicity. The calculation of required derivatives is in this case degenerated to the calculation of derivatives of the elementary transformation matrices with respect to the generalised co-ordinates. The whole algorithm is summarised in the equations (10)-(16) and can be numerically or analytically proceeded and is best suited for the use with complicated mechanisms as shown in the present example or special built robot wrists ,[7]. This method can naturally be used also for determination of the Jacobian matrix of classical mechanisms with just one DOF in each joint. The difference is in the equation (4) where just one matrix appears and vector $\bar{\varphi}_i$ degenerates to a scalar quantity.

References

[1] D. E. Whitney, "The Mathematics of Co-ordinated Control of Prosthetic Arms and Manipulators" *Journal of Dynamic Systems, Measurement, and Control, Trans. of the ASME*, 303-309, 1972.

[2] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control,* The MIT Press, 1981.

[3] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair mechanisms Based on Matrices", *Journal of Applied Mechanics, Trans. of the ASME,* 215-221, 1955.

[4] W. J. Vetter, " Matrix Calculus Operations and Taylor Expressions", *SIAM Review*, **Vol.15**, 352-369, 1973.

[5] W. J. Vetter, "Derivative Operations on Matrices", *IEEE, Trans. on Automatic Control,* 241-244, 1970.

[6] J. W. Brewer, "Kronecker Products and Matrix Calculus in System Theory", *IEEE, Trans. on Circuits and Systems,* **CAS-25**, 772-781, 1987.

[7] S. Ng and D. Wang, "Modelling and Control of a Flexible Spherical Wrist", *Robotica*, Vol.14, 155-163, 1996.

INTERACTION OF MODELLING AND NONLINEAR CONTROL DESIGN

P.C. Müller
Safety Control Engineering, University of Wuppertal
Gaußstr. 20, D-42097 Wuppertal, Germany
E-mail: mueller@wrcs1.urz.uni-wuppertal.de

Abstract. The methods of control design influence the requirements on the quality of a mathematical model of control systems. Applying the method of exact linearization and nonlinear decoupling by state feedback, a very accurate model is needed. On the opposite, for fuzzy control a good physical understanding is needed but a mathematical model is not required. In this contribution it will be illustrated by the design of a highly accurate position control of a robot, how the control design methods interact with system modelling.

Introduction

In this contribution it will be illustrated how the methods of control design influence the requirements on the quality of a mathematical model. As an example the position control of an industrial robot is considered.

Today's industrial robots are almost exclusively equipped with independent joint controllers for the position control, although the robot dynamics are highly nonlinear e.g. due to the position-dependent inertia moments and the coupling effects through the Coriolis moments. Therefore, for the improvement of the control performance different control concepts to decouple and compensate the nonlinear dynamics have been developed in the robotics research since years [1]. Such concepts can be subdivided into two groups. On the one hand multivariable controllers based on the multibody models of robot dynamics are designed, e.g. by the method of exact linearization by state feedback. On the other hand the structure of independent joint control is kept and the nonlinear effects are compensated through feedforwarding the "computed torques" obtained from desired trajectories or by the feedback of joint torques which are measured at the driven sides of each robot axis ("joint torque control") .

Of these two model-based methods the method of exact linearization is methodically more precise than the computed torque method, although it is more involving due to the on-line state feedback than the latter, in which the required feedforwarding torques can be computed off-line. In both cases, however, a complete knowledge of models is assumed. Parameter inaccuracies and incompletely known friction effects lead herewith to a need for a robust control design. These problems disappear however with the method of joint torque control. But for this control additional mesurement devices are required.

Avoiding the disadvantages of the mentioned control design methods - complete knowledge of a system model, sensitivity problems, additional measurements - the method of nonlinearity estimation and compensation can be applied alternatively [2]. This control design is based on a simple model and on usual measurements; the required informations for the compensation are estimated by observers. According to [3, 4], it will be shown that this approach leads to the design of robust position controllers for robots. In addition to the robustness, its decentralized structure offers the advantage that the concept of independent joint control can still be used.

Dynamical Model of Robots

An elastic joint robot together with its motor dynamics can be modeled according to [5] as:

$$\begin{aligned} \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{K}(\mathbf{q}-\mathbf{p}) &= \mathbf{0}, & \text{(1a)} \\ \mathbf{J}\ddot{\mathbf{p}} - \mathbf{K}(\mathbf{q}-\mathbf{p}) &= \mathbf{m}, & \text{(1b)} \\ \mathbf{T}\dot{\mathbf{m}} + \mathbf{m} &= \mathbf{G}\mathbf{u}, & \text{(1c)} \end{aligned}$$

in which $\mathbf{q}$ is the vector of joint coordinates, $\mathbf{M}(\mathbf{q})$ the positive definite mass matrix, $\mathbf{h}(\mathbf{q},\dot{\mathbf{q}})$ the Coriolis and centripetal as well as the gravitational forces. The vector $\mathbf{p}$ defines the motor angles relative to the gear ratios, $\mathbf{J}$ and $\mathbf{K}$ are the diagonal matrices which stand for the effective moments of inertia of the motors and the stiffnesses between motor and robot arm respectively; $\mathbf{m}$ are the drive torques of the motors, T the time constants and G the torque gains; $\mathbf{u}$ is the vector of input voltages of the motors. If the effects of friction are also considered, then they are included in terms of $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$.

Control Design by the Method of Exact Linearization

The application of the method [6] of exact linearization and nonlinear decoupling by state feedback to the robot control problem (1) has been shown in [3]. Using the joint coordinates $\mathbf{q}$ as output variables, a system decoupling can be explicitly given as

$$\mathbf{q}^{(5)} = \mathbf{a}_6(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}, \dot{\mathbf{p}}, \mathbf{m}) + \mathbf{M}^{-1}(\mathbf{q})\mathbf{K}\mathbf{J}^{-1}\mathbf{T}^{-1}\mathbf{G}\mathbf{u}, \tag{2}$$

where the function $\mathbf{a}_6$ is obtained from differentiations of (1). Choosing the drive voltages $\mathbf{u}$ as

$$\mathbf{u} = \mathbf{G}^{-1}\mathbf{T}\mathbf{J}\mathbf{K}^{-1}\mathbf{M}(\mathbf{q})(\mathbf{v} - \mathbf{a}_6), \tag{3}$$

a decoupled linear system

$$\mathbf{q}^{(5)} = \mathbf{v} \tag{4}$$

has been obtained. Here $\mathbf{v}$ is the new input vector. If, e.g., a desired trajectory $\mathbf{r}(t) = \mathbf{q}_d(t)$ has to be realized, it can thus be reached by the feedback

$$\mathbf{v} = \mathbf{q}_d^{(5)} + \mathbf{K}_4(\mathbf{q}_d^{(4)} - \mathbf{q}^{(4)}) + \mathbf{K}_3(\mathbf{q}_d^{(3)} - \mathbf{q}^{(3)}) + \mathbf{K}_2(\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}}) + \mathbf{K}_1(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_0(\mathbf{q}_d - \mathbf{q}), \tag{5}$$

in which $\mathbf{K}_i, i = 0, \cdots, 4$ are diagonal matrices with positive diagonal elements which determine the dynamics of each joint controllers.

The function $\mathbf{a}_6$ follows from differentiating $\mathbf{q}$ five times having regard to the equations (1a-c). The vector function depends on all state variables $\mathbf{q}, \dot{\mathbf{q}}, \mathbf{p}, \dot{\mathbf{p}}$ and $\mathbf{m}$ and looks very complicated. Even in the example of a one-axis robot the expression of $\mathbf{a}_6$ is extensive [4]. For a robot with six degree-of-freedoms the calculations have to be performed on a computer by formula manipulation programs.

As fruitful and beautiful the method of exact linearization and nonlinear decoupling by state feedback is from a theoretical point of view, as many problems appear in case of practical application [3]:

- Depending on the chosen output variables the so-called "zero dynamics" may appear which have to be asymptotically stable [6]. Here, using $\mathbf{y} = \mathbf{q}$ the problem does not appear. But if we use more realistic output variables $\mathbf{y} = \mathbf{p}$, then there is a problem of zero dynamics which is not asymtotically stable but only stable in the sense of Lyapunov.
- The computational effort for the calculation of the vector function $\mathbf{a}_6$ is generally big. Therefore symbolic computer programs have to be applied.
- Because the derivatives of nonlinear functions are needed, the functions are assumed to be sufficiently smooth. But in many applications this condition is not satisfied, e.g. in the case of Coulomb friction, backlash, impacts, saturations. Here, we have essentially to consider the problem of Coulomb friction.
- The knowledge of the exact model is required. Therefore, sensitivity problems may appear with respect to unmodelled effects or uncertain parameters. To overcome this problem special design methods are applied to guarantee robustness introducing an outer control loop [5].
- Due to the complete decoupling the amount of energy of the control may be high. For decoupling also "good" couplings of the system are counteracted unnecessarily.
- Generally the realization of the control (3, 5) is very cumbersome or even impossible. Usually all the state variables are fed back and have to be known. It is unrealistic to assume that all state variables are measured; this would be to expensive. The application of nonlinear state observers generally is a difficult problem, e.g. the separation principle does not hold which implies that the design of a stable closed loop control system is difficult using state observers.

Being aware of these problems the question arises if there exist alternative methods to design a feedback control for nonlinear dynamic systems avoiding the above-mentioned disadvantages.

Control Design by Nonlinearity Estimation and Compensation

To demonstrate an effective alternative control design we write down equations (1a -c) with respect to each single axis. The mass matrix is divided into a constant diagonal matrix of the mean values of the moments of inertia and a remaining position-dependent part,

$$\mathbf{M}(\mathbf{q}) = \mathbf{M}_0 + \Delta\mathbf{M}(\mathbf{q}). \tag{6}$$

The mean values are chosen with regard to a typical work space of the robot or along a desired trajectory. Summarizing further for each axis i all nonlinear terms in n_i,

$$n_i = \sum_j \Delta M_{ij}(\mathbf{q})\ddot{q}_j + h_i(\mathbf{q}, \dot{\mathbf{q}}), \tag{7}$$

equations (1a-c) can be separately considered for the single axis:

$$\begin{aligned} M_{0i}\ddot{q}_i + n_i + K_i(q_i - p_i) &= 0, &(8a)\\ J_i\ddot{p}_i - K_i(q_i - p_i) &= m_i, &(8b)\\ T_i\dot{m}_i + m_i &= G_i u_i. &(8c) \end{aligned}$$

This one-axis model can then be described correspondingly in state space. Leaving out the index i for the sake of brevity, the description

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{N}n + \mathbf{B}u, &(9a)\\ y &= \mathbf{C}\mathbf{x} &(9b) \end{aligned}$$

is obtained with the state vector $\mathbf{x} = [\,q \quad \dot{q} \quad p \quad \dot{p} \quad m\,]^T$ and the matrices

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{K}{M_0} & 0 & \frac{K}{M_0} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{K}{J} & 0 & -\frac{K}{J} & 0 & \frac{1}{J} \\ 0 & 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix}, \quad \mathbf{N} = \begin{bmatrix} 0 \\ -\frac{1}{M_0} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{G}{T} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T. \tag{9c}$$

Here the measurement of the joint coordinate q is assumed.

The objective of the position control of robots is the tracking of the joint coordinate $q(t)$ along a desired trajectory $r(t) = q_d(t)$ which is determined by path planning. The control error is

$$z = \mathbf{F}\mathbf{x} + Rr \tag{9d}$$

where

$$\mathbf{F} = [-1 \quad 0 \quad 0 \quad 0 \quad 0], \qquad R = 1. \tag{9e}$$

The design of each joint controller is based on the model (9). The nonlinearities and coupling effects, which are contained in n, will be compensated by the method of nonlinearity estimation and compensation [2]. The tracking control is reached through a feedforward, which is methodically determined using the method of disturbance rejection control [4]. Altogether an asymptotically stable control with

$$z(t) \to 0 \qquad \text{for} \qquad t \to \infty \tag{10}$$

is the desired design objective.

The time signal n of the nonlinearities and couplings is approximated by time functions, which are themselves solutions of an adequately selected linear dynamic system

$$n(t) \approx H_1 v_1(t), \qquad \dot{v}_1(t) = V_1 v_1(t). \tag{11a}$$

It was shown in [2] that this approximation can be best carried out with step functions, so that an integrator model is chosen in (11a):

$$H_1 = 1, \qquad V_1 = 0. \tag{11b}$$

The desired trajectory $r(t) = q_d(t)$ is assumed to be known. For the feedforward control however, the derived variables $\dot{q}_d(t)$, $\ddot{q}_d(t)$, $q_d^{(3)}(t)$, $q_d^{(4)}(t)$ and $q_d^{(5)}(t)$ are needed, which although are theoretically available too, but do not exactly correspond to the derivations of the actually requested trajectory due to possible disturbance influences. Therefore, they are estimated by an observer which is constructed like (11a) as well, see [4]:

$$r(t) \approx \mathbf{H}_2\mathbf{v}_2(t), \qquad \dot{\mathbf{v}}_2(t) = \mathbf{V}_2\mathbf{v}_2(t). \tag{12a}$$

The approximation is based on step and ramp functions, so that

$$\mathbf{H}_2 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0], \qquad \mathbf{V}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{12b}$$

is selected.

The design of observers for the estimation of the signals n and $\dot{r}, \ddot{r}, r^{(3)}, r^{(4)}, r^{(5)}$ as well as $\dot{q}$, p, $\dot{p}$ and m is based on a linear system, which is obtained by inserting (11, 12) in (9):

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{v}_1 \\ \dot{\mathbf{v}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{N}H_1 & 0 \\ 0 & V_1 & 0 \\ 0 & 0 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ v_1 \\ \mathbf{v}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \\ 0 \end{bmatrix} u, \tag{13a}$$

$$\begin{bmatrix} y \\ r \end{bmatrix} = \begin{bmatrix} \mathbf{C} & 0 & 0 \\ 0 & 0 & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ v_1 \\ \mathbf{v}_2 \end{bmatrix}, \tag{13b}$$

$$z = [\mathbf{F} \quad 0 \quad R\mathbf{H}_2] \begin{bmatrix} \mathbf{x} \\ v_1 \\ \mathbf{v}_2 \end{bmatrix}. \tag{13c}$$

This extended system with the given system matrices is completely observable. Therefore, an observer can be designed according to one of the usual methods. It is shown at the same time that the observer can be separated into two parts for $n, \dot{q}, p, \dot{p}, m$ and $r, \dot{r}, \ddot{r}, r^{(3)}, r^{(4)}, r^{(5)}$:

$$\begin{bmatrix} \dot{\hat{\mathbf{x}}} \\ \dot{\hat{v}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{L}_x\mathbf{C} & \mathbf{N}H_1 \\ -L_{v1}\mathbf{C} & V_1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{v}_1 \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u + \begin{bmatrix} \mathbf{L}_x \\ L_{v1} \end{bmatrix} y, \tag{14}$$

$$\dot{\hat{\mathbf{v}}}_2 = (\mathbf{V}_2 - \mathbf{L}_{v2}\mathbf{H}_2)\hat{\mathbf{v}}_2 + \mathbf{L}_{v2}r. \tag{15}$$

The desired estimated values are obtained from (14, 15), e.g. $\hat{n} = H_1\hat{v}_1$.

With the estimated variables $\hat{\mathbf{x}}$, $\hat{v}_1$ and $\hat{\mathbf{v}}_2$ a feedback

$$u = -\mathbf{K}_x\hat{\mathbf{x}} - K_{v1}\hat{v}_1 - \mathbf{K}_{v2}\hat{\mathbf{v}}_2 \tag{16}$$

is constructed. The gain matrix $\mathbf{K}_x$ of the state feedback can be determined by standard methods like pole assignment (the complete controllability of the matrices ($\mathbf{A}$, $\mathbf{B}$) is fulfilled in the present case). The gain matrix K_{v1} for the compensation of the nonlinearities and coupling effects and the gain matrix $\mathbf{K}_{v2}$ for the feedforwarding of the desired trajectory are determined from the equations [2, 4]

$$(\mathbf{A} - \mathbf{B}\mathbf{K}_x)\mathbf{X}_1 - \mathbf{X}_1V_1 - \mathbf{B}K_{v1} = -\mathbf{N}H_1, \tag{17a}$$

$$\mathbf{F}\mathbf{X}_1 = 0 \tag{17b}$$

and

$$(\mathbf{A} - \mathbf{B}\mathbf{K}_x)\mathbf{X}_2 - \mathbf{X}_2\mathbf{V}_2 - \mathbf{B}\mathbf{K}_{v2} = 0, \tag{18a}$$

$$\mathbf{F}\mathbf{X}_2 = -R\mathbf{H}_2. \tag{18b}$$

Here the problem is also separated into two parts. $\mathbf{X}_1$ and $\mathbf{X}_2$ are auxiliary matrices, which characterize the stationary behaviour of $\mathbf{x}(t)$ depending on $v_1(t)$ and $\mathbf{v}_2(t)$. However, the solutions of K_{v1}, $\mathbf{K}_{v2}$ are of primary interest. They result in

$$K_{v1} = -\frac{1}{K}K_{x3} - (\frac{1}{G} + K_{x5}), \qquad \mathbf{K}_{v2} = \begin{bmatrix} -K_{x1} - K_{x3} \\ -K_{x2} - K_{x4} \\ -\frac{M_0}{K}K_{x3} - (\frac{1}{G} + K_{x5})(M_0 + J) \\ -\frac{M_0}{K}K_{x4} - \frac{T}{G}(M_0 + J) \\ -(\frac{1}{G} + K_{x5})\frac{M_0J}{K} \\ -\frac{TM_0J}{GK} \end{bmatrix}^T. \tag{19}$$

With that, the controller is finally obtained.

The proposed control design method has been demonstrated in [2-4]. In [7] the control design of nonlinearity estimation and compensation has been applied to a PUMA 560 robot to improve the conventional independent joint control. The end-effector had to track a circular path in the 3-dimensional space. To evaluate the tracking performance, the absolute position error - the distance between the nominal and actual end-effector position - has been used as a performance criterion. In all cases where the model (1) is uncertain with respect to Coulomb friction and/or load the proposed controller gave much better results than the above-mentioned nonlinear controller according to the method of exact linearization.

Robustness

While the control (3, 5) is sensitive in general, the control (16) is structurally robust against unmodelled or uncertain effects. If the parameters in the system description (1) are inaccurate, e.g. due to the varied loads, or unknown friction torques appear, the real system behaviour is then described with a modified model

$$\begin{aligned} \mathbf{M}'(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}'(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{K}(\mathbf{q}-\mathbf{p}) &= \mathbf{0}, && (20a)\\ \mathbf{J}\ddot{\mathbf{p}} - \mathbf{K}(\mathbf{q}-\mathbf{p}) &= \mathbf{m}, && (20b)\\ \mathbf{T}\dot{\mathbf{m}} + \mathbf{m} &= \mathbf{G}\mathbf{u}. && (20c) \end{aligned}$$

Whereas the controller (3-5) is still based on the nominal model (1) so that the mismatch problem with its possible sensitivity is present, the controller (16) can completely react upon the modified system behaviour. Let

$$\mathbf{M}'(\mathbf{q}) = \mathbf{M}_0 + \Delta\mathbf{M}'(\mathbf{q}), \tag{21}$$

then one has only to replace the term n_i in the joint axis description (8) by

$$n_i' = \sum_j \Delta M_{ij}'(\mathbf{q})\ddot{q}_j + h_i'(\mathbf{q},\dot{\mathbf{q}}). \tag{22}$$

As the design of the controller (16) is based on the fact that n_i - and thus n_i' - are to be interpreted as unknown variables, and hence to be estimated by the observer (14), the controller fulfills its task also by varied model data. The controller (16) is structurally robust against parameter inaccuracies and unmodelled effects.

Conclusion

The method of nonlinearity estimation and compensation has been proved to be a good alternative to the method of exact linearization and nonlinear decoupling by state feedback, especially in case of practical applications. It is an efficient approach for the design of robust position control of robots. In addition to its robustness its decentralized structure offers additional advantages such that the concept of independent joint control can further be used, even in an improved manner. The requirements on the modeling of the robot dynamics are very low. The control design is easily performed because it is based on linear system theory only. Practical applications in robot control have shown the efficiency of the proposed control.

This example of robot control shows the interaction between the requirements on the mathematical modelling and the design method applied to the nonlinear control problem. The method of nonlinearity estimation and compensation is only based on linear models (9) of the dynamics of each joint and link independently of the nonlinear and coupling effects. These effects are estimated by state observers and counteracted in each joint. Compared with this simple but effective control design the method of exact linearization and nonlinear decoupling needs a very exact mathematical model without guaranteeing good results in practical applications.

Completely different approaches are fuzzy control and the application of artificial neural networks which has been not discussed here. Both approaches do not need mathematical models. But fuzzy control requires a good physical understanding to define the necessary rules of operation. They are difficult to obtain for multi-degree-of-freedom systems. Only for simple robot systems fuzzy control may be applied successfully but for real industrial robots the complexity of the problem still prevents a highly accurate position control by the fuzzy logic approach. For neural networks the phase of learning is the critical point. Either it is based on a good mathematical model - and then it depends on parameter uncertainties or unmodelled effects - or it has to be learned in real life which may lead to good accuracy of repeatability but it does not guarantee high absolute accuracy. Therefore, the proposed method of nonlinearity estimation and compensation combines the advantages of simple modelling, easy control design by methods of linear system theory, and efficient results.

References

1. Spong, M.W., Lewis, F.L., and Abdallah, C.T.(eds.), Robot Control - Dynamics, Motion Planning and Analysis. IEEE Press, New York, 1992.
2. Müller, P.C., Schätzung und Kompensation von Nichtlinearitäten mit Störgrößenbeobachtern. In: Entwurf nichtlinearer Regelungen (Ed. Engell, S.) Oldenburg, München-Wien, 1995.
3. Müller, P.C, Non-Linear Robot Control: Method of Exact Linearization and Decoupling by State Feedback and Alternative Control Design Methods. Appl. Math. and Comp. Sci, 5 (1995), 359-371.
4. Hu, R. and Müller, P.C., Independent Joint Control: Estimation and Compensation of Coupling and Friction Effects in Robot Position Control. J. Intelligent and Robotic Systems, 15 (1996), 41-51.
5. Spong, M.W. and Vidyasagar, M.: Robot Dynamics and Control. Wiley, New York, 1989.
6. Isidori A., Non-Linear Control Systems (2nd ed.). Springer, Berlin-Heidelberg, 1989.
7. Hu, R. and Müller, P.C., Tracking Control Design for Robots Using the Method of Nonlinearity Estimation and Compensation. Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, Beijing, 1996, 3107-3112.

ENERGY FLOW MODELING OF MECHATRONIC SYSTEMS VIA OBJECT DIAGRAMS

M. Otter[1]**, and H. Elmqvist**[2]
[1]Institute for Robotics and System Dynamics, DLR, Postfach 1116,
D-82230 Wessling, Germany, email: Martin.Otter@DLR.de
[2]Dynasim AB, Research Park Ideon, S–223 70 Lund, Sweden, email: Elmqvist@Dynasim.se

Abstract. Object diagrams are generalizations of block diagrams and of bond graphs, and allow modeling of input/output signal flow as well as energy flow. It is shown how object diagrams can be used for the modeling of mechatronic systems, consisting of 3-dimensional mechanical systems, electrical circuits, drive trains, control blocks etc. Emphasis is given to the description of energy flow between different components. Contrary to domain-specific packages, all the details of an object diagram, such as equations of elements, graphical layout of icons or submodels, are defined in a "soft" way, not built into the package, and can be directly accessed by the user, to either use it as a template for user-defined model types or to make modifications that are necessary for the problem at hand.

1. Introduction

The modeling of mechatronic systems is difficult because such systems consist of components from different domains, especially they contain mechanical, electrical, hydraulic, control system and drive train devices. It seems unsuitable to base modeling on a domain specific software package like a multibody program or an electric circuit program, because these packages usually have quite limited modeling capabilities in the other domains. What is needed is a neutral methodology as a *basis* for multi-domain modeling. This is the topic of this paper.

A very popular neutral description form are *block diagrams* to model the signal flow of causal input/output blocks. Block diagrams are well suited for the modeling of *control systems*. For modeling of physical systems, such as electrical circuits or mechanical systems, block diagrams are the wrong tool, because the natural description form of such systems is not an input/output behavior, and it needs great effort to manually transform into this form. See for example the simple electric circuit in figure 1. Given the physical device, it is straightforward to draw the electric circuit diagram shown on the left side of figure 1. Transforming the electric circuit into the block diagram in the middle of figure 1 needs some work and is not practical for larger circuits. Furthermore, the physical insight provided by the electric circuit diagram is lost in the block diagram.

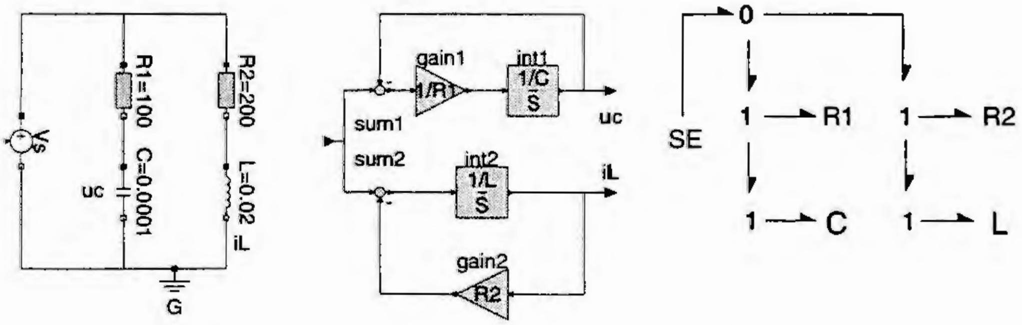


Figure 1: Electric circuit as block diagram and as bond graph.

Block diagrams are inappropriate for the modeling of physical systems because they are designed to visualize the *signal flow*. However, the major and common characteristic of all physical systems is the *flow of energy* which cannot be mapped into signal flow in a natural way since energy flow is acausal.

These considerations lead directly to *bond graph* modeling [9, 7], another neutral model description form. Bond graphs are designed to model the energy flow of systems, i.e., they describe the main property of physical systems. Due to the reduction of system descriptions in different domains down to the same basic entities, like resistors, transformers, etc., the bond graph methodology has greatly enhanced the understanding of physical system modeling and is therefore also very well suited for basic modeling courses.

On the other hand, the advantage of bond graphs to describe systems by few basic elements hinders at the same time its *practical applicability*: Modelers want to think in terms of their "usual" entities. The basic components of electrical engineers include "capacitor" and "electrical transformer", whereas mechanical engineers prefer to think in terms of "rigid bodies" and "ideal joints", although in bond graph view these are similar elements. See e.g. the bond graph of the circuit in figure 1. Although, the bond graph in the right side of the figure has the same structure as the circuit diagram in the left side, it is quite obvious that engineers prefer the electric circuit diagram.

Furthermore, it turns out that all domains have unique aspects which need specific treatment as will be explained in more detail below. The bond graph methodology is not flexible enough in such cases and often needs enhancements which destroy the simple basic structure. Finally, the bond graphers are very much focused on the high level description and don't have automatic and efficient algorithms to transform the acausal bond graph down to state space form which is needed in order to apply standard integration solvers.

A third neutral description form is *object-oriented modeling*, developed by Hilding Elmqvist in the late seventies [4] as a general tool for all kinds of systems described by differential and algebraic equations. In the beginning, object-oriented models have been defined in textual form by a specific language, like Dymola [4, 2] or Omola [1]. The graphical view of such models are similiar to functional diagrams [3], often used in engineering to visualize the decomposition of a system into components. In object-oriented modeling these diagrams are sometimes called *object diagrams*. An early version appeared in the experimental modeling system HIBLIZ [5]. Object diagrams are simple to understand and are quite general. It is useful to build object diagram model libraries, in order to encapsulate knowledge and to allow easy reusability. Due to the generality of object diagrams, it might not always be obvious how to apply them in a suitable way to a specific domain to build up model libraries. To overcome this difficulty, it will be explained how object diagrams can be used to model *energy flow* which leads to a general guideline how to apply object diagrams for physical system modeling. Recently, object diagrams became especially attractive for the modeling community due to the availability of comfortable object-diagram editors with easy-to-use graphical user interfaces [2].

2. Object Diagram

Object diagrams are used to describe systems according to their decomposition into components. An example of a simple object diagram is shown in figure 2. Each rectangle corresponds to a physical component. In an actual drawing, each object is represented by an icon which should give an intuitive understanding of the component. The objects are connected together by lines which typically represent the actual, physical connection between components. The interface points of objects at which they can be connected together are called *cuts*, alternatively *ports*, *terminals* or *attachment points*. Variables used in a cut define the possible interaction with other objects. Connections impose constraints on these variables. An object is described in a declarative way by equations which use only local and cut variables of the corresponding object. There are two types of cut variables: *across* and *through* variables. Corresponding across variables at a connection point are identical, whereas the sum of corresponding through variables is zero. Object diagrams are hierarchical, i.e., an object may contain another object diagram. It turns out that these few basic elements are already sufficient to model nearly every type of physical system. A block diagram is just a special case of an object diagram, having only across variables with a specific causality in a cut.

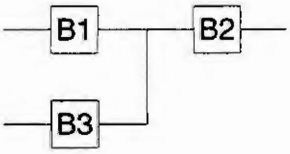
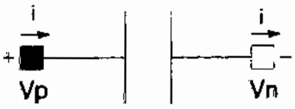


Figure 2: Simple object diagram

A capacitor is a typical example for a component in an object diagram, see figure 3. The two cuts of



```
model class Capacitor
    parameter C
    cut p (Vp / i), n (Vn / −i)
    local u
    Vp − Vn  = u
    C*der(u) = i
end
```

Figure 3: Electrical capacitor as a component of an object diagram.

a capacitor object are the two pins at which the element can be connected together with other electrical components. Cut variables are the electric potential V and the current i flowing into/out of the pin. On the right side of the figure, the equations of the capacitor as a function of the local variables and the cut variables are given in Dymola notation. Note, that all cut variables to the left of the slash are defined to be across variables, whereas all cut variables to the right of the slash are defined to be through variables.

It now becomes clear that the electric circuit in the left side of figure 1 can be interpreted as an object diagram. In fact, the object diagram editor of Dymola was used to model this circuit. Contrary to the more specialized bond graph methodology, efficient symbolic transformation algorithms exist to transform the declarative description of an object diagram down to state-space form [4]. That is, object diagrams can be practically applied to large and complex system models.

3. Energy Flow Diagram

As already mentioned, object diagrams are easy to understand and are quite general. However, it is often not obvious how to select the cut variables, and the nice interpretation of energy flow present in the bond graph methodology is missing. In this section these difficulties will be resolved by using the basic idea of bond graphs, simplify it and apply it to object diagrams.

The idea is to interpret the lines connecting blocks in an object diagram as energy flow. It is well known that energy flow (= power) can be described in all physical domains as the product of two variables

$$P = \frac{dE}{dt} = e \cdot f$$

In bond graph terminology these variables are called effort and flow variable, respectively, and it is always possible to interchange the role of the two variables. This is different for object diagrams, where e is the *potential*, $f = dm/dt$ is the *flow* defined to be "carrier/time", and m is the *carrier*, i.e., the "particle" which transports the energy.

It is *defined* that potential variable e is an across variable, i.e., at energy flow connection points all corresponding potential variables are identical. Usually, every domain has only one mechanism to transport energy between two components, e.g, wires in electrical circuits, flanges in mechanical systems, surfaces in heat flow systems. As a consequence, the selection of potential variables is usually unique. Again note, this is in contrast to bond graph methodology, where a bond not necessarily corresponds to a connection between components and therefore the role of effort and flow variables can be interchanged.

When energy flows together at one point, and no energy is stored in this point, the sum of the energy flows must be zero, i.e.,

$$0 = \sum_i P_i = \sum_i e_i \cdot f_i = e \cdot \sum_i f_i$$

Since, by definition, the potentials at connection points are identical, if follows that the sum of the flow variables f_i must vanish at connection points, i.e., flow variables must be treated as through variables in an object diagram. This property is well know in every domain, e.g., for electrical circuits it is Kirchhoffs current law, for mechanical systems it is Newton's *actio=reactio*. Note, that this property is not a new basic law but is derived directly from the energy conservation law.

For several domains the three describing variables of energy flow are collected together in table 1. For mechanical systems, *bond graph* literature uses the reversed view, i.e., velocity is used as flow variable and force is used as effort variable. For object diagrams this choice is not possible, because whenever *components* like bodies or joints are connected rigidly together, the velocities of the components at connection points are identical, i.e., velocity has to be a potential variable. "Hydraulic" means incompressible fluid

Type	Potential e	Carrier m	Flow f
electric	V: electrical potential	q : charge	$\dot{q} = i$: charge flow = current
translational	v : velocity	p : momentum	$\dot{\mathbf{p}} = \mathbf{f}$: momentum flow = force
rotational	ω: angular velocity	$\mathbf{L}$: angular momentum	$\dot{\mathbf{L}} = \tau$: momentum flow = torque
hydraulic	p : pressure	V: volume	$\dot{V}$: volume flow
adiabatic flow	h : enthalpy	m: mass	$\dot{m}$: mass flow
heat	T: temperature	S : entropy	$\dot{S}$: entropy flow
chemical	μ : chemical potential	N: particle	$\dot{N}$: particle flow

Table 1: Energy flow variables in various domains.

flow with low velocity. "Adiabatic flow" characterices the special case of fluid flow with low velocity where heat flow is neglected. As can be seen, entropy is the carrier of heat flow, as is charge for flow of electrical energy. This leads to the nice analogy that entropy is similiar to electrical charge since it is the carrier or particle transporting the energy.

In the following, object diagrams will be called ***energy flow diagrams*** provided that all the lines connecting objects correspond to energy flows. It is now easy to give some general guidelines for the construction of ***object diagrams***: If signals are the major transport mechanism between components, use a block diagram description. If energy is the major transport mechanism between components, use an energy flow description. As describing variables in cuts use the corresponding potential and flow variables according to table 1 above. For several domains this scheme must be slightly modified due to some pecularities of the specific domain.

Heat flow

For heat flow the described scheme suggests temperature and entropy flow as cut variables. However, this is not a good choice as pointed out in [10]: For the simplified lumped-heat-capacity method of heat flow analysis [6], the two most important elements are heat resistance with the equation $\dot{Q} = kA/\Delta x(T_1 - T_2)$, and heat capacitance with the equation $\dot{Q} = cm\dot{T}$, where $\dot{Q}$ is the heat flow and T is the temperature. Both equations are linear functions of the temperature. Since $\dot{Q} = T\dot{S}$, both equations can be transformed into an equivalent description form containing temperature and entropy flow, where the equations are non-linear functions of T. Furthermore, the common boundary condition of perfect insulation can be easier expressed in terms of heat flow ($\dot{Q} = 0$) instead of entropy flow. Consequently, it is better to use temperature and heat flow instead of temperature and entropy flow as cut variables, as it is usually also done in the thermodynamic literature. In this case heat flow is a through variable. In bond graph literature this choice leads to the so-called "pseudo bond graph" [7].

Mechanical Systems

The table above suggests velocity and force, as well as angular velocity and torque as cut variables. However, both in 3D-mechanics as well as for special cases, like positioning drive trains (= 1D rotational mechanical systems) the position of a component is important and it is therefore necessary to use positional variables in a cut. Still, this is not sufficient: For example, the simple drive train in figure 4 consists of a common configuration of two shafts with inertia connected rigidly together by a gear box.

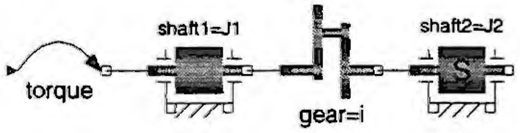


Figure 4: Object diagram of a drive train with two shafts connected by a gear box.

Assume first that both shafts would be described by an equation of the form "$J\ddot{\varphi} = \tau_a - \tau_b$", where φ is the absolute angle of a shaft and τ_a, τ_b are the cut-torques acting at the driving and at the driven side of a shaft. This means that the angle φ of both shafts would be used as state variables. However, since the two shafts are rigidly connected together a constraint equation exists between these two state variables, i.e., this system has a DAE-index > 1 and can therefore not be transformed to state space form by purely algebraic transformations.

Alternatively, angle φ, angular velocity ω and angular acceleration α can be used as cut variables and one of the two shafts is described by equation "$J\alpha = \tau_a - \tau_b$" and the gear box is described by "$\varphi_a = i\varphi_b, \quad \omega_a = i\omega_b, \quad \alpha_a = i\alpha_b, \quad \tau_b = i\tau_a$" where i is the gear ratio. Since the dynamic equation of one of the two shafts is now only an algebraic equation, the overall system of equations is well-defined and can be transformed to state space form. Consequently, for such types of systems it is better to transport not only position, but also velocity and acceleration through the mechanical cuts.

4. Example

The power of object diagrams and in particular of energy flow diagrams is demonstrated with a quite complicated example. In figure 5 the detailed model of a robot from [8] is shown in form of an object diagram. Dymola was used to build the object diagram and to transform the whole system to state space form with 66 states. With exception of the controller, the robot is described by an energy flow diagram, because all the connections correspond to energy flows. In the right part of the figure, the 3D mechanical construction of the robot is shown, i.e., it is a connection of ideal joints and rigid bodies containing animation information. The joint axes are driven by drive lines, called "axisX", which are mounted at the inner sides of the joints. All the drive lines have the same structure which is given in the left side of the figure. One drive line consists of a controller, an electrical motor and a gear box. Furthermore, the desired angular acceleration is integrated two times to arrive at the desired angular velocity and the desired joint angle. The controller is modelled as a block diagram consisting of basic transfer function blocks. The electrical motor is an energy flow diagram in the form of an electrical circuit. The gear box is also an energy flow diagram with 1D rotational mechanical components.

5. Conclusions

Component oriented modeling in form of object diagrams is a general and useful methodology to graphically model complex systems. Block diagrams and energy flow diagrams are special cases of object diagrams. For physical systems, energy flow is the basic transportation mechanism between components. Therefore, variables describing energy flow are good candidates for the interface variables of components. As shown by example, peculiarities in some domains may lead to some modifications of this scheme. Due to the generality of object diagrams, such changes can always be made and are uncritical. This is in contrast to bond graph methodology which often require non-standard generalizations.

References

[1] Andersson, M., *Omola — An Object-Oriented Language for Model Representation*, Licenciate thesis TFRT-3208, Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden, 1990.

[2] Elmqvist, H., Brück, D., and Otter, M., *Dymola - User's Manual*, Version 3.0, Dynasim AB, Lund, Sweden, 1996.

[3] Cellier, F.E., *Continuous System Modeling*, Springer Verlag, 1991.

[4] Elmqvist, H., *A Structured Model Language for Large Continuous Systems*, Ph.D. Dissertation, Report CODEN: LUTFD2/(TFRT-1015), Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden, 1978.

[5] Elmqvist, H., and Mattson, S.E., *Simulator for Dynamical Systems Using Graphics and Equations for Modeling*, IEEE Control Systems Magazine, January, 1989, pp. 53-58.

[6] Holman, J.P., *Heat Transfer*, 7th ed., McGraw-Hill, New York, 1992.

[7] Karnopp, D.C., Margolis, D.L., and Rosenberg, R.C., *System Dynamics: A Unified Approach*, John Wiley, 2^{nd} edition, 1990.

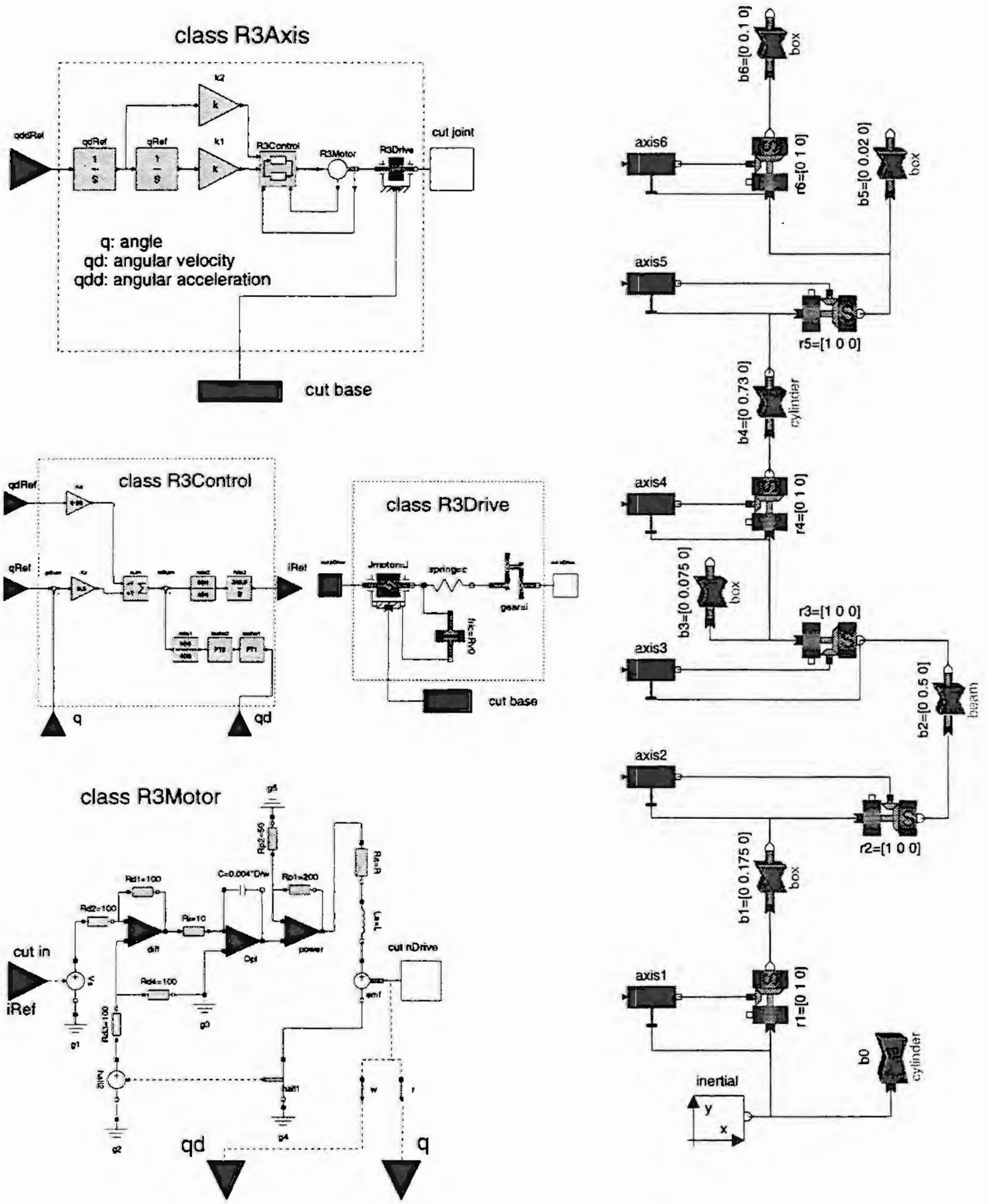


Figure 5: Detailed robot model as object diagram

[8] Otter, M., *Objektorientierte Modellierung mechatronischer Systeme am Beispiel geregelter Roboter*, Ph.D.-Dissertation, Fortschritt-Berichte VDI, Reihe 20, Nr. 147, VDI-Verlag Düsseldorf.

[9] Paynter, H.M., *Analysis and Design of Engineering Systems*, MIT Press, Cambridge, Mass., 1961.

[10] Pieters, S., Personal communication to M. Otter.

OBJECT-ORIENTED MODELLING OF CONTROL ENGINEERING DATA

U. von Döllen and M. Schlothane
University of Bochum, Dept. of Mechanical Engineering, Control Group
D-44780 Bochum, Germany

Abstract. Most people involved in software engineering acknowledge the value of object-oriented programming techniques, but in spite of this principally positive judgement the consequent use of object-oriented methods is linked with some problems in practice. This paper will describe these general problems especially focused on object-oriented modelling of control engineering data and discuss possible solutions. The mathematical modelling of a neuro-fuzzy based control system for a gas supply network is demonstrated as an example for application.

1. Introduction

Integration of data and functions is an important feature of the object-oriented methodology, the object-oriented programming in particular. The main attention of implementation in structured programming lays on functionality, since there is a clear separation of data and functions. In object-oriented programming the functionality is linked directly to the available data and thus focuses on the importance of data. Data and functionality are capsulated inside of objects. The advantages of this methodology are generally acknowledged and described in detail in [1][2][3][4].

Abstraction of a concrete problem to a number of objects, convenient definition of the objects' interfaces as well as definition of the relations between different objects, is the most difficult and most important task for the software-engineer. For definition of objects the classical object-oriented method focuses on the available data. In practice, this principally convenient objective can not to be realised easily, because complete integration of functionality is difficult at least of two reasons.

The first reason is reusability of existing software which, in practice, is highly recommendable if not necessary. Especially in control engineering there are many software tools available for simulation, analysis, and design that provide a wide range of different instruments. However, this functionality is separated from the data to be processed. Reusing this functionality by integration into objects together with the data to be processed, requires a very high programming effort using object-oriented methods. This effort is hardly to do and not efficient. An obvious alternative to a pure object oriented concept is combining object-oriented methods on existing data with reusing existing functionality by programming suitable interfaces. To enable such a combination, we defined data objects whose internal functionality is restricted to a few elementary operations. By integration of corresponding interfaces it is possible to use external functionality of different software packages.

The second important reason is the hardly to reach completeness in integration of functions. A certain class always consists of structured data and corresponding operations that can be applied to them. Application of a huge number of operations is possible to many structured data types. Decreasing the complexity of the data type increases the number of operations. For reaching completeness of operations applicable to such data types object-oriented proceeding offers two possibilities. Either one class with an immense number of (internal) operations is designed or an extensive class hierarchy or a collection of so-called friend classes has to be built, in which all classes are derived from a base class with one single structured data type but containing just a small number of operations. In both cases the pure object-oriented concept guarantees that after compilation of the corresponding library, respectively the application program, all operations of one class are defined and should not be modified. That means all operations to the defined data objects are definitively fixed. Corresponding to the object-oriented concept only operations to one specific data type are possible. Yet, in practice, it is desirable to apply one operation to different structured data types. Principally this requires the design of a new structured data type for all operations that are intended to link already defined structured data types by a functional connection.

To solve the described problems, we suggest a new modified object-oriented concept. The main idea is the development of application systems that process different kind of data from different types of hardware resources. The system engineer should be enabled to concentrate on the solution of the real problem. In this context it must be guaranteed that a variety of algorithms and functions can be applied to the data. For example, different freely configurable visualisations of all data should be possible.

For definition of the flexible connections between data and operations an object-oriented concept must be easily understandable and intuitively to use. Besides building of higher level management systems requires the possibility to start operations and algorithms at any point of time. For that reason an object oriented solution is needed that connects a number of time period related algorithm executions to a number of algorithm executions related to discrete time points. All these reasons lead to a definition of a new, object-oriented concept.

2. Data Objects

Process modelling as well as the procedure of designing control systems becomes significantly easier - or is at least made possible - by using suitable software tools for analysis and simulation. It is, therefore, desirable for the engineer to be able to link all software modules by a common interface. Moreover, such a kind of interface would simplify very much the information exchange between several users of different programs. Apart from that, the coupling of different program packages to a control engineering database is generally useful. The access to such a database requires, of course, the definition of a general data interface and a suitable organisation of data. No matter whether you choose a direct or an indirect coupling, the use of object-oriented methods is advisable for the computational implementation. The present issue is to develop a class library that solves the above-mentioned problems by means of object-oriented principles.

The base classes of this class library consist of simple mathematical description classes like scalar, vector or matrix. Based on these elementary classes, more complex classes for system description are established (frequency response, dynamic response, root-locus, state space description, etc.). The core of the class library is the mathematical description. A specialisation of derived object-classes for different application areas always refers to this core and therefore enables a data exchange at least at this lowest level of abstraction.

To achieve a maximum of flexibility for data processing, a completely object-oriented data interface is necessary. For this reason, the implemented data management is based on data-objects that enable a simple coupling between all system components. Seen from the point of the information processing system components, data access is always realised via public methods of the data objects. The functions that enable a dynamic connection to different data sources are integrated in the data object definition. Besides virtual data sources (i. e. data are held dynamically in memory) and a link to ASCII-files, a database interface (Microsoft Open Database Connectivity - ODBC) is implemented [5]. A main goal of object-oriented data management with the help of data objects is to capsulate hardware-caused functionality

(Figure 1). From the point of the application software, data access is realised via the external functionality of the data objects. Apart from simple read and write operations, a restricted number of functionality for data analysis and data validation is available on this level. (Consistence, plausibility, etc.) The real access to the database, ASCII-file, etc. is realised by the internal functionality of the data objects, i. e. invisible for the application software.

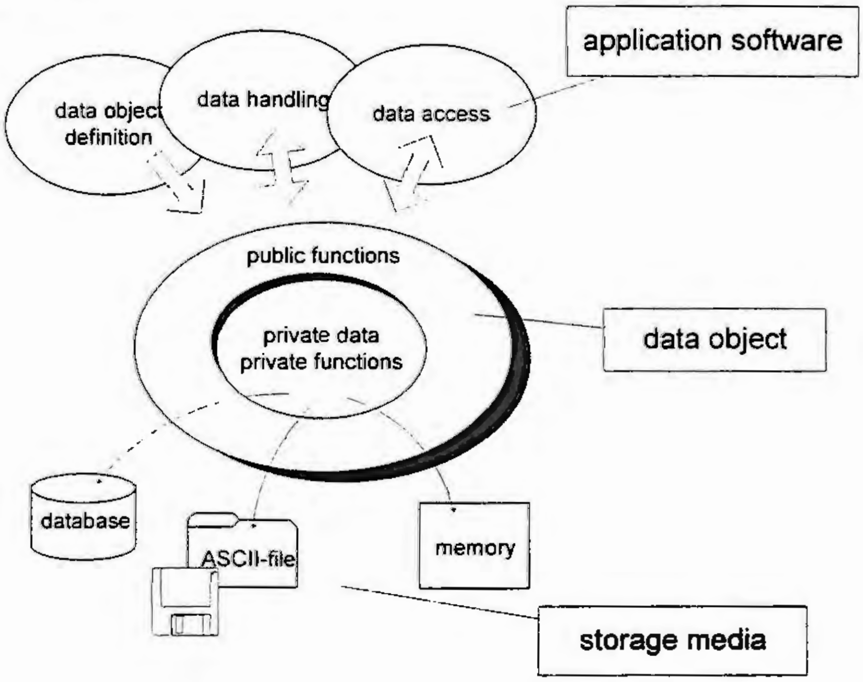


Figure 1. Capsulation of data and functions

3. Operation Objects

The class of data objects described above includes no functions to execute operations on the containing data. Functionality for data processing - like simple data manipulation or execution of algorithms on data - is necessary for every application. Usually this functionality is implemented in derived classes. This inevitably leads to specific solutions. Therefore, derivation from a general data object class, as it is usual in object-oriented modelling, is not suitable in this case. Instead, the data objects are introduced as isolated and independent data types with basic functionality for external use as described before. A separation of data types and operations applied to these data types is made to avoid the problem of recurrently subclassing the data objects base class. Spoken in object-oriented terms an abstract operation is attached to the abstract data type.

All operations applied to abstract data types - e.g., the execution of control engineering algorithms - are defined as functions inside abstract operations. For this reason a new algorithm needs no modification nor extension in the functionality of an abstract data type, i. e. subclassing the data object class. Thus algorithms can freely be defined as abstract operations. Abstract operations do not need to be changed if a new abstract data type is created. That is why the user of a class hierarchy based on this general concept is completely free in designing abstract data types, too. The presented approach describes data and operations completely separated from each other. Therefore, definition and analysis of possible links between such objects are intentionally not integrated in one of the abstract data types itself. This is done by a separate class for building up link structures between data objects and operation objects. Thereby a basis is given for unrestricted flexible definition of data types as well as operation types and application of different operations to different data. An example might be the application of a mathematical operation to data objects that contain different time axes. Thus, the overall approach regards the treatment of data by an operation as a black box. I. e., what data are to be processed in what way and with which operations is not yet fixed by definition of abstract data types and abstract operations. This way a maximum degree of freedom is reached for the development of new operations.

Apart from executing operations on data, the objective to get information out of available data requires visualisation. Because of the big importance of visualisation a specialised view class, which actually might be interpreted as an operation class without data manipulation, is defined analogue to the data operation class.

4. System Structure and Flow Control

The classes of data objects and operation objects constitute the basis for the object oriented structure of the whole system. Modelling a process management system using these classes inevitably leads to structures that contain multitude instances of these classes. An overall object-oriented concept has to provide a suitable mechanism to connect data objects and operation objects. For that reason we introduce a class that maps the structure of a designed process management system using data objects and operation objects. Additionally, a mechanism is necessary that controls the execution of operation objects defined by the user. That means a class solving the task of a flow control for all operation objects inside an application system has to be designed.

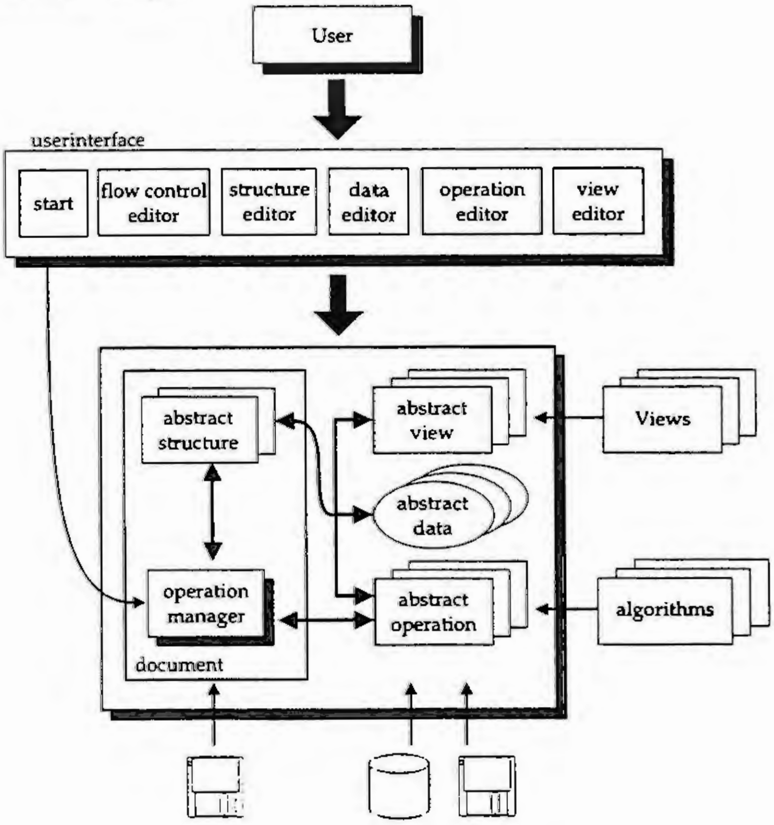


Figure 2. Architecture of the overall class concept

The class concept defines so-called link objects used to link an application's instances of the described classes (data object, operation object) to the desired structure. This class contains the complete functionality for assignment of data to operation input and output. Specifying the type all possible links are fixed. By public functions for definition and analysis of link structures it is possible to examine which data objects can be linked to which operation objects. An appropriate subclassing is necessary if execution only of specific operations should be enabled for new defined data objects.

The described classes provide the basic tool to link data objects with operations and views. The overall concept is completed by a class for flow control of operation execution in an application. This class is called operation manager and contains access to the public functions of link objects and operation objects. It enables definition of a general execution sequence of operation objects in an application. Execution of algorithms is started according to this flow control. Figure 2 shows the overall architecture for a process management system in context.

5. Example: Control System for a Gas Supply Network

By implementation of the described process management system in an easy to use operating interface, a software package was realised that supports the solving of practical problems. In co-operation with a big energy company, the process management system was used for development of a control system for a regional gas supply network [6][7]. The application provides a device for the so-called dispatching. It is used for technical and contractual optimisation in process control for this network [8]. In process management not only the specific structures of a certain energy supply network and the available resources have to be considered. Additionally, long term objectives have to be made for the operators resulting from the company's contractual regulations. In the past, these tasks that have to be executed in addition to the operative control and supervision functions were solved manually by different dispatchers. Thus, the decisions determining the transformation of long term objectives into operative actions were not comparable and not reproducible [9][10].

Using the object-oriented process management system, a model based strategy for gas supply control was developed. The application can be used for simulation of any time period in the past and for supporting actual decisions in process control. These control methods accomplishing the task of optimal purchasing, storing in or out, and gas supply had to be developed on the basis of the mathematical models for the network and the proposed characteristic of the reference variable. Transforming measured data into mathematical models allows the interpretation and handling during the development process later on. Because of the non-mathematical and partly non-technical background of the dispatchers, it was requested to provide a maximum of transparency within the system's models. Fuzzy systems have proved to be an effective way of modelling complex and non-linear technical systems [11] while allowing the interpretation and the understanding of the model's transfer behaviour which is mainly embodied in the rulebase. Thus, a neuro-fuzzy system provides direct linguistic interpretation of an approximating function given from measured data and, besides, it offers full access to further mathematical handling through its representation in form of relations [12].

The basis for the development of the mathematical model was the analysis of the company's database documenting all relevant technical data for a period of two years. The whole database was separated in two parts. First, the system model, which is based on a neuro-fuzzy structure, was developed and optimised in a learning process using data of one year. After that, model verification was done with the remaining data material. Finally, a fuzzy control system was designed for evaluation of gas supply and optimised performing simulations based on this mathematical model.

The process of development was supported by several advantages of the object-oriented management system. In the modified object-oriented concept, scalar values and series of statistical values are available as data types for creation of data objects. The user has any possibilities for definition and selection of several data sources. In this concrete application, the permanently stored energy supply amount is represented in data objects.

Data sources and data targets during learning differ from that during control. In this example the basis for control is a time period of only a few days up to the actual date, which for performance reasons is stored in a smaller database. Because of the separation of data from operations the link to these different data sources is made very easy. Using the new object-oriented approach handling of different data sources in a flexible system structure was made possible.

The neuro-fuzzy systems and the corresponding functionality are integrated as operation objects. Moreover extensive mathematical operations for application to data objects can be implemented as operation objects. Especially functions for data manipulation over longer time periods are important in this specific case. Mainly they serve for supervision of operation sequences up to certain moments in cycle time. E.g., evaluating energy supply amounts or computing prognosis values for any time period, etc. By the help of the corresponding link objects the user is able to define the desired system structure, i.e., linking the input and output interfaces of data objects to any number of operation objects or visualisation objects.

References

1. Coad, P., Yourdon, E., Object-Oriented Design. Englewood Cliffs, Prentice Hall, 1991.
2. Meyer, B., Objektorientierte Softwareentwicklung. Hanser 1990.
3. Achtert, W., Der Umstieg auf objektorientierte Programmierung. Elektronik 20/1994, 63-77.
4. Raasch, J., Systementwicklung mit strukturierten Methoden - ein Leitfaden für Studium und Praxis. Hanser Verlag, München, Wien, 1993
5. Microsoft ODBC 2.0, Programmer's Reference and Software Development Kit Guide. Microsoft Press, Washington, 1994.
6. v. Döllen, U., Konzepte für die übergeordnete Prozeßführung in der Energiewirtschaft. Dissertation, Schriftenreihe des Lehrstuhls für Regelungssysteme und Steuerungstechnik, Heft 46, Ruhr-Universität Bochum, 1996.
7. v. Döllen, U., Schlothane, M., Fuzzy Modeling of Gas Supply Networks. Proceedings of the 1995 EUROSIM Conference, EUROSIM '95, Vienna, Austria, 373-378.
8. v. Döllen, U., Homann, K., Einsatz der Fuzzy-Logik für das Gas-Dispatching. gwf - Gas/Erdgas 135 (1994) Nr. 5, S. 275-282.
9. Cerbe, G.(Ed.), Grundlagen der Gastechnik: Gasbeschaffung, Gasverteilung, Gasverwendung. Hanser Verlag, München, 1981.
10. Holschuhmacher, W., Wolf, H., Dispatching als technisch-gaswirtschaftliche Aufgabe, Herausforderungen an das Dispatching. gwf - Gas/Erdgas 132 (1991) Nr. 10/11, 470-476.
11. Preuß, H.-P., Fuzzy Control - heuristische Regelung mittels unscharfer Logik. atp - Automatisierungs-technische Praxis 34 (1992) 4, 176-184, 34 (1992) 5, 239-246.
12. Kosko, B., Neural Networks and Fuzzy Systems. Prentice Hall, 1992.

HIERARCHIC MODULAR STATEMODELS FOR DEVELOPMENT AND VERIFICATION OF CONTROL SYSTEMS

M. Rempe
University of Bochum, Dept. of Mechanical Engineering, Control Group
D-44780 Bochum, Germany

Abstract. The main idea of this contribution is the use of model-based methods in all phases of system-development. The statemodels consist of various discrete, continuous, and hybrid elements, organised in a hierarchic structure of system modules.

Introduction

Development of modern control systems should be based on effective strategies. Especially for such complex problems as flight control systems or any other controllers for aircrafts the requirements of security and robustness of the resulting systems are very high. Furthermore, the cost effectiveness should be as good as possible to be competitive on the global market.

This paper gives an overview of a solution to the described problem which considers especially the need of tests and validation methods for control systems. The conventional process of system development requires many types of documents for each step of design and verification. The mere test description and evaluation for a simple controller takes up a lot of expensive work if it is not supported by effective tools and methods. We believe that the most effective way of supporting the whole process, is the use of a formal description language, which enables, on the one hand, the definition of complex system behaviour with any functional details; it should be realised on a modern software tool with all necessary functions and interfaces on the other.

Strategy for System Development

The procedure to develop modern control systems based on modular statemodels is illustrated by Fig. 1. Note that one and the same system model is used for all phases of development and verification. The typical process is described by the steps of design and validation based on the „*V*" *method* [1]. In addition to the main process diagram, the use of the modular statemodels for each step is indicated. The design starts with the definition of the main modules such as controllers or system components. After having defined the logical and physical architecture of modules, the model will be completed with the exact definition of each state, transfer, etc. During the design phase, each step can be validated by off-line simulation and formal analysis of the model. For the complete model, a C-code generation can be done to realise a prototype of the control system.

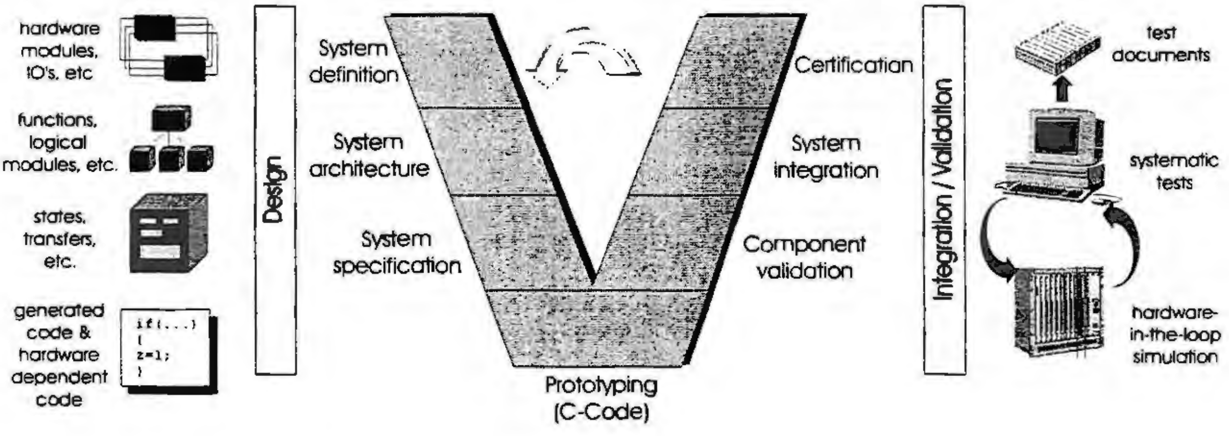


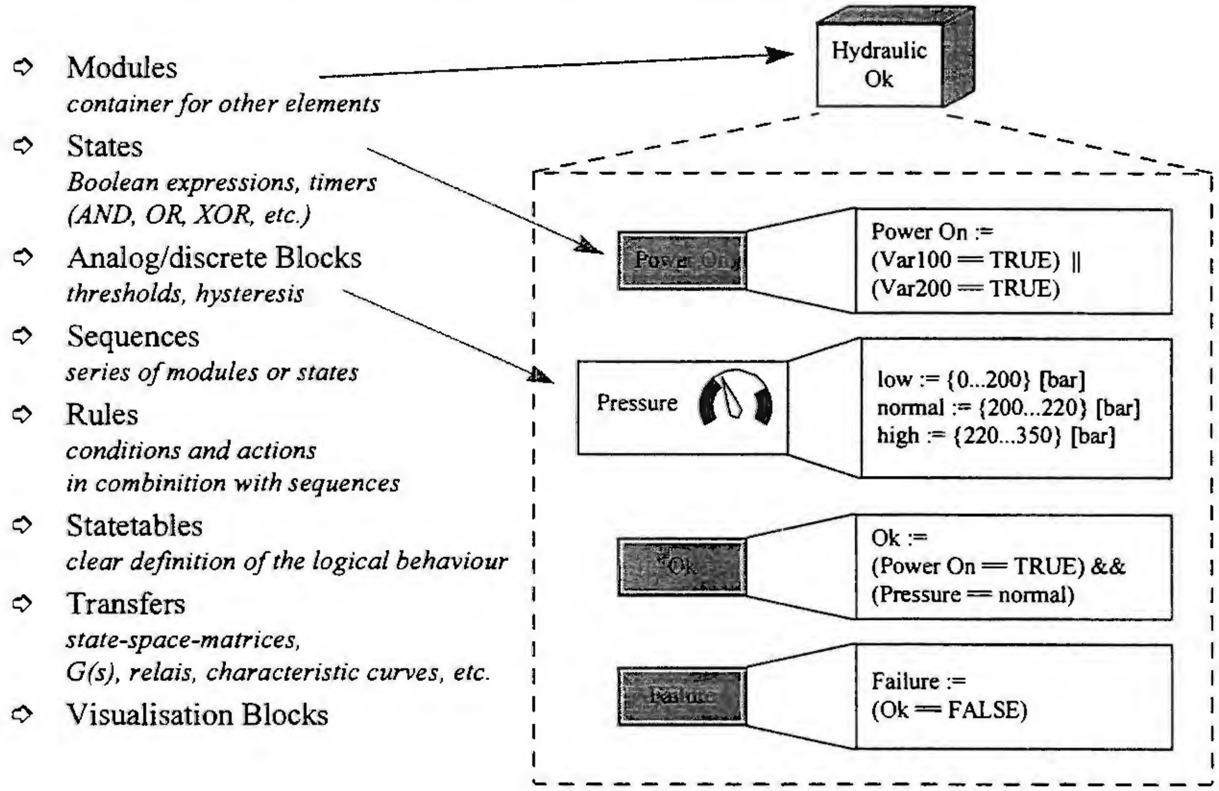
Fig. 1: The „V-Model" for system development and verification

With the prototype running on a real-time computer, hardware-in-the-loop-simulations are possible to validate the functionality of the whole system by integration of any hardware components. The results of these simulations can be evaluated by the same statemodel, which was used for prototyping. The details of these simulations are described in the later sections of this paper. At first, we will shortly present the architecture of the statemodels by mean of two short examples. Thus the modelling methods are illustrated.

Short Description of Hierarchic State Models

The statemodels consist of several elements as shown in Fig. 2. They represent a description language such as Petri-nets [2] or statecharts [3] which has been designed to realise a method of system modelling as mentioned above.

The basic element of the modelling language is the *state*, which contains a Boolean expression of system variables and other states. All defined states of a system can be "set" or "not set" and they may have a time controlled behaviour. The hierarchic structure of the models is enabled by the *modules*. The modules may contain all existing elements, including other modules, and like any other element they have a graphical symbol and are named. A module and the elements placed in this module belong together. For example, there could be a module called „lamp", containing the states „on" and „off'. This little model defines the terms lamp:on and lamp:off, which can be used for any other states or conditions, to describe the system behaviour. One of the methods to define conditions and events is the use of *rules* and *sequences*. A rule is a formal expression, which realises a conditioned action. In combination with a sequence of modules and states, a complex system behaviour can be described with only a few rules. There is also the possibility to define logical terms within the *statetables*.



- Modules
  container for other elements
- States
  Boolean expressions, timers (AND, OR, XOR, etc.)
- Analog/discrete Blocks
  thresholds, hysteresis
- Sequences
  series of modules or states
- Rules
  conditions and actions in combinition with sequences
- Statetables
  clear definition of the logical behaviour
- Transfers
  state-space-matrices, G(s), relais, characteristic curves, etc.
- Visualisation Blocks

Fig. 2: The elements of the modular statemodels

Fig. 2 shows a module called „Hydraulic", which can be in the states „Power On" „Ok" and „Failure". Further on, there is a continuos variable called „Pressure" with the states „low" „normal" and „high". The definition of the state „Ok" can be interpreted as: The hydraulic is „Ok", when „Power On" is set and the pressure is normal (between 200 bar and 220 bar).

For the continuous part of a system, *transfers* and *analog-discrete* blocks are implemented. For example a transfer could be the state space matrix, generated by MATLAB/SIMULINK; other descriptions such as linear transfer functions, polynomials, or relays, are also possible. Transforming an continuous value into a discrete state space is the task of the analog-discrete block, which contains several discrete states for a continuous variable, defined by names, thresholds and hysteresis. The block called „Pressure" in Fig. 2 with its states „low", „normal" and „high" depending on a continuous value is a typical application of analogue-discrete blocks.

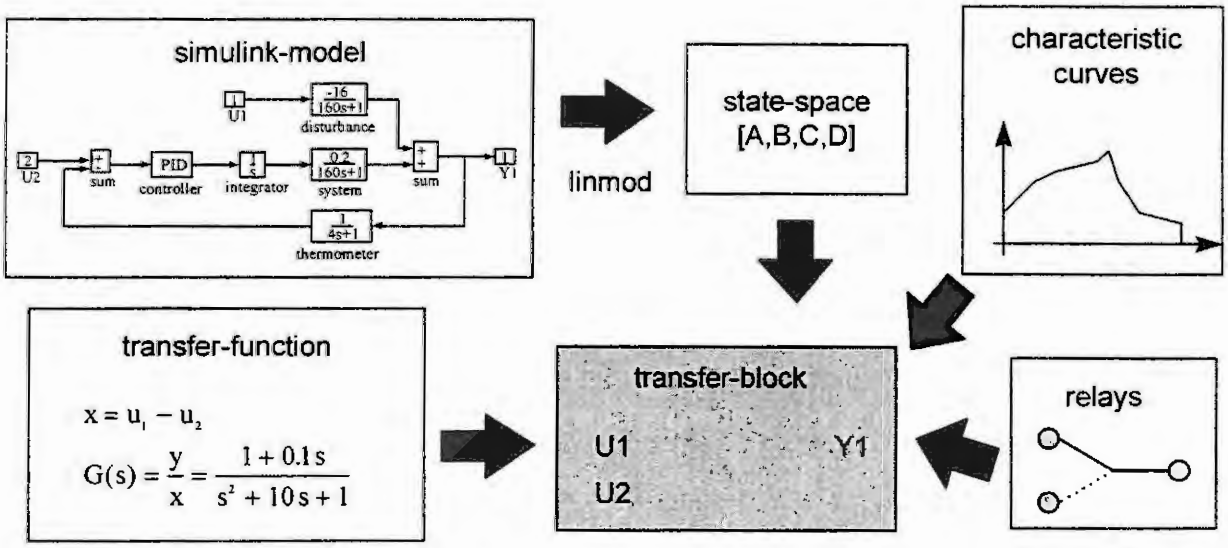
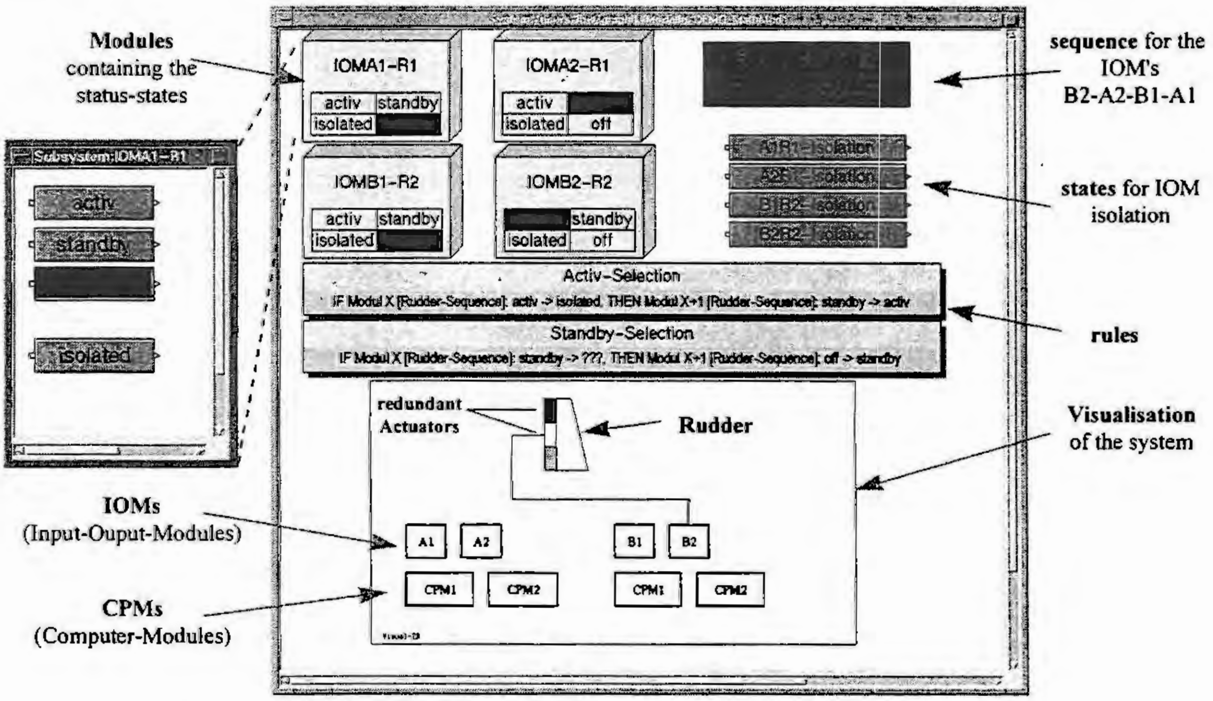


Fig. 3: Possibilities for use of transfers

An Example: Redundancy Management of a Flight Control System

One of the major features of fault-tolerant electronic flight control systems is the redundancy management according to which faulty modules are isolated and redundant fault-free modules are activated. The redundancy management is vital for safety and reliability of aircraft control. With respect to the importance of safety, the flight control system must survive several failures. The following example demonstrates the use of statemodels for a redundancy management in a flight control system. This example uses rules and sequences of modules for a clear definition of a complex behaviour.

Fig. 4 shows a redundancy management consisting of four redundant input-output modules (IOMs) for the rudder of an aircraft. The rules shall perform, that one IOM is active and another is in standby mode.



Activ-Selection

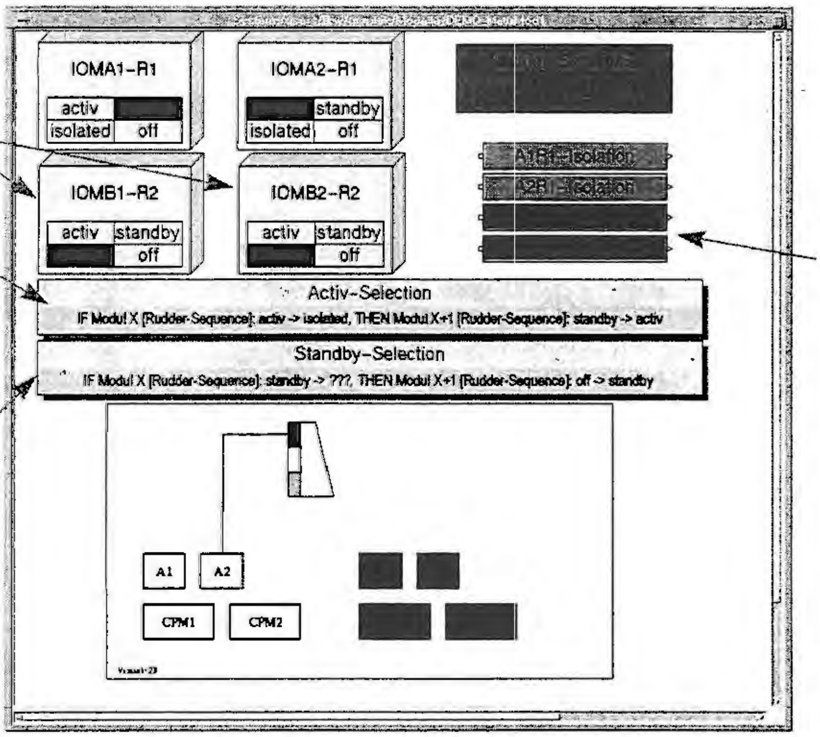
IF Modul X [Rudder-Sequence]: activ -> isolated, THEN Modul X+1 [Rudder-Sequence]: standby -> activ

Standby-Selection

IF Modul X [Rudder-Sequence]: standby -> ???, THEN Modul X+1 [Rudder-Sequence]: off -> standby

Fig. 4: A simple model for a redundancy management

Fig. 5 shows the system after a detected failure in the IOM B1 and B2. Regardless of the succession of these failures the two rules perform that the IOM A2 is active and IOM A1 is in standby mode.



Activ-Selection

IF Modul X [Rudder-Sequence]: activ -> isolated, THEN Modul X+1 [Rudder-Sequence]: standby -> activ

Standby-Selection

IF Modul X [Rudder-Sequence]: standby -> ???, THEN Modul X+1 [Rudder-Sequence]: off -> standby

IOMB1 and B2
isolated

Rule:
Activ-Selection
detectst hat the activ
IOM has been isolated
and activates the standby
IOM

Rule:
Standby-Selection
sets the standby
mode for the next
"off" IOM in
Rudder-Sequence

Simulated
failures
for IOMs
B1 and B2

Fig. 5: New configuration after two failures

The following event-list (Table 1) illustrates the simulation results. In the same way results of a hardware-in-the-loop-simulation could be presented.

Time [ms]	Event	Remarks
1690	Set state System:B1R2-Isolation TRUE	Simulated Failure in IOMB1
1700	Transition: System ??? -> B1R2-Isolation	
1700	Transition: IOMB1-R2 off -> isolated	
1830	Set state System:B2R2-Isolation TRUE	Simulated Failure in IOMB2
1840	Transition: System ??? -> B2R2-Isolation	
1840	Transition: IOMB2-R2 activ -> isolated	
1840	Condition in rule Activ-Selection	
1840	Set state IOMA2-R1:activ TRUE	Rule sets IOMA2 activ
1850	Transition: IOMA2-R1 standby -> activ	
1850	Condition in rule Standby-Selection	
1850	Set state IOMA1-R1:standby TRUE	Rule sets IOMA1 standby
1860	Transition: IOMA1-R1 off -> standby	

Table 1: Event-list for the simulated situation

Another Example: Level-Controller for a Water-system

The following example contains a very simple version of a Level-Controller. A realistic version of such controller (for example as it is used for the Water-system in an Airbus A340) contains many more elements, has many sub-modules and functions, and cannot be discussed in this paper. This example shall only demonstrate the use of the statemodels for such problems.

The Controller has connections to two valves, one for the input and another for drain. The statetable „LevelControl" ensures that whenever the level of the tank is not normal (between 1.0 and 3.0), the corresponding valve opens. The connection between the controller and the valves is defined by system-variables. It is possible to replace the simulated valves by hardware to execute a hardware-in-the-loop-simulation. This can be done without changing the definition of the controller.

The operation of the valves is realised by a simple statetable, which is disabled, if the valve is not „ok". This is a simple description of a solenoid-operated mechanism.

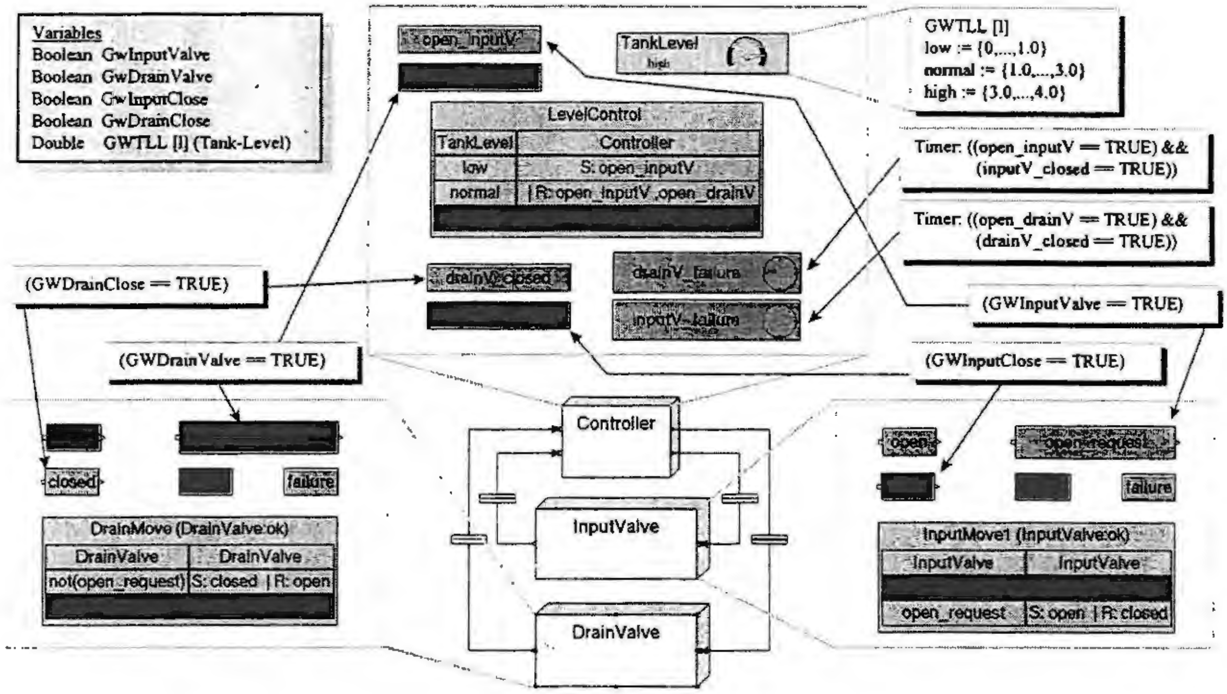
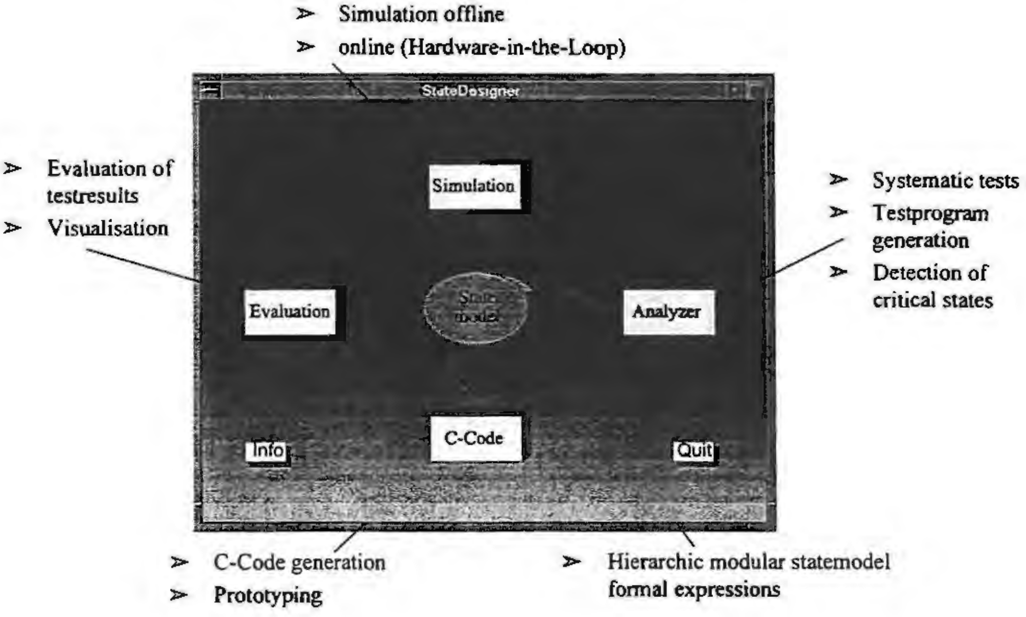


Fig. 6: Level-Controller with two valves

The time-controlled failure detection in the controller is done by comparison of actual and setpoint values of the valves. If there is a difference for more than one second, the corresponding failure-state will be set.

Graphical User-Interface

The basic function of the tool „State Designer" is the simulation of the models. In early design phases the models may be not detailed, but simulation is also possible. In later phases, when a complete model of the system behaviour is defined and has been analysed, prototyping can be done by using the C-code generator of the tool.



- Simulation offline
- online (Hardware-in-the-Loop)

- Evaluation of testresults
- Visualisation

- Systematic tests
- Testprogram generation
- Detection of critical states

- C-Code generation
- Prototyping

- Hierarchic modular statemodel formal expressions

Fig. 7: The graphical user interface and its functions

The next step of system development consists of the validation and integration of the system by implementing the programmed controller and some hardware components on a test-bench such as the flight-control test-bench at DASA, Hamburg. There, the models can be simulated with hardware-in-the-loop (HWILS) and, furthermore, it is possible to execute systematic tests, simulate them in real-time and analyse the produced data with the result of complete documents containing plots, event-lists and other informations. All these functions are based on the same statemodel, however it is not necessary to have a detailed model for these function. Only if prototyping should be done, the „last bit" of the conditions and the states have to be defined.

Hardware-in-the-loop-Simulation

One of the most remarkable attribute of the statemodels is the possibility of hardware-in-the-loop-simulations without any modifications on the models. This function is enabled by a common list of variables, which is used for the definition of the states, transfers, etc.

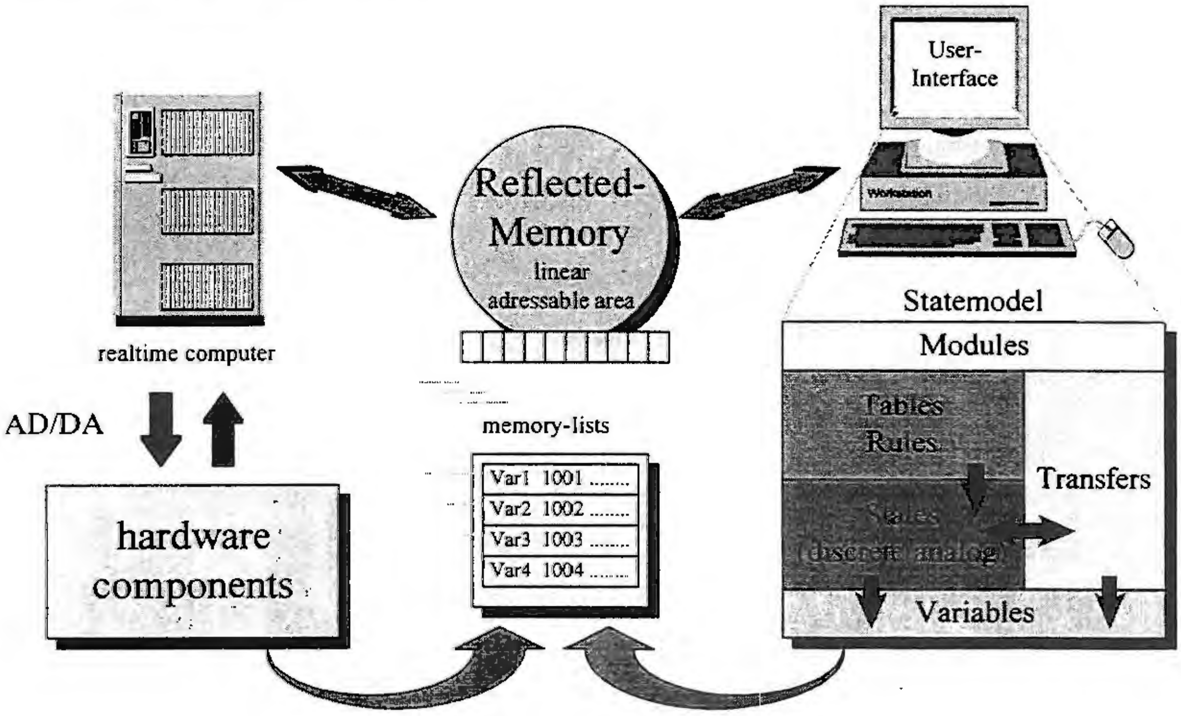


Fig. 8: Hardware-in-the-loop-simulations

Summary

We presented a strategy for system development based on hierarchic modular statemodels. With this solution a integrated sequence of work for the whole process becomes possible. The main advantage of this strategy lies in the reduction of required interfaces between all steps of design and verification because a single model can be used for all tasks from the first definition up to the certification documents.

The next step of the presented project is the realisation of a technology demonstrator at DASA Airbus, Hamburg. With this application the practical use of the tool State Designer will be improved and several functions for the data management will be added with the result of an complete system development environment.

References

1. Bröhl, A.-P. und W. Dröschel (Hrsg.). Das V-Modell. Der Standard für die Softwareentwicklung. 2. Aufl. R. Oldenburg Verlag, 1995.
2. Reisig, W., Systementwurf mit Netzen. Springer-Verlag 1985
3. Harel, D., Statecharts: A visual Formalism for Complex Systems, Science of Computer Programming, Vol. 8, 1987
4. Yourdon, E.: Moderne strukturierte Analyse, Wolframs Fachverlag 1992

NONLINEAR DYNAMIC MODELING OF A GAS ENGINE USING AN RBF-NETWORK

R. Korb[1], H.P. Jörgl[1] and B. Lutz[2]
[1]Technical University Vienna, E328
Gußhausstraße 27-29, A-1040 Vienna
email: korb@impa.tuwien.ac.at
[2]Jenbacher Energiesysteme AG
Achenseestraße 1-3, A-6200 Jenbach

Abstract. A Radial Basis Function network (RBFN) is used to obtain a model of a gas engine, an unstable two-input/single-output system (MISO-system), to be used for the design of the speed control system. The RBFN-centers are chosen using the stepwise orthogonalization algorithm, and an input space compression which helps to avoid sparse data sets is presented. The quality of the nonlinear RBFN-model is demonstrated by comparing measured and simulated data.

1. Introduction

Various methods for modeling nonlinear dynamic systems have been developed in the recent years [1,2]. RBF networks [3] turned out to be a highly successful tool for approximating nonlinear functions. In this paper the well developed theory of RBF network training is utilized for nonlinear dynamic modeling of a gas engine.

The paper is organized as follows: In section 2 the gas engine plant is described briefly and in section 3 a short introduction to radial basis functions is given. In section 4 the NARMA model structure used in this work is presented and in section 5 the nomenclature for an RBF network of the gas engine plant is defined. The training of the network is discussed in section 6 and the input space compression is introduced in section 7. Simulation results are shown in section 8 and finally, some conclusions are drawn.

2. Gas engine plant

The combustion engines investigated in this work, are used for the co-generation of heat and electrical power. They are distinguished by a high utilization of primary energy of almost 90 %. Furthermore it is possible to burn gases, which cannot be utilized by any other process.

Gas engine plants can be operated within a large electric net or in a stand alone mode with an electric resistance as load. Here, only the stand alone mode of operation mode which is the more challenging one, is investigated, because in this case the frequency must be kept constant by a speed controller.

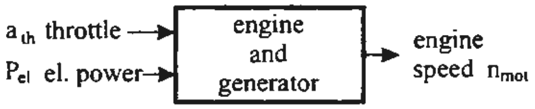


Figure 1: Black box model of the plant

As the speed control loop is to be modeled, the throttle position is the control variable, the engine speed is the controlled variable (output variable) and the electrical power acts as the disturbance variable. The gas mixer is controlled by a separate controller, thus guarantying low NO_X emissions in the exhaust gas. Figure 1 shows the two inputs and the output of this MISO-system.

Everybody is familiar with the stable behavior of combustion engines in cars. They are stable, because the load increases strongly as the engine speed increases. In the case of the investigated combustion engines the situation is completely different: The load is independent of the engine speed. Therefore the plant is unstable and measurements are not possible in open loop.

3. Radial basis functions

A Radial Basis Function (RBF) is a function that maps the n-dimensional input domain $I\!R_l^n$ onto the real axis. All input vectors x which have the same distance to the center $\mathbf{c} \in I\!R_l^n$ give the same output as

$$\phi(\mathbf{x},\mathbf{c}) = \phi(\|\mathbf{x} - \mathbf{c}\|) = \phi(r), \qquad (1)$$

In equation (1), $\|\cdot\|$ denotes the Euclidean norm.

A number of different basis functions are suggested in literature. In this paper the reciprocal multiquadric (RMQ) is used:

$$\phi(r) = \frac{1}{\sqrt{r^2 + \beta}}, \quad \beta > 0 . \tag{2}$$

In [3] Pottmann showed for the RMQ that β can be varied in a wide range. Additionally, in this paper the input domain is mapped on a hypercube by simple scaling and shifting, so that $\mathbf{x} \in [-1,1]^n$ holds. This drastically simplifies the choice of β.

4. The NARMA model

The nonlinear autoregressive moving average model (NARMA-model) can be derived from the simpler linear case, namely the ARMA process. The ARMA process for a SISO-system is mathematically described by the difference equation

$$y[kT] + a_1\, y[(k-1)T] + \ldots + a_n\, y[(k-n)T] = b_0\, u[(k-d)T] + b_1\, u[(k-1-d)T] + \ldots + b_n\, u[(k-n-d)T], \tag{3}$$

where n is the system order, d is the discrete time delay and T is the sampling time. Equation (3) describes a $(2n + 1)$ - dimensional hyperplane in the $(2n + 2)$ dimensional hyperspace.

If this description of a plane is extended to that of a surface, a wide range of nonlinear dynamic systems can be described. The general NARMA model for a SISO-system is given by the nonlinear difference equation

$$y[kT] = \mathrm{F}(y[(k-1)T], \ldots, y[(k-n)T], u[(k-d)T], u[(k-1-d)T], \ldots, u[(k-n-d)T]) . \tag{4}$$

Here, $\mathrm{F}(\cdot)$ is an arbitrary, nonlinear function. For sake of simplicity all inputs of the function F in equation (4) are collected in a column vector $\mathbf{x}[kT]$. Then, equation (4) can be written as:

$$y[kT] = \mathrm{F}(\mathbf{x}[kT]) . \tag{5}$$

5. RBF networks for nonlinear two input one output systems

One possibility to describe the hypersurface given by equation (4) is the weighted summation of radial basis functions with different centers $\mathbf{c}$ and appropriate weighting factors w_i. It has been shown previously [1] that RBF-networks are universal approximators. This means that any function F in equation (5) will be described by a RBF-network arbitrarily precisely, if a sufficient number of RBFs are used. The equation of a RBF-network is given by

$$y[kT] = \sum_{i=1}^{m} w_i\, \phi(\|\mathbf{x}[kT] - \mathbf{c}_i\|) + e[kT] . \tag{6}$$

Since only a finite number of RBFs are used ($m < \infty$), $y[kT]$ is not exactly described by this sum and an error $e[kT]$ remains.

In order to be able to describe linear systems exactly without using an infinite number of RBFs, additional linear elements are included, i.e. equation (6) is extended to

$$y[kT] = b + \mathbf{x}^T[kT]\theta_L + \sum_{i=1}^{m} w_i\, \phi(\|\mathbf{x}[kT] - \mathbf{c}_i\|) + e[kT] \tag{7}$$

Using the abbreviations

$$\phi_i(\mathbf{x}[kT]) = \phi(\|\mathbf{x}[kT] - \mathbf{c}_i\|) , \tag{8}$$

$$\phi(\mathbf{x}[kT]) = [\phi_1(\mathbf{x}[kT]), \ldots, \phi_m(\mathbf{x}[kT])]^T \tag{9}$$

and

$$\theta^T = [b, \theta_L{}^T, w_1, \ldots, w_m] , \tag{10}$$

equation (7) can be written compactly as

$$y[kT] = [1, \mathbf{x}^T[kT], \phi^T(\mathbf{x}[kT])]\theta . \tag{11}$$

Since an RBF-network for the description of a nonlinear dynamic two-input/single-output system is to be developed and since the system to be modeled does not exhibit jump phenomena, the vector **x** in equation (11) has to be extended and becomes:

$$\mathbf{x}\left[kT\right]=[y[(k-1)T]\cdots y[(k-n)T],u_1[(k-d-1)T]\cdots u_1[((k-d-n)T],u_2[(k-d-1)T]\cdots u_2[(k-d-n)T]]^T \quad (12)$$

A graphical representation of equation (11) is given in figure 2.

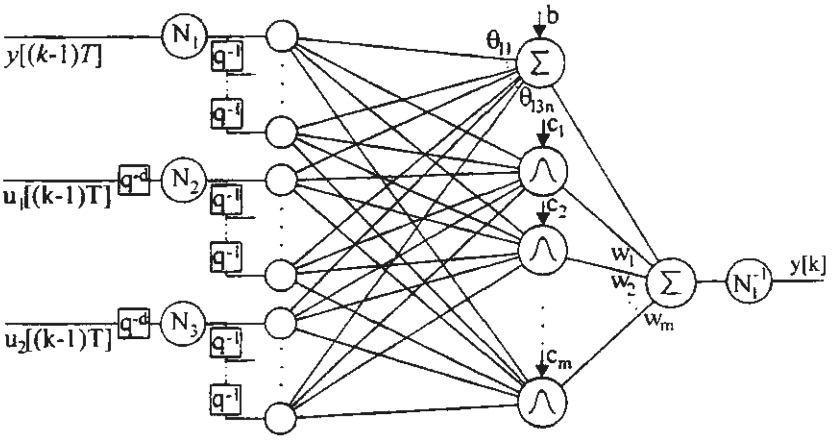


Figure 2: RBF-network with scaling, time shift layer, linear and RBF-part and inverse scaling

6. RBF-network training

Training of an RBF-network means adaptation of the parameters m, β, $\mathbf{c}_i$ and θ in order to minimize $e[k\,T]$ for all k. The number m of the hidden nodes in the network is often restricted by computational limits. The simultaneous estimation of β, $\mathbf{c}_i$ and θ yields a non-convex optimization problem with local minima. In order to circumvent this problem, β is estimated, before $\mathbf{c}_i$ and θ are computed simultaneously by an orthogonalization procedure [3,5]. First of all a matrix $\mathbf{X}_S$ is built, which includes the evaluation of the bias term, the linear terms and all RBF candidates at all data points of the training data set.

$$\mathbf{X}_S=\begin{bmatrix} 1 & \mathbf{x}^T[0] & \phi_1(\mathbf{x}[0]) & \cdots & \phi_N(\mathbf{x}[0]) \\ 1 & \mathbf{x}^T[T] & \phi_1(\mathbf{x}[T]) & \cdots & \phi_N(\mathbf{x}[T]) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \mathbf{x}^T[N\,T] & \phi_1(\mathbf{x}[N\,T]) & \cdots & \phi_N(\mathbf{x}[N\,T]) \end{bmatrix} \quad (13)$$

Additionally,

$$\mathbf{y}=[y[0]\quad y[T]\quad\cdots\quad y[N\,T]]^T \quad (14)$$

and

$$\mathbf{e}=[e[0]\quad e[T]\quad\cdots\quad e[N\,T]]^T \quad (15)$$

are defined.

The the m linear and nonlinear terms which best reduce the variance of the residual error are chosen from the columns of $\mathbf{X}_S$ using the Gram-Schmidt orthogonalization. The columns $\mathbf{x}_k$ of $\mathbf{X}_S$ minimizing $\varepsilon^T\varepsilon$ are chosen step by step. Thus the this algorithm must be followed:

$$I_1=\{1,\ldots,1+3n+N\} \quad (16)$$

Calculate in every step j ($j\in[1,m]$):

Exclude the columns, which lead to wrong results because of the finite precision of the floating point algebra:

$$s_j=\underset{k\in I_j}{\arg}\,(\mathbf{x}_k{}^T\mathbf{x}_k<10^{-12}) \quad (17)$$

$$I_j:=I_j\setminus\{s_j\} \quad (18)$$

Find the index of the column reducing $\varepsilon^T\varepsilon$ most:

$$l_j = \arg\max_{k \in I_j}\left(\frac{(\mathbf{x}_k{}^T\mathbf{y})^2}{\mathbf{x}_k{}^T\mathbf{x}_k}\right) \qquad (19)$$

Calculate in every but the last step:

Exclude l_j from the set of candidates

$$I_{j+1} = I_j \setminus \{l_j\} \qquad (20)$$

Orthogonalize all candidates regarding $\mathbf{x}_{l_j}$

$$\mathbf{X}_S = \left(\mathbf{I} - \frac{\mathbf{x}_{l_j}\mathbf{x}_{l_j}{}^T}{\mathbf{x}_{l_j}{}^T\mathbf{x}_{l_j}}\right)\mathbf{X}_S \qquad (21)$$

In this algorithm it is possible to include the computation of the weights, but it is not advisable, because there are numerically better conditioned methods for the weight estimation after finishing the RBF-selection.

After the choice of the m best indices, a matrix $\mathbf{X}$ similar to that in equation (13) is built. It contains m columns of $\mathbf{X}_S$.

$$\mathbf{X} = [\mathbf{x}_{l_1} \quad \mathbf{x}_{l_2} \quad \cdots \quad \mathbf{x}_{l_m}] \qquad (22)$$

Finally, the weighting vector (equation 10) is calculated:

$$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\,. \qquad (23)$$

7. Input space compression

Unfortunately, the high input domain dimension of an RBF-network leads to bad training results as the training data set is sparse in this case. Sparse data sets must be avoided, since due to wrong extrapolation, unlearned regions may lead to wrong RBF-network outputs.

A possibility to avoid unnecessary high dimension is the input space compression. One method is suggested in [4], here a different method is presented:

The linear part of the matrix $\mathbf{X}_S$ (eqn. 13) is collected in the matrix $\mathbf{L}$. Each row contains the input signals of the RBF-network at a certain instant $k\,T$.

$$\mathbf{L} = [\mathbf{x}[0] \quad \mathbf{x}[T] \quad \cdots \quad \mathbf{x}[N\,T]]^T \qquad (24)$$

Orthogonalization of $\mathbf{L}$ yields this algorithm with column pivoting:
Start with

$$\mathbf{M} = \mathbf{L}\,. \qquad (25)$$

Calculate in every step j ($j \in [1,3n]$):

Select the column with the greatest 2-norm

$$N_j = \max_{k \in I_j}(\mathbf{m}_k{}^T\mathbf{m}_k) \qquad (26)$$

$$l_j = \arg\max_{k \in I_j}(\mathbf{m}_k{}^T\mathbf{m}_k) \qquad (27)$$

Notice that $\mathbf{m}_k$ is the k-th column of $\mathbf{M}$. Orthogonalize $\mathbf{M}$ with respect to $\mathbf{m}_{l_j}$ in every but the last step

$$\mathbf{M} := \left(\mathbf{I} - \frac{\mathbf{m}_{l_j}\mathbf{m}_{l_j}{}^T}{\mathbf{m}_{l_j}{}^T\mathbf{m}_{l_j}}\right)\mathbf{M} \qquad (28)$$

There are two possible ways to reduce the input space dimension. One is to use only the first selected input vectors $\mathbf{l}_{l_1},\ldots,\mathbf{l}_{l_p}$ of $\mathbf{L}$. The other possibility is the calculation of N_j/N_1 in every step and finishing the loop (equation 26 to 28), if this division yields a value less than a chosen limit.

The input space compression matrix is given by the equation

$$\mathbf{T} = \left[\mathbf{I}_{l_1} \quad \mathbf{I}_{l_2} \quad \cdots \quad \mathbf{I}_{l_p}\right]^T, \qquad (29)$$

where $\mathbf{I}_{l_i}$ is the i-th column of the ($n \times n$) identity matrix $\mathbf{I}$.

Consequently, it is possible to obtain an input vector of reduced dimension without loosing essential information even in the presence of noise, as this input space transformation removes (nearly) redundant input signal components only.

$$\mathbf{x}_R[kT] = \mathbf{T}\mathbf{x}[kT] \tag{30}$$

Notice that this method permits automatic order estimation, since it is possible to choose a large system order and then use the automatic reduction by equation (30).

8. Simulation results

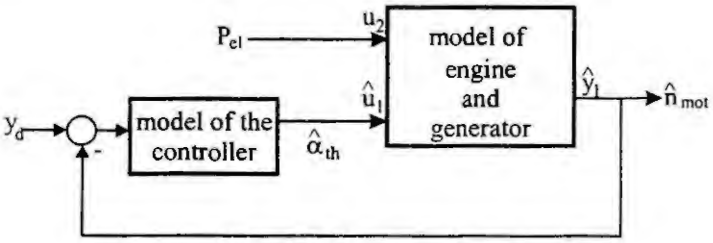


Figure 3: simulation in the closed loop

Data acquisition and validation of the model is possible in closed loop only, as the plant is unstable. The plant is excited by a variable electrical load and a noisy oscillating desired speed value. The measured signals $y_d[kT]$ (desired engine speed) and $u_2[kT] = P_{el}[kT]$ (electrical power output) are used as inputs in the simulation and the simulated signals $\hat{u}_1[kT] = \hat{\alpha}_{th}[kT]$ (throttle position) and $\hat{y}[kT] = \hat{n}_{mot}[kT]$ (engine speed) are compared to the measured signals $u_1[kT] = \alpha_{th}[kT]$ and $y[kT] = n_{mot}[kT]$. The order of the plant is assumed to be 8. The inputs and outputs of the three following models are compared with measured data sets:

- The linear model,
- the nonlinear RBF-network model (16 linear terms and 34 RBFs) and
- the RBF-network model with input order compression (reduction of input dimension from 24 to 13, 1 bias term, 13 linear terms and 36 RBFs).

In figure 4 a portion of the training data (N = 1620) is shown.

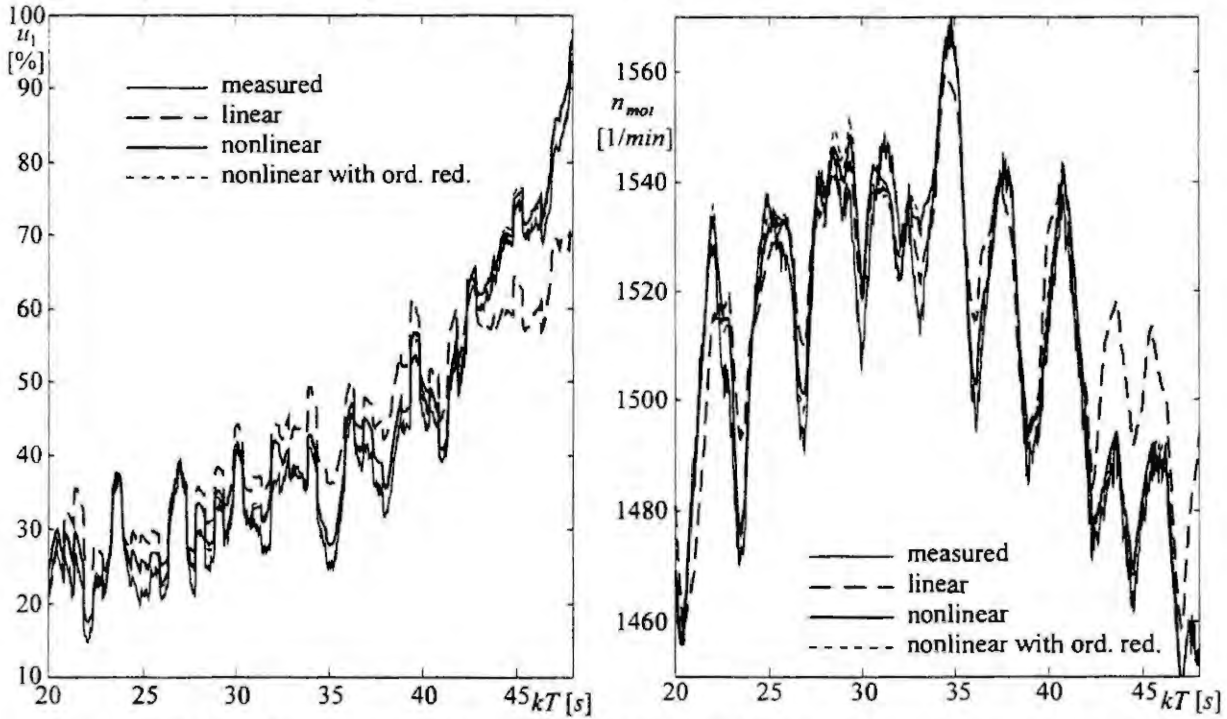


Figure 4: Comparison of the measured and simulated throttle positions (left) and engine speeds (right) for three different models using the training data set

In figure 5 a portion of the validation data set is depicted.

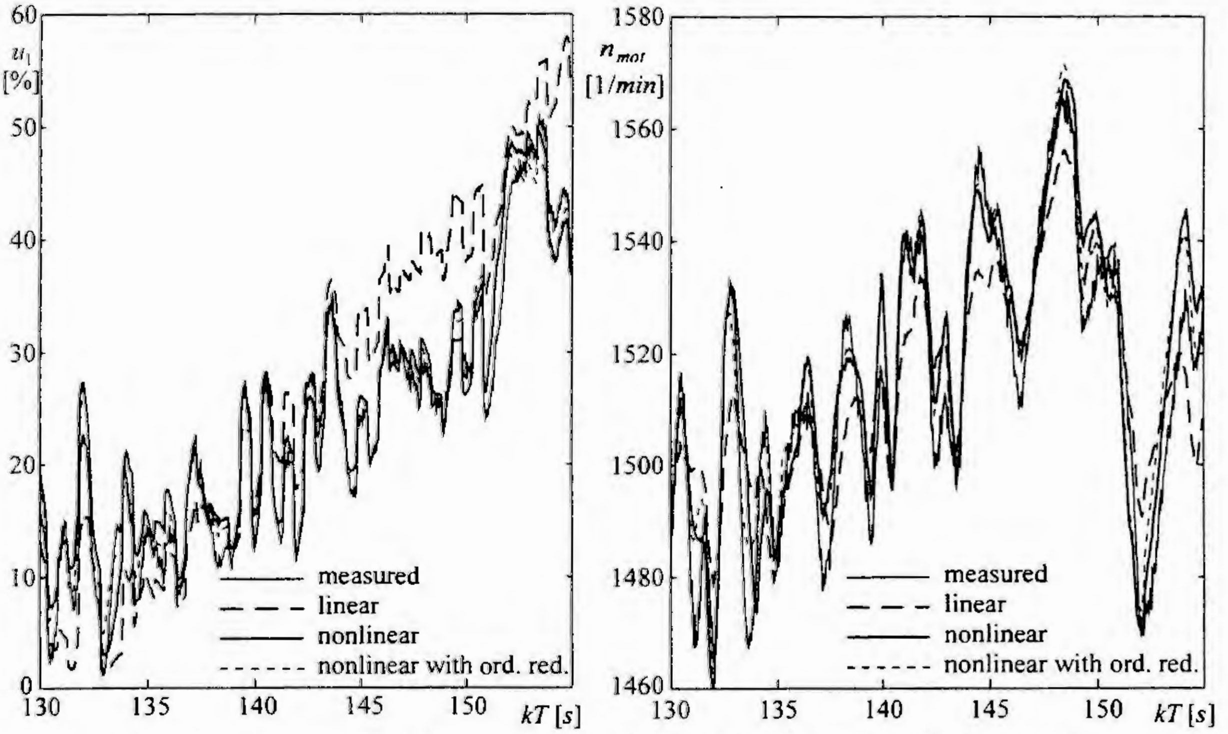


Figure 5: Comparison of the measured and simulated throttle positions (left) and engine speeds (right) for three different models using the validation data set

From figures 4 and 5 it can be observed that the nonlinear models yields drastically improved simulation results compared to those with the linear model. Unfortunately, it is not possible to show data sets where the difference between the linear and the nonlinear model is more obvious, as it is not possible to drive the engine in these regions for a longer period of time.

9. Conclusion

In this paper the investigations of the applicability of RBF networks in modeling an industrial combustion engine plant are presented. It is shown that the nonlinear model obviously improves the simulation results obtained using the linear model. The input space compression which is introduced in this paper reduces the sparsity of the data sets and the computational burden without deteriorating the simulation results.

References:

1. Hunt, K.J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P.J., (1992): Neural Networks for Control Systems- a Survey. Automatica 28, p1083-1112.
2. Judinsky, A., Zhang, Q., Delon, B., Glorennec, P-Y. and Beneviste, A., (1994): Wavelets in Identification. Technical Report 849, IRISA/INRIA, 106p.
3. Pottmann, M. (1993) Radial Basis Function Networks for Process Identification and Control. Dissertation. Technical University of Vienna
4. Scott, D.W., (1992): Multivariate Density Estimation. Wiley, New York, 317p
5. Zhang, Q. (1994) Using Wavelet Network in Nonparametric Estimation. Internal Report , IRISA/INRIA

MODULE–ORIENTED MODELING OF AIRCRAFT STRUCTURAL BEHAVIOUR

G. Beuck and L. Merz,
DASA–Airbus GmbH, Kreetslag 10, D – 21129 Hamburg,
Email: L.Merz@airbus.de

Abstract

One integrated hard– and software environment is described usable within all project stages and for several people and different disciplines involved in the design. A module–oriented development approach eases teamwork, lowers design costs and increases quality. The treatment of discontinuities and events is a matter of continuing development work.

Introduction

Modeling, simulation and optimization of aircraft behaviour depends on (see Fig. 1)

- the available data basis (early project stage means few, rough data)
- the objecives: technical quality, time and costs
- number of different engineering domains involved
- number of different aircraft components, systems and environments interacting
- Existing software–, hardware– and people–ware–limitations

All these points are more or less interrelated. The ultimate aim is to use one development tool through all complexity levels of offline–simulation and –design up to hardware–in–the–loop simulation and flight tests (Fig. 2). During the last five years DASA–Airbus GmbH, Hamburg, and the Ruhr–Universität Bochum (Lehrstuhl für Regelungssysteme und Steuerungstechnik, Prof. Dr. Fasol) have been going together quite a few steps in this direction [1].

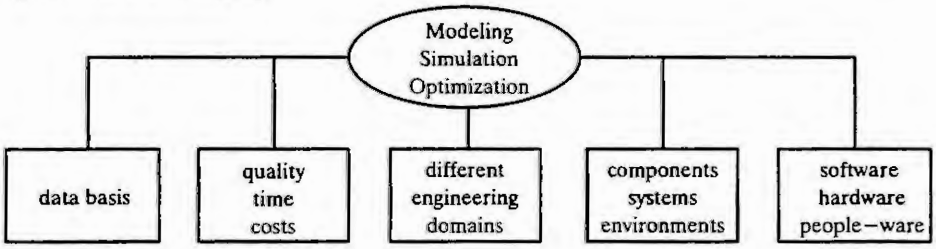


Fig. 1 Functionality structure

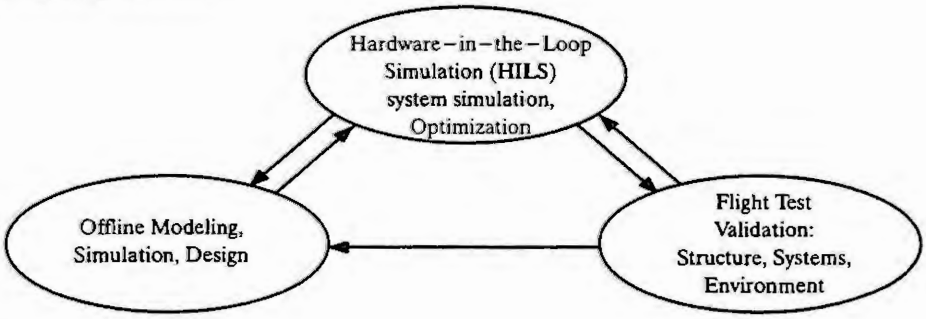


Fig. 2 Project stage

Loads and Energy of Aircraft Structure

Civil transport aircraft are built to carry passengers and/or cargo safe, economical and comfortable from one place to the other. The aircraft structure besides other tasks houses people and materials and protects against environmental dangers. This paper concentrates on designing the aircraft structure to withstand all kind of forces or loads. We distinguish between steady loads, when no inertia loads are present and dynamic loads, when loads and system energy are time–dependant.

The mechanical dynamic process starts with the introduction of loads and energy into the aircraft structure. These can be environmental inputs like atmospheric turbulences and landing impacts or man–generated inputs like engine forces and system–driven forces leading to control surface deflections (Fig. 3).

The dynamic response of the aircraft structure gives rise to aerodynamic reaction forces, inertia forces and internal structure loads.

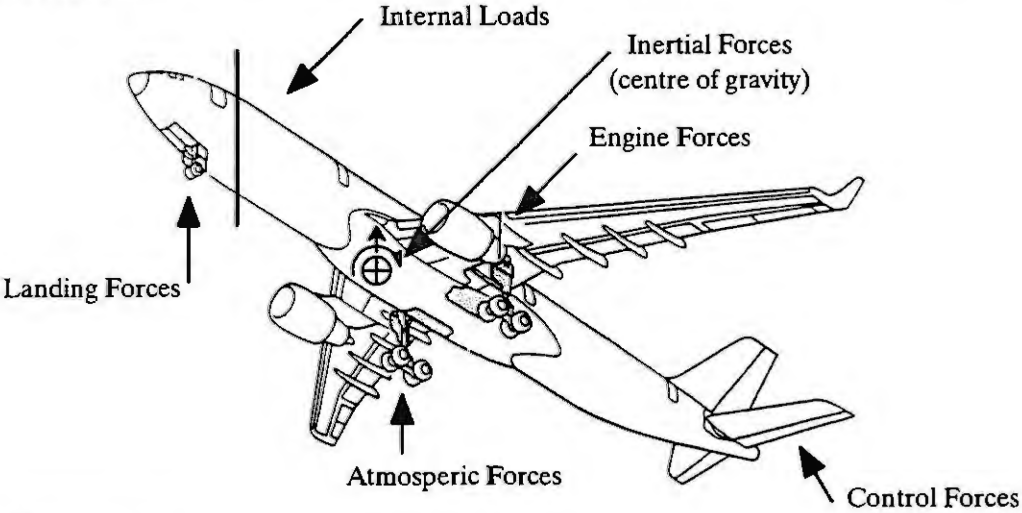


Fig. 3 Dynamic loads

Modules for Structural Dynamics Modeling

To reach the just defined objectives in an economical manner we decided to put into the simulation environment as much modularity, flexibility and transparency as possible.

The following modules have been chosen (Fig. 4):

Structure:
fuselage, wings, horizontal tailplane, vertical stabilizer, engine pylons, non–rotating part of engines

control surfaces:
ailerons, spoilers, flaps, slats, elevators, rudder

Landing mechanisms:
nose landing gear, main landing gear, tail bumper

Engines:
rotating part of jet engines, auxiliary power unit (APU)

Mechanical–hydraulic part of actuators:
cylinder, piston, servovalve, actuator support structure, control surface linkage, supply and return pipes, hydraulic flow characteristics, ...

Electric part of actuators:
position sensor, position gain, monitoring device, ...

Electrical flight control systems:
sidestick control (pilot actions), feedback sensors, gain, time constants, threshholds, authority limits, protections, altermate laws, monitoring systems, ...

Aerodynamic forces:
turbulence excitation, control surface excitation, rigid body and elastic movement, unsteady effects

Interiors:
seats, crew, passengers, cargo, galleys, ...

Further moduls:
gravity forces, thrust characteristics, runway characteristics, transformation between reference systems (forces, movements)

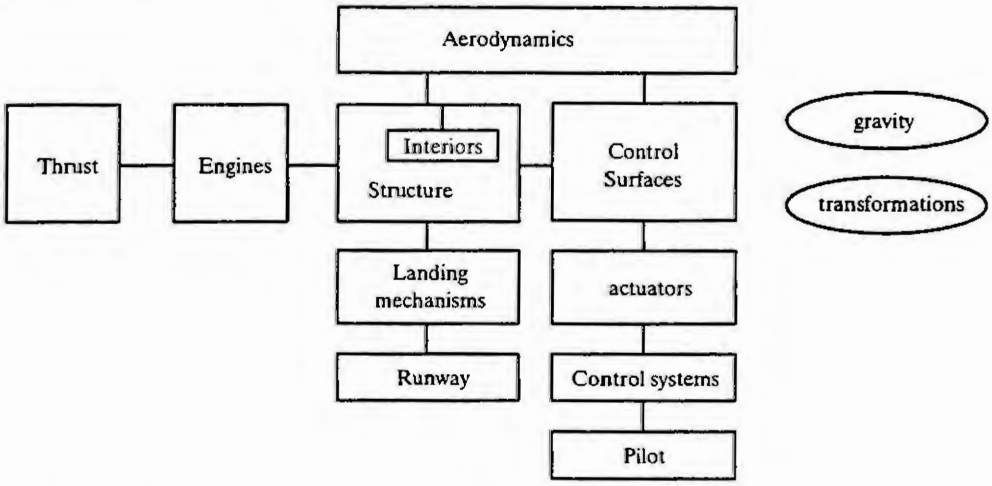


Fig. 4 Modules for structural dynamics modeling

Coupling of the Structural Modules

The first step is to develop the equations of motion for the modules structure (including interiors), control surfaces, engine and landing mechanisms without coupling and only considering inertia and elastic forces. An elastic body with continuously distributed mass has got an infinite number of degrees of freedom (DOF). By finite element modeling (NASTRAN) the number of degrees of freedom for the total aircraft structure come down to about 8000. By modal reduction we arrive at about 100 time dependant variables (Fig. 5).

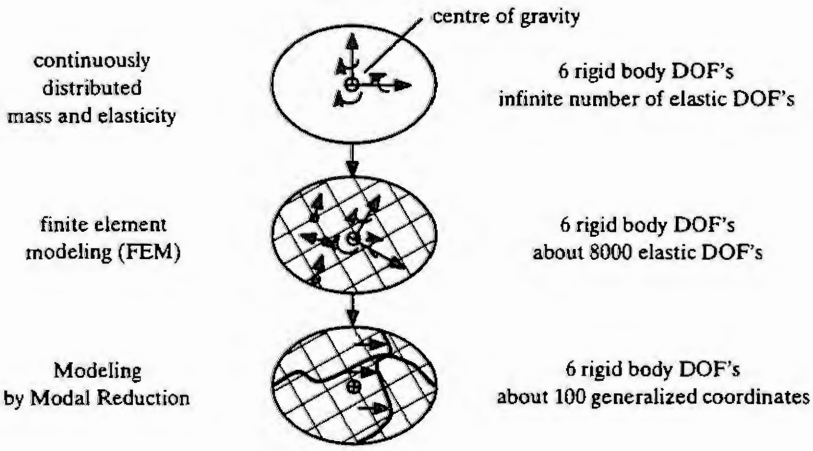


Fig. 5 Structural modeling

The second step is to couple the structural modules.

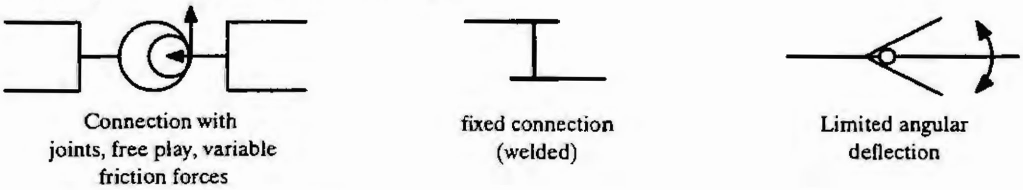


Fig. 6 Module coupling and variable structure systems

A connection may lead to a reduction form zero up to six DOF's (Fig. 6). Zero DOF's means no interaction at all and six DOF's means a fixed connection. If during the simulation the number of DOF's vary, we have got a variable structure system. Especially the existence of free play gives rise to it. In general we have got to describe discontinuities in otherwise continuous models. An efficient way to handle such systems is the use of so–called state–machines. Otherwise you easily loose control by the huge amount of if–then statements. Of special interest is a good numerical behavior of the simulation across the discontinuities [6], [7].

Coupling of the Aerodynamic Modules

Aerodynamic forces arise, if there is a relative movement between a body and the surrounding air. Once you know the atmospheric turbulence and the rigid body and the elastic movement of the surface of the aircraft, the aerodynamic pressures and friction forces can be derived (Fig. 7).

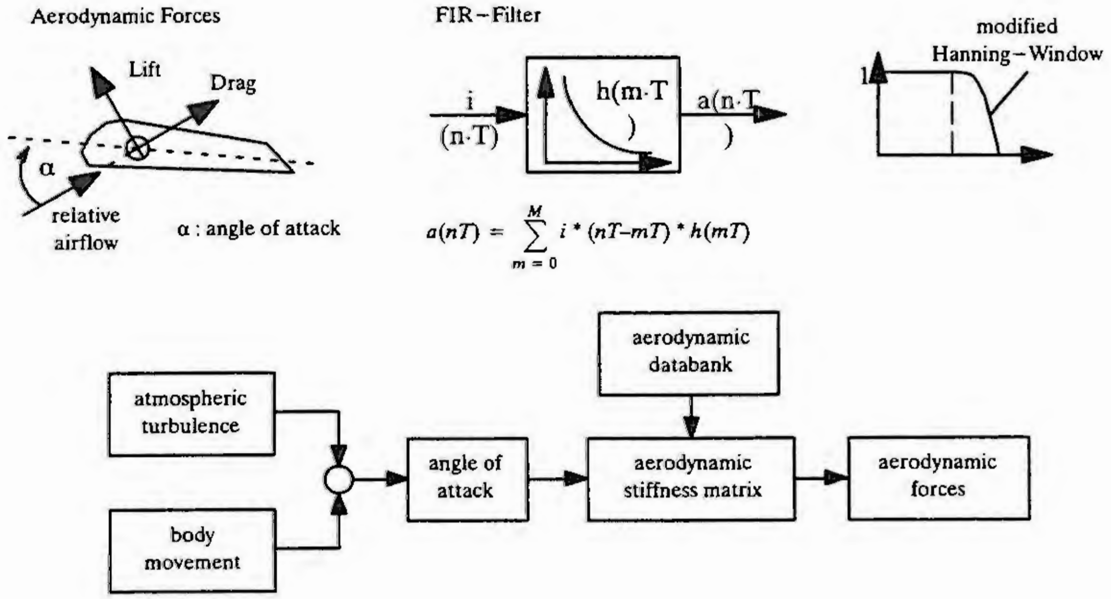


Fig. 7 Aerodynamic forces computational flow

Aerodynamic databanks contain global forces and moments as well as their distributions across the aircraft. The values are either computed theoretically or are test results. They are mainly dependant on the vertical and lateral angle of attack and the different control surface deflection angles.

As the simulation is done in the time domain and the unsteady aerodynamic forces are given in the frequency domain, there is a need to transform them into the time domain. One way of achieving high accuracy with a reasonable computational burden is the use of Finite Impulse Response (FIR) filters combined with a modified Hanning–window [5]. The aerodynamic forces are added to the equations of motions of the purely structural part.

Mechanical–Hydraulic Part of Actuators

An electrohydraulic aircraft control–surface system mainly consists of the following components: cylinder, piston, spool value, hydraulic lines, hydraulic fluid, orifices, accumulators, controller, mechanical linkages to the structure and the control surface (Fig. 8).

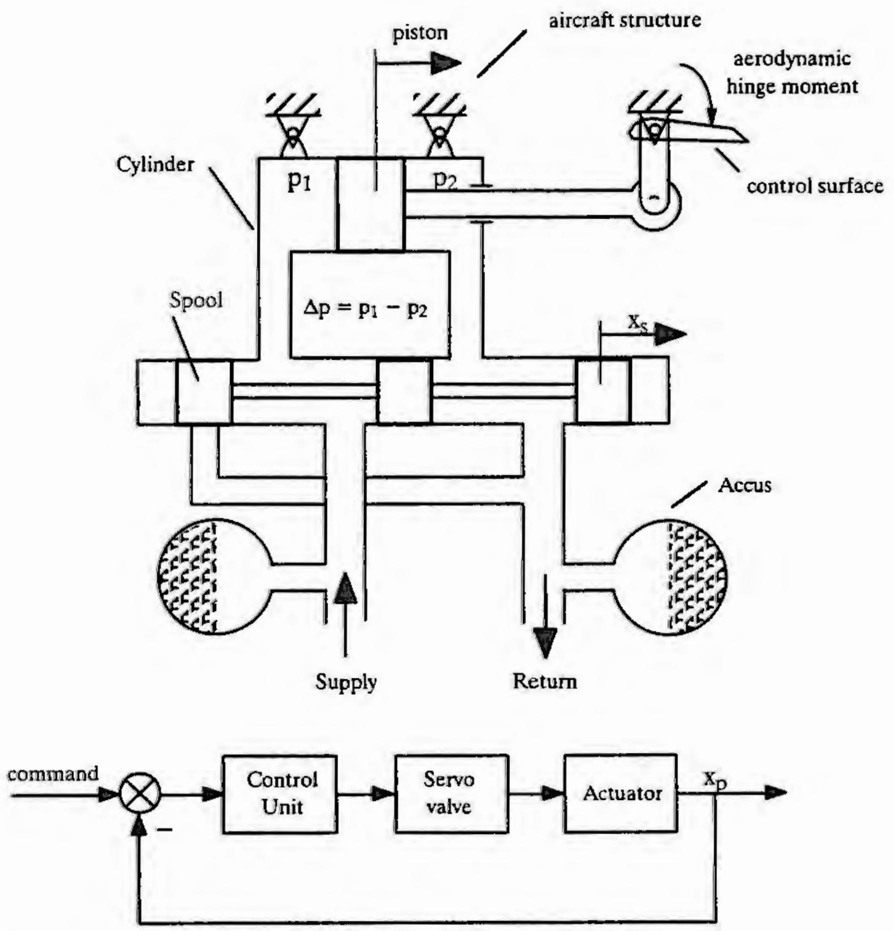


Fig. 8 Aircraft control surface servo model

Some Model Equations

Piston and Load dynamics

$$\underbrace{m_p * \ddot{x}_p}_{\text{Inertial force}} = \underbrace{-k}_{\text{Control surface force}} \quad \underbrace{- B_v * \dot{x}_p}_{\text{Viscous friction}} \quad \underbrace{+ F_c}_{\text{Coulomb friction (piston, cylinder)}} \quad \underbrace{+ \Delta P}_{\text{Hydraulic force}}$$

Flow equation:

$$Q = (K_v * x_s) * \sqrt{\Delta P}$$

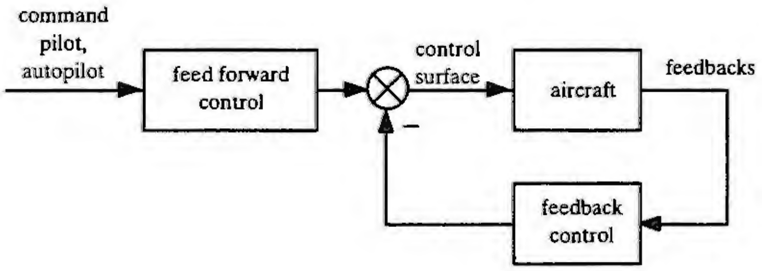
Q – flow
$(K_v * x_S)$ – nonlinear flow coefficient
ΔP – pressure drop

Further nonlinear contributions are the accumulator loading / unloading, the pressure drops along the hydraulic lines, the influence of the aerodynamic hinge moment and the travel limits. The coupling to the structure and to the control surface occurs at the bearings and through the mechanism composed of levers, rods and torque tubes. Again looseness and frictions in the joints are a challenge to simulators.

Coupling of other Modules

Modules like engines and thrust or landing mechanisms and runway are treated similar to structure and aerodynamics. The interior modules (seats, passengers, cargo, ...) are often subjected to impact or crash szena-

rios. Structure may be loaded up to breakage. In addition human injury criteria are an important design criteria.
The modules Electrical Flight Control System (EFCS) and electric part of actuators have got low energy and are represented by block–diagrams with inputs and outputs. They initiate hydraulic or electric power and there is no coupling back from the high energy modules.



- digital control
- many non–linearities (thresholds, limiter, variable gain and time–constants, ...)
- protections
- alternate control laws

Fig. 9 Simplified structure of electrical control

The transfer from the continuous to discrete systems and back is done by A/D– and D/A–converters.

System of Differential Equations of Motion

After all modules are coupled we get the system of differential equations of motion of the total aircraft:

$$M * \ddot{x} + (D–A) * \dot{x} + (C–B) * x = E * b + F * c + G * t + H * l$$

$\ddot{x},\ x,\dot{}x$	–	vector of degrees of freedom and its derivatives
M	–	mass matrix
C	–	stiffness matrix
D	–	damping matrix
A, B	–	matrices of aerodynamic forces due to aircraft movement
E	–	matrix of gust forces
F	–	matrix of aerodynamic forces of control surfaces
G	–	matrix of engine forces
H	–	matrix of landing impact forces
b	–	gust speed
c	–	control surface deflection
t	–	engine control input
l	–	landing impact input

Integrated Hard– and Software Environment

We described the physical modeling of a large interconnected system based on model components form different engineering domains (Fig. 9). To simulate such a total system we ideally need one development environment from offline– and hardware–in–the–loop simulation to flight tests. As people from different engineerings disciplines are involved care must be taken of simple interfaces and an easy integration of the single contributions.
We bought an integrated hard– and software environment, which can be used during all steps of the design process from modeling to flight tests (Fig. 10). It consists of the 3 modules CASE–tool MATLAB / SIMULINK, the real–time hard– and software–world of dSPACE and a host computer [2], [3], [4].

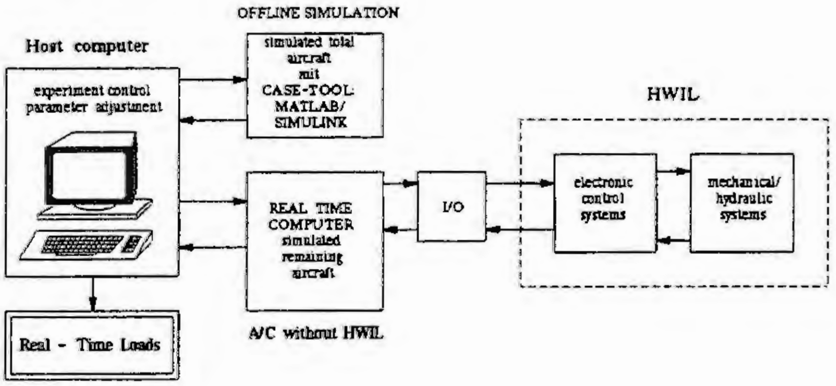

OFFLINE SIMULATION
Host computer
experiment control
parameter adjustment
simulated total aircraft mit CASE-TOOL: MATLAB/ SIMULINK
HWIL
REAL TIME COMPUTER simulated remaining aircraft
I/O
electronic control systems
mechanical/ hydraulic systems
Real - Time Loads
A/C without HWIL

Fig. 10 Development environment

Summary

- We presented one development tool usable within all project stages and for several people and different disciplines involved in the design.
- It is very expensive to develop software in–house. An integrated hard– and software environment offered by dSPACE, which is based on the CASE–tool MATLAB / SIMULINK and a high–speed and flexible real–time processor is a solution at a reasonable price.
- Module–oriented development approach eases teamwork, lowers design costs and increases quality; Care must be taken of simple and transparent interfaces to ease the integration process.
- As we are dealing with variable structure systems a matter of continuing research and development is the treatment of discontinuities and events.

References

[1] 7 Diplomarbeiten, Lehrstuhl für Regelungssysteme und Steuerungstechnik, Prof. Dr. Fasol, Ruhr–Universität Bochum, Authors: Ewald, Handschuh, Kalinowski, Merz, Müller, Rempe, Rinza

[2] Hanselmann, H: dSPACE GmbH "Hardware–in–the–loop Simulation as a Standard Approach for Development, Customization, and Production Test", International Congress and Exposition, Detroit, Michigan, March 1–5, 1993

[3] Otterbach, Kiffmeier dSPASE "Eine neue Generation hochleistungsfähiger Echtzeitsimulatoren auf Basis des Alpha AXP^{TM} Prozessors", ASIM '96, Dresden, 16.–19. Sep. 1996

[4] SIMULINK: "A Program for Simulation Dynamic Systems", User's Guide, March 1992, The MathWorks Inc., USA

[5] Brigham, O.E.: "The Fast Fourier Transform and its Applications", Prentice–Hall International, Inc. 1988

[6] Elmquist, Cellier, Otter:. "Object–oriented Modeling of Hybrid Systems", ESS'93, The Netherlands – Delft, Oct. 1993

[7] Hessel, E: "Integration moderner Methoden und Werkzeuge in eine durchgängige Entwicklungsumgebung zum Entwurf mechatronischer Systeme", ASIM–Simulation Technischer Systeme, DASA–Airbus, Hamburg, 20.–21. Feb. 1995

STRUCTURE OF RAINFALL-RUNOFF NONLINEAR MODEL WITH PHYSICAL INTERPRETATION

C. Verde & C. Cruickshank
Instituto de Ingeniería, UNAM
PB 70-472, 04510 Coyoacán DF, México
Fax (52)-56228091, verde@servidor.unam.mx

Abstract. Problems of modeling for rainfall-runoff relations are discussed and the need for developing models on the basis of hydrological concepts is emphasized. Particularly a non-linear lumped model structure with surface and subterranean reservoirs in parallel is proposed. The model accounts for linear storage, antecedent precipitation index, or soil moisture factor, and time delay of the river response. In order to satisfy the mass balance, the net rainfall (inputs) to the model are divided between the two reservoirs. The non-linear effects of the process are concentrated in the rainfall terms of the model and their structure is validated taking into account the physical meaning of the hydraulic process. As the tool to calibrate the specific parameters set an Extended Kalman Filter is used. The structure has been applied with success to several watersheds with different characteristics.

Keywords: Non-linear rainfall-runoff model, lumped model, on line estimation, Extended Kalman Filter.

Introduction

Different approaches to the modeling of rainfall-runoff processes have been proposed during the last decades. Taking into account that the precipitation-runoff system is nonlinear, distributed, and involves stochastic variables, the main problem in most of the lumped models is the lack of connection with physical meaning and significance of models components. Todini [7] presented a good historical review of the methods to estimate the parameters and the model structures mentioning their main disadvantages. Moreover, the fact that the precipitation has a spatial distribution and that it cannot be controllable demands additional effort to propose and validate a general model for the rainfall-runoff process. On the other hand the complexity and environmental changes in the eco-systems make today necessary to have models which can take into account changes and uncertainties of the system and can be easily adapted on line according to real conditions.

From an identification point of view, other issues for nonlinear models is the determination of their structure, since most of the methods assume, a priori, a given structure. Haber and Unbehauen presented in [6] a very good survey about the input/output approaches to identify structure of polynomial terms in nonlinear dynamic systems and pointed out that the structure identification is only part of the whole identification procedure. Some general block-oriented models are proposed which consist of static and dynamic linear terms. However some of the structures require particular input patterns and therefore they cannot be used in the rainfall-runoff problem.

The above mentioned facts motivated to present in 1994, [2] a lumped model for the rainfall-runoff process with physical interpretation and based on the block-oriented model with static nonlinear terms for the rainfall and dynamic linear terms for the reservoirs. The first version of the proposed model considered only one storage element and it was validated estimating the parameters by a Least Square Algorithm, LSA. In other words a discrete model with the structure given by

$$Q_T (i + 1) = c\, Q_T (i) + f\,(P(i - r)) + e(i) \qquad (1)$$

was considered, where Q_T represents the total runoff volume at the catchment outlet, the input term $f(P(i - r))$ is a nonlinear function of the total volume of the precipitation P over the watershed, ***r*** is the delay or arrival time, e is associated to the model error and ***i*** is the time increment. Later on, it has been shown in [3] that the parameter estimation of (1) can be improved when an Extended Kalman Filter, EKF, is used instead of the

LSA. Particularly the EKF yields time variant parameters during long identification periods of time showing the variation of water demand and changes of soil use along the history. This model allowed to explain more than 90% of the runoff variability. Moreover it was possible to determine coefficients of surface and subsurface runoff. Because less satisfactory results were obtained for daily and hourly rainfalls, i.e. for inputs with high frequency components, it was concluded that a high frequency dynamics should be added to the model.

In this paper, the model structure has been improved considering a dynamic model with two reservoirs in parallel, adding pass terms of the direct rainfall volume, and holding the nonlinear structure of the input as before. The total rainfall volume for a given period is divided between losses and two reservoirs or storage subsystems. One of the storages, called surface reservoir, is associated with the river network with low capacity and rapid response. The other one has a very slow response with large capacity and describes the subterranean volume effects. An advantage of the nonlinear structure is the capability of its generalization for more than one input. The proposed new model has been validated with real data in a vast variety of circumstances, time intervals from one hour to one month, watershed areas from a few to thousands of square kilometers [4]. In the following the results obtained for the Chanchos river in Uruguay are shown.

Description and Structure of the Physical Model

A watershed may be viewed as a system receiving rainfall as input; the destiny of the fallen water may be briefly described as follows; part of it is intercepted on the ground and later lost to the atmosphere; another part infiltrates to the ground and is retained there for some time but later also lost to the atmosphere through evapotranspiration from the vegetation. The soil and surface retention constitutes a first reservoir which is more or less wet according to the amount of previous rains; when there is rain, it gains in wetness, when there is no rain, it looses humidity. In the present model this process is described simply by an index of antecedent precipitation computed by a first order autoregressive model, as will be shown later (eq. 4). This index must have a limiting value which represents the maximum capacity of the soil and surface retention and should depend on the type and depth of soil, type of vegetation, and watershed slope.

Another route for the fallen water is channel runoff which travels along the surface channel network of the upper part of the watershed, part of which infiltrates and feeds subsurface deposits; this process defines two more reservoirs whose water is not lost before it arrives to the watershed outlet where it is measured (output); the first one is formed by the channel network itself with water traveling fast to the outlet; the other one is the subterranean deposit with water that travels slowly to the lower river stretches. It is only natural to think that when the first reservoir (the soil) is dry, the rain will be retained in it and the amount of water going to the channels will be small; as more humidity enters the soil more water will go as surface runoff. In the present model, the percentage of the rain that becomes surface and subsurface flow is considered, mainly, proportional to the antecedent precipitation index, allowing a smaller possibility of a direct linear response by the coefficient a_0 of eq. 3.

Figure 1 shows the block diagram of the proposed structure for one input in which z^{-1} means backward shift operator. The generalization for more inputs requires to consider a structure for each input similar to the one inside the dashed block of the diagram. From the conceptual catchment model of Fig. 1 it can be seen that the total rainfall volume for a given period is divided into two parts: losses and input to reservoirs. Losses include interception, water consumption, ponding and evapotranspiration which are unknown. Surface and subterranean reservoirs are considered lumped linear storage elements and the total runoff volume at the catchment outlet $Q_T(i)$ is given by a linear contribution of two subsystems: the surface volume, $Q_S(i)$, and the subterranean volume, $Q_U(i)$, i.e.

$$Q_T(i) = Q_S(i) + Q_V(i) \quad , \tag{2}$$

where i is the sample time increment. The variable v(i) of Fig. 1 represents the direct input volume which has a nonlinear behavior depending on the past rains via the antecedent precipitation index A(i) or moisture index. One can say that the volume v(i) is produced as a simple impulse immediately after the rain impulse occurs but its arrival to the catchment outlet can be delayed r sample periods. In other words, it may be interpreted as a variable unit response of the catchment, depending on the antecedent precipitation index A(i) or soil moisture.

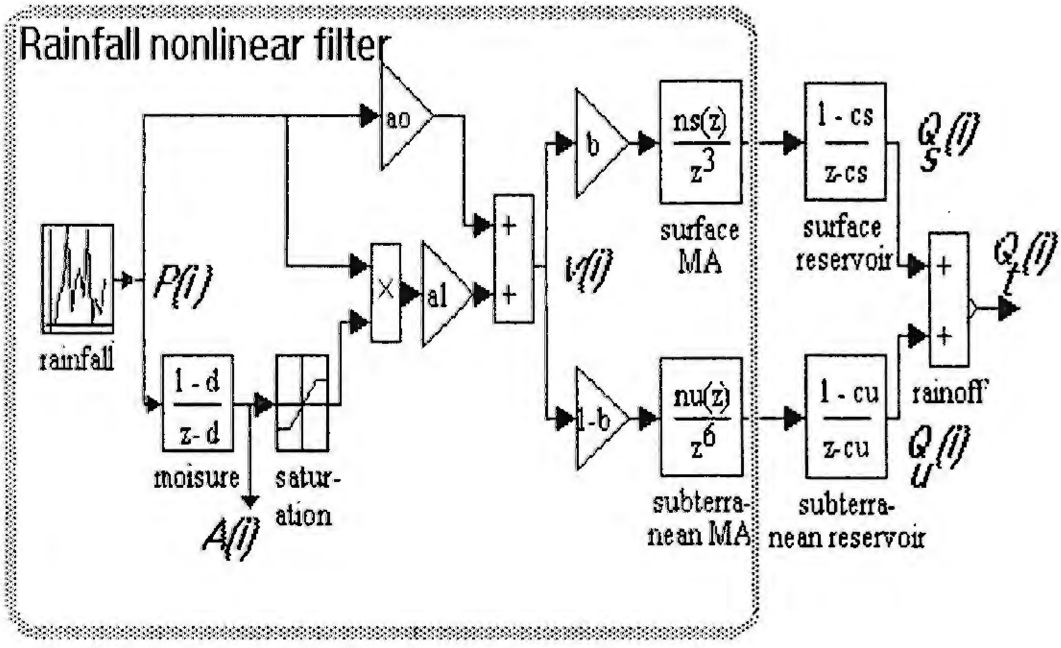


Fig. 1 Block diagram of the conceptual catchment model for one input

The general equation of the total input can be witten as:

$$v(i) = a_0\ p\ (i) + a_1\ p(i)\ A(i)\ , \tag{3}$$

where the antecendent precipitation index, API, of the erea is governed by

$$A(i + 1) = d\ A(i) + (1 - d)\ p(i) + e(i) \tag{4}$$

with ist upper constraint

$$A(i)\ <\ A_{max}\ =\ (1 - a_0\ )\ /\ a_1 \tag{5}$$

which can be obtained from the mass balance condition of eq. (3), since the runoff can not exceed rainfall. Experience with the model has shown that the forgetting value $0 \le d \le 1$ of the antecedent precipitation index in equation (4) can be taken near to one for monthly intervals, as 0.2 for daily intervals, and close to zero for hourly increments.

The behavior of each reservoirs is assumed linear governed by an ARMA model and given by

$$Q_S(i+1) = c_S Q_S(i) + b(1-c_S)\ \sum_{k=0}^{m_S} \gamma_S(k) v(i-k) e_S(i) \tag{6}$$

for the surface reservoir and

$$Q_u(i+1) = c_u\ Q_u(i) + b(1-c_u)\ \sum_{k=0}^{m_S} \gamma_u\ (k) v(i-k) e_S(i) \tag{7}$$

for the subterranean reservoir. In both subsystems $e_s(i)$ and $e_u(i)$ are model errors, and $g_s(k)$ and $g_u(k)$ are parameters which characterize the rise time of the impulse response of each subsystem. The coefficients c_s and c_u determine the dynamic response of both subsystems. The surface and subterranean runoff coefficients which define the gain of each subsystem are given by the terms bv(i)/p(i) and (1-b)v(i)/p(i) for each reservoir respectively can be readily obtained. The delay value of the input to the reservoir involved in the moving average tenn of eqs. (6) and (7) should be tried for each particular application, although a close approach to its value can be obtained through a rain-runoff cross-correlation diagram.

To avoid loss of physical sense, it is assumed that variables are non stationary. The model can be easily extended for the case of several inputs. It is to be noted, that this simplified model is non linear since the terms A(i)p(i) in equation (3) involve products of rainfall volumes.

Experimental Results

In order to show the type of model results, an application is presented here for the watershed of Chanchos river in Uruguay which received a millenary storm on march 1994 [5]; it has an area of 18.7 km^2 and has a rather impermeable surface (intemperized crystalline rock) with poor soil cover. Rain was measured in a recording station with a sample time of half an hour and therefore half an hour was the time interval considered in the identification process; data are shown in Fig.2. Runoff was calculated through the rising levels of a reservoir at the end of the catchment and the elevation area curve for the reservoir. This estimation is taken as the measured discharge or output of the process.

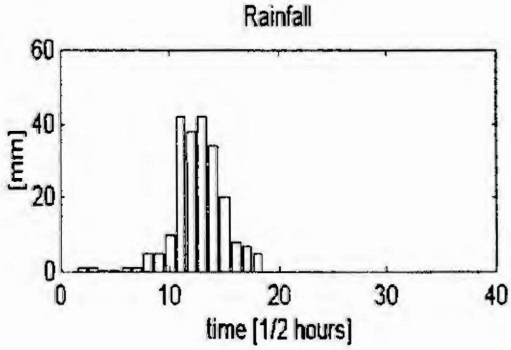


Fig. 2 Rainfall event used for the estimation

Fig. 3 shows the computed values using the proposed model for the evolution of the total runoff volume Q_T together with the evolution of the subterranean volume Q_U. From the rising and decay of both curves, it is concluded that the rising shape of the total runoff is mainly defined by the surface state behavior and the decay exponential shape by the subterranean state.

Errors among the real and computed total volumes are given in Fig. 4; the mean square error is 0.7 mm. Because of the great intensity of the rain, total runoff coefficients reach the value of one at the peak of the avenue as is shown in Fig. 5 where the surface, subterranean and total runoff coefficients are shown.

Fig. 6 shows the variation of the dynamics parameter of the surface reservoir, eq. (5), during the on line identification procedure. Its value does not remain constant, however the deviation around the nominal value is tolerable from a practical point of view.

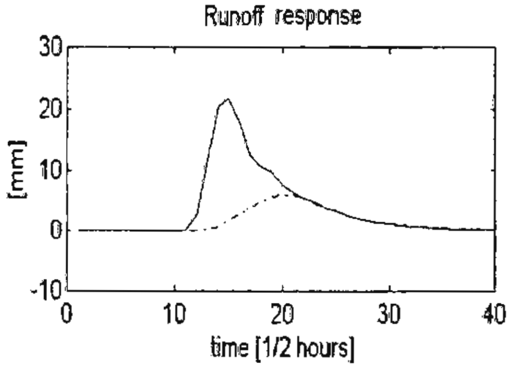


Fig.3 Estimations of total volume ___ and of subterranean volume .-.-

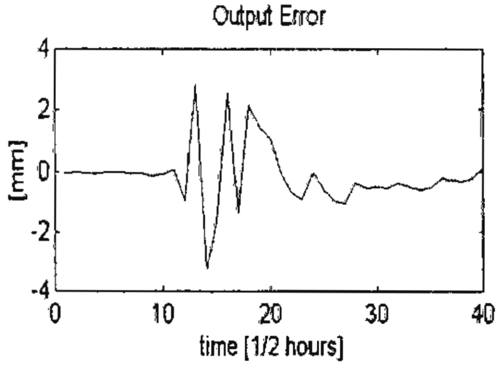


Fig. 4 Error between estimated and measured volume

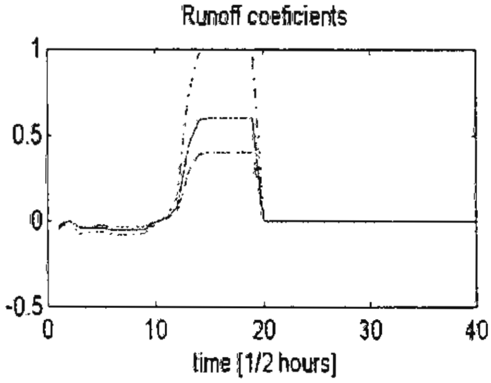


Fig. 5 Runoff coefficients for subterranean flow...., surface flow ---- and total flow -.-.-.

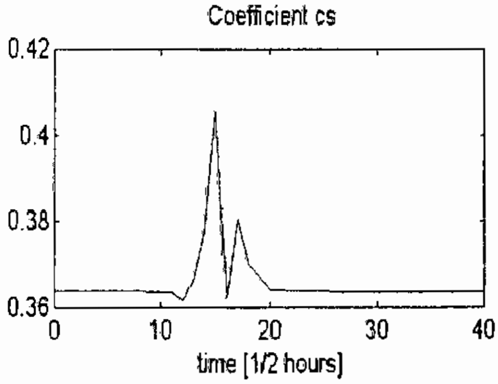


Fig. 6 Evolution of the estimation for the parameter c_s of the surface subsystem

Conclusions

A simple nonlinear model is introduced in this paper which can reproduce catchment processes for different time increments, sizes and shapes. The essential point in this model is that a physical significance is given to each term and also that the nonlinear character of the rising limb of hydrographs has been taken into account, while the recession part is linear.

References

[1] Bras R.L. & Rodriguez-Iturbe I. *Random functions in hydrology.* Academis Press, 1984.

[2] Cruickshank C. Funciones de transferencia en modelos de cuencas. *XVI Congreso Latinoamericano de Hidráulica,* 1994, V3, pp157-167, Santiago de Chile.

[3] Cruickshank C., Verde C. Some Examples of Simple Rainfall-Runoff Models with Dynamic Parameter Identification. *XI International Conference on Computational Methods in Water Resources,* Cancun, Mexico, July, 1996.

[4] Cruickshank C. Hacia un modelo generalizado lluvia escurrimiento en cuencas. *XVII Congreso Latinoamericano de Hidráulica,* pp 5, xx, 1996 Guayaquil, Ecuador (spanish).

[5] Genta J.L. and Charbonier F. Personal Communication, 1996.

[6] Haber R., Unbehauen H. Structure Identification of Nonlinear Dynamic Systems - A survey on Input/Output Approaches. 1990, *Automatica* Vol. 26, pp 651-677. GB.

[7] Todini E., Rainfall-runoff modeling: Past, present and future, *1988, J. of Hydrol.*, 100, 341-352.

MODELLING AND SIMULATION OF PURIFICATION PROCESSES IN WASTEWATER TREATMENT PLANTS USING THE MODULAR TOOLBOX KSIM

M. KOEHNE, M. SCHUHEN and D. SCHOENBERGER
ZESS Zentrum für Sensorsysteme und IMR Institut für Mechanik und Regelungstechnik
Universitaet Siegen, D-57068 Siegen

Abstract. A rough outline of the modular toolbox KSIM for dynamic simulation of purification processes in wastewater treatment plants is given. The features and characteristics of KSIM, the information transfer inside the simulation model, the design of the modules and the input of the module parameters are discussed. A simulation example is chosen to demonstrate the use of KSIM for the development of control strategies optimizing the wastewater treatment plant operation of nitrogen removal. Two different control strategies for the operation of an equalizing basin at the influent of the biological purification stage of a wastewater treatment plant are compared and assessed by means of the simulation results.

1. INTRODUCTION

Biological wastewater treatment in modern sewage plants contains the removal of organic and anorganic components like carbon, nitrogen and phosphorus resulting in high demands on the purification efficiency as well as the economic plant operation. Therefore, the computer aided simulation of sewage plants gets an increasing significance. Today, the dynamic simulation is an effective tool for the design of new plants or for the optimization of the operation of existing plants.

Suitable simulation tools and a simulation enviroment with plant components and process oriented model libraries are necessary to carry out simulation studies. The main features of those model libraries are their flexibility and the possibility of extension with respect to the requirements of the user.

The model library KSIM (KläranlagenSIMulation in German), which has been developed at the Institute of Mechanics and Control Engineering (IMR), is a plant component and process oriented library of this type. KSIM is used at the IMR mainly for research works in the area of sewage plant automation, but also for solving practical problems and teaching students in simulation.

2. THE MODULAR TOOLBOX KSIM

KSIM is based on the simulation and computation environment Matlab/Simulink of MathWorks /Math/. The elaborated graphical user interface and user-friendly operation are the main reasons for using the environment Matlab/Simulink. New modules can be realized using the fundamental operation blocks, the inherent system language or external available programming languages (C und FORTRAN).

The simulation model of a wastewater treatment plant or parts of a plant is composed of accomplished coded modules based on mathematical models, which can be taken from KSIM by the user. The linkage of the modules in the graphical editor of Matlab/Simulink is done by simple line connections.

The simulation is performed using the resources offered by Matlab/Simulink. Several possibilities are available for the input, output, presentation and processing of measurement data or simulation results. The whole functionality of Matlab/Simulink is preserved. Any flow sheet (e.g. singlestage, multistage or alternating) of wastewater treatment plants and processes can be modelled and simulated with the KSIM modules available and the resources offered by Matlab/Simulink.

KSIM consists of a number of sublibraries. Different modules of primary clarifiers, activated sludge processes, final clarifiers, equalizing basins, controllers and control strategies, plant components and auxiliary simulation tools are available.

Models of primary clarifiers:
Up to now this sublibrary only contains a simple delay model describing the wastewater flow rate inside the primary tank. We are working on detailed models considering the settling processes and the solubility of wastewater components too.

Activated sludge models:
Several application oriented models based on the biokinetics of the Activated Sludge Model No.1 and No.2 developed by the IAWQ* /HGGM86/ /HGMM94/ are included. Both models describe those processes of biological treatment, assumed to be essential for the carbon and nitrogen removal. Moreover, the Activated Sludge Model No.2 considers the biological and chemical phosphorus removal.
Furthermore, several modules performing reduced biokinetic models are available (e.g. a module describing carbon removal). For several applications the use of less detailed models turned out to be more effective. By this, besides a reduction of simulation time needed, the adaptation of the simulation models to a real plant or a part of a plant was facilitated.

Models of secondary clarifiers:
Two models of secondary clarifiers are available. The layer model according to the FLUX theory /Vita89/ offers a great variety of applications to the user. In order to describe the settling velocity of the sludge, it is possible to choose between the theories of Vitasovic /Vita89/ and Takács /TPN90/. The total volume of the secondary clarifier can be divided into an arbitrary number of layers (represented by completely stirred tanks). The geometry of the real secondary clarifier can be considered by the definition of different heights and areas of the layers. Besides the layer model, there is another module using the model of an ideal secondary clarifier (all particulate components will settle completely), only describing the transportation processes inside a secondary clarifier.

Equalizing basins:
For the simulation of equalizing measures in the influent of a wastewater treatment plant different equalizing basins are modelled. In order to be able to use them in a simulation model, control strategies for equalizing basin operation have to be given. Several modules with predefined strategies are filed in a sublibrary.

Controller and control strategies:
Besides the modules comprising strategies for equalizing basin operation, this library contains a selection of standard controllers (e.g. PI and PID controllers). They are realized as continuously or discrete working controllers.

Plant components:
This sublibrary comprises modules for the description of mixers and dividers. Moreover there are modules representing different pumps and measurement instruments. Their use allows the simulation of any plant flow sheet.

Auxiliary simulation tools:
This sublibrary mainly comprises auxiliary tools for the graphical presentation of the results during the simulation.

3. LAYOUT OF THE MODULES

The layout of the KSIM modules follows closely the symbolism of piping & instrument diagrams (P & I - diagrams) used for the description of the plants. In doing so, creation and comprehensibility of a simulation model is facilitated. Fig. 1 shows the P & I - diagram of the biological purification stage of a wastewater treatment plant with an equalizing basin (parallel to the main stream) and the corresponding KSIM simulation model. The wastewater flow rate inside the aeration tank (denitrification and nitrification) is described by completely stirred tank reactors.

* International Association on Water Quality, former International Association on Water Pollution Research and Control (IAWPRC).

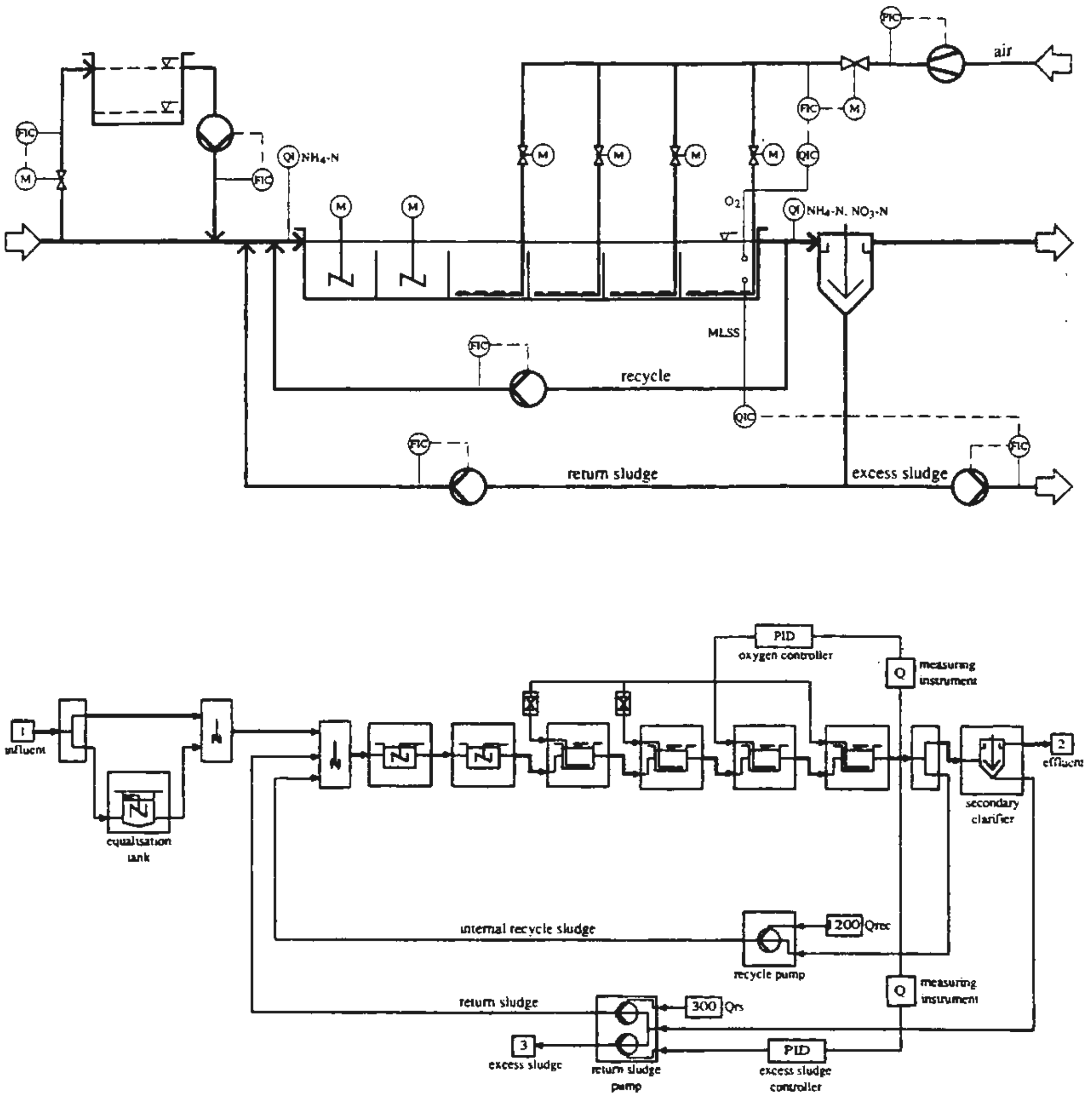


Fig. 1: P & I - diagram and corresponding KSIM simulation model of the biological purification stage of a wastewater treatment plant with an equalizing basin.

4. FLOW OF INFORMATION INSIDE A SIMULATION MODEL

Inside a simulation model, the output of a preceding module constitutes the input of the following module. The information is handed over as a vector. One distinguishes system vectors and auxiliary vectors (Fig. 2b). The structure of the system vector (Fig. 2a) is closely defined. It contains all information necessary for the simulation of a system (time, hydraulic parameters, concentrations, temperature). Auxiliary vectors are those vectors needed for the description of a model, not having the structure of a system vector. In Fig. 2 the module of the measurement instrument MS measures one particular parameter and delivers the input (auxiliary vector 1) for the controller. Auxiliary vector 2 contains the output (manipulated variable) of the controller. The defined structure of the input and output vectors realized during model programming allows any combination of arbitrary modules to create a simulation model. This compatibility survives if new modules are created according to the structure of the system vector.

System vector

Pos.	information	
1	t	time
2	Q	total flow rate
3	R	return sludge
4	S	excess sludge
5	Z	internal recycle sludge
6 to 6+i	Xi	particulate components
7+i to n-1	Sj	soluble components
n	T	temperature

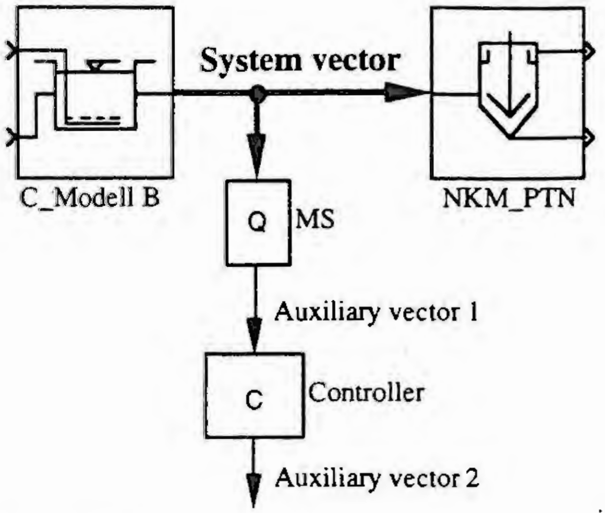
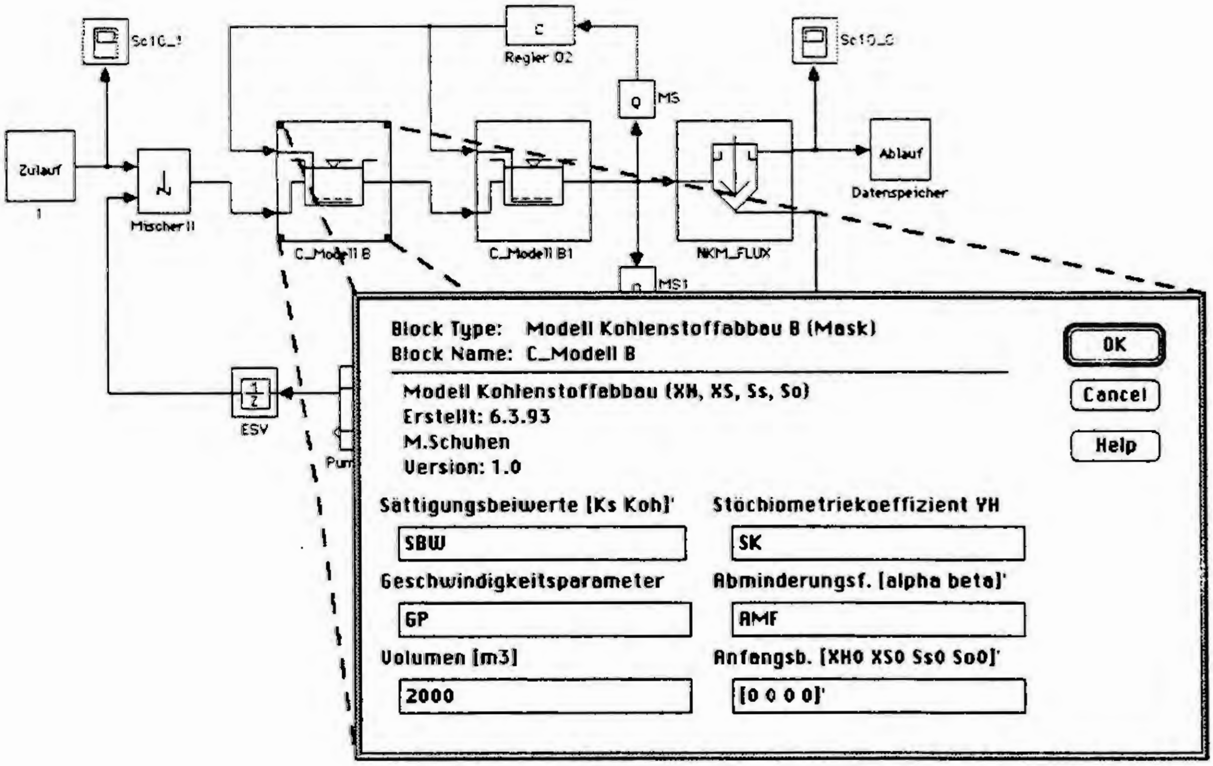


Fig. 2: Flow of information by system vectors and auxiliary vectors.

5. INPUT OF THE MODEL PARAMETERS

The parameters and initial conditions to be defined for a simulation can be chosen individually according to the situation in a real plant. The inputs of the model parameters are executed either locally by calling the dialog box of the module or globally by parameter files. Fig. 3 shows the dialog box of the module used for the description of carbon removal inside a particular tank reactor of the biological purification stage of a wastewater treatment plant. The dialog box is opened by clicking the module. In this case, the volume and the initial conditions are defined locally. The stoichiometric coefficients and kinetic parameters of the biokinetic model based on the Activated Sludge Model No. 1 are defined globally.



Block Type: Modell Kohlenstoffabbau B (Mask)
Block Name: C_Modell B

Modell Kohlenstoffabbau (XH, XS, Ss, So)
Erstellt: 6.3.93
M.Schuhen
Version: 1.0

Sättigungsbeiwerte [Ks Koh]': SBW
Stöchiometriekoeffizient YH: SK
Geschwindigkeitsparameter: GP
Abminderungsf. [alpha beta]': AMF
Volumen [m3]: 2000
Anfangsb. [XH0 XS0 Ss0 So0]': [0 0 0 0]'

Fig. 3: Dialog box of the module representing carbon removal.

6. SIMULATION EXAMPLE

By a simulation example the efficiency of using an equalizing basin in the influent of the biological purification stage of a wastewater treatment plant concerning nitrogen removal is demonstrated. Two different strategies are presented and assessed.

Plant description: The biological purification stage of the simulated wastewater treatment plant (Fig.1) is a plant with predenitrification. The oxygen concentration is measured at the end of the nitrification tank and controlled to a value of 2 g/m^3. In tanks using air controllers the manipulated variable is the air flow. Depending on tank volumes, the air is distributed evenly to each unit. The return and excess sludge flow rates inside the purification stage are regulated to realize an average concentration of particulate components of 3.5 g/m^3. The volume of the aertion tank is 5000 m^3. 3585 m^3 are designated for nitrification, 1415 m^3 for denitrification. The secondary clarifier has a volume of 4500 m^3, the equalizing basin of 1000 m^3. The latter is charged by a flow rate-controlled weir and discharged by pumps.

Simulation Model: The denitrification volume is separated into two, the nitrification volume into four completely stirred tank reactors. As biokinetic model for the processes inside the biological purification stage, the Activated Sludge Model No. 2 is used. The secondary clarifier is modeled as an ideal clarifier (supposing a complete settling of the particulate components) consisting of four stirred tanks. The equalizing basin is represented by only one completely stirred tank.

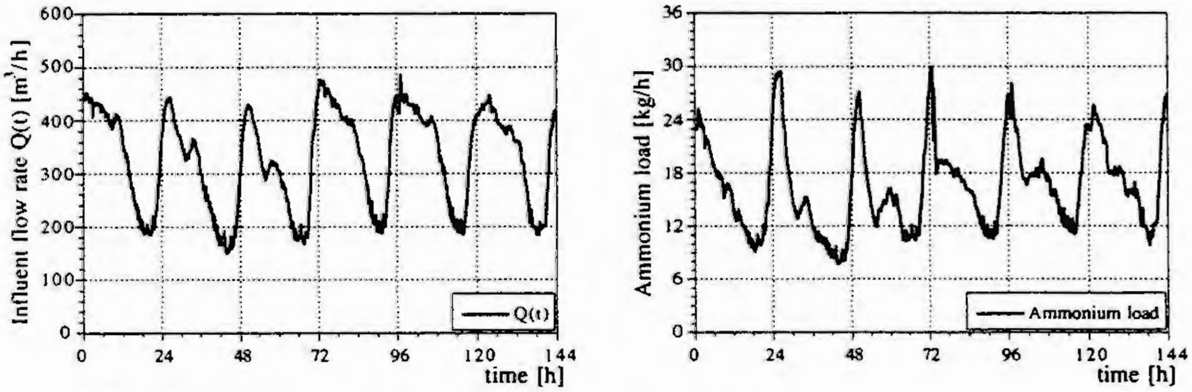


Fig. 4: Influent wastewater flow rate Q(t) and ammonium (NH_4-N) load.

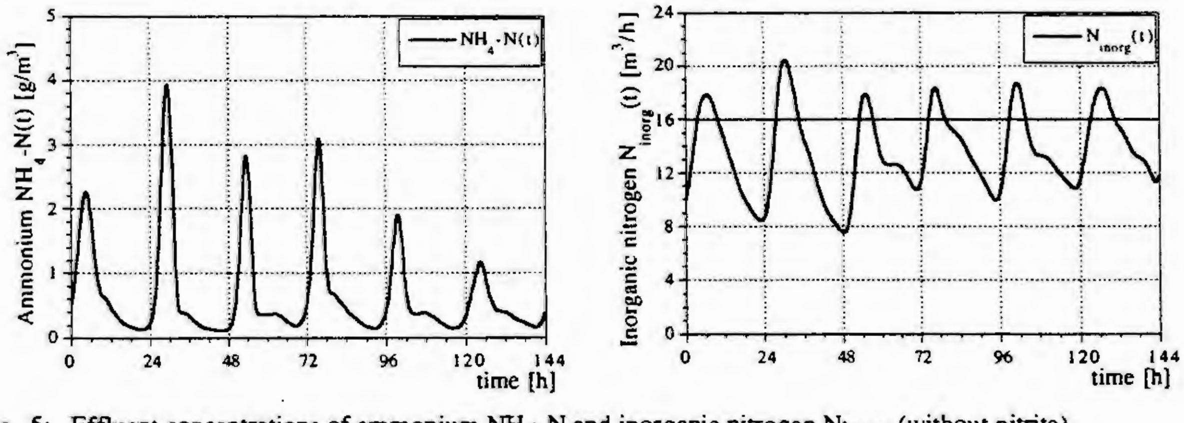


Fig. 5: Effluent concentrations of ammonium NH_4-N and inorganic nitrogen N_{inorg} (without nitrite).

As parameters of interest, here the influent wastewater flow rate Q and the ammonium (NH_4-N) load (product of Q and the NH_4-N concentration) are presented (Fig. 4). This figure represents a typical dry weather load situation. The operator of this wastewater treatment plant is demanded not to exceed effluent concentrations of 5 g/m^3 for NH_4-N and 16 g/m^3 for the inorganic nitrogen N_{inorg} (here ammonium NH_4-N + nitrate NO_3-N). Simulation results for the operation without equalizing basin (Fig. 5) show, that for several hours a day the guidelines are not met. This is due to the fact, that denitrification (NO_3-N removal) can not take place completely, because there are not enough readily biodegradable organic substrates. In this paper, the expensive addition of external carbon sources (e.g. acetate or methanol) will not be taken into consideration.

Strategy 1:
This strategy provides a compensation of the ammonium load in the influent of the plant. This load is restricted to a maximum value F_{max}. In case, the actual load exceeds this value, the overhead is stored in the equalizing basin. If the load remains under F_{max}, stored wastewater is added to the purification process again. In order to reduce the load variation according to the actual influent situation, it is necessary to choose F_{max} in conformity with the actual load situation of the plant. For this F_{max} is calculated to be the moving average of the flow rate of the past 24 hours:

$$F_{max}(kT) = \frac{f}{24} \sum_{k-24}^{k} Q(kT) \cdot NH_4 - N(kT) \qquad (1)$$

k: Running variable
T: Sampling time [h].
f: Safety factor, f≥1
NH_4-N(kT): Ammonium concentration at the influent of the equalizing basin [g/m^3].
F_{max}(kT): Maximum value of the NH_4-N load (starting storage of wastewater) [kg/h].
Q(kT): Wastewater flow rate entering the aeration tank [m^3/h].

Strategy 2:
In this case, equalizing basin operation depends on the N_{inorg} concentration (without nitrite) determined in the effluent of the purification stage. The storage of wastewater starts, if the N_{inorg} concentration exceeds a maximum value $N_{inorg,max}$. The volume of wastewater to be stored is calculated according to the change in N_{inorg} concentration during the span of time between the measurements.

$$Q_{zu}(kT) = Q_{zu,max} \cdot \frac{N_{inorg}(kT) - N_{inorg}((k-1)T)}{\Delta N_{inorg,max}} \qquad (2)$$

N_{inorg}(KT): Inorganic nitrogen N_{inorg} concentration (without nitrite) [g/m^3].
$N_{inorg,max}$: Maximum value of the N_{inorg} concentration (starting storage of wastewater) [g/m^3]).
$\Delta N_{inorg,max}$: Maximum value of the change in N_{inorg} concentration [g/(m^{3}h)].
Q_{zu}(kT): Wastewater flow rate entering the equalizing basin [m^3/h].
$Q_{zu,max}$: Maximum wastewater flow rate being allowed to enter the equalizing basin [m^3/h].

The wastewater flow rate charging the equalizing basin is restricted to a value always permitting a minimum flow rate of 50 m^3/h to enter the aeration tank. Stored wastewater is added to the purification process, if the N_{inorg} concentration decreases again. The addition is determined to avoid a renewed rise in N_{inorg} concentration, caused by an excessive flow rate. Taking into consideration the particular local maximum N_{inorg} concentration, the volume to be added is calculated according to:

$$Q_{ab}(kT) = Q_{ab,max} \cdot \left[\frac{N_{inorg}{}^*}{N_{inorg}(kT)} - 1 \right] \qquad (3)$$

N_{inorg}*: Local maximum value of the N_{inorg} concentration (without nitrite) [g/m^3].
Q_{ab}(kT): Wastewater flow rate to be added from the equalizing basin [m^3/h].
$Q_{ab,max}$: Maximum flow rate allowed to be added from the equalizing basin [m^3/h].

Simulation results:
Both strategies partly lead to evident reduction in nitrogen concentration in the effluent of the aeration tank (Fig. 6). In both strategies NH_4-N effluent concentration remains under 1 g/m^3. Applying strategy 1, the prescribed N_{inorg} effluent concentration of 16 g/m^3 is violated at no time. Using strategy 2 it is exceeded negligibly only twice. Looked at it as a whole, applying strategy 2 means higher effluent concentrations than applying strategy 1. This is due to the time delay between the influent and effluent of the aeration tank. Using strategy 2 always results in a delayed storage or addition of wastewater. As a consequence it is impossible or at least only partly possible to reduce the peaks in the ammonium load at the influent of the purification stage (Fig. 7). Compared to strategy 1 this is crucial. On the basis of these simulation results, the use of strategy 1 is more suitable.

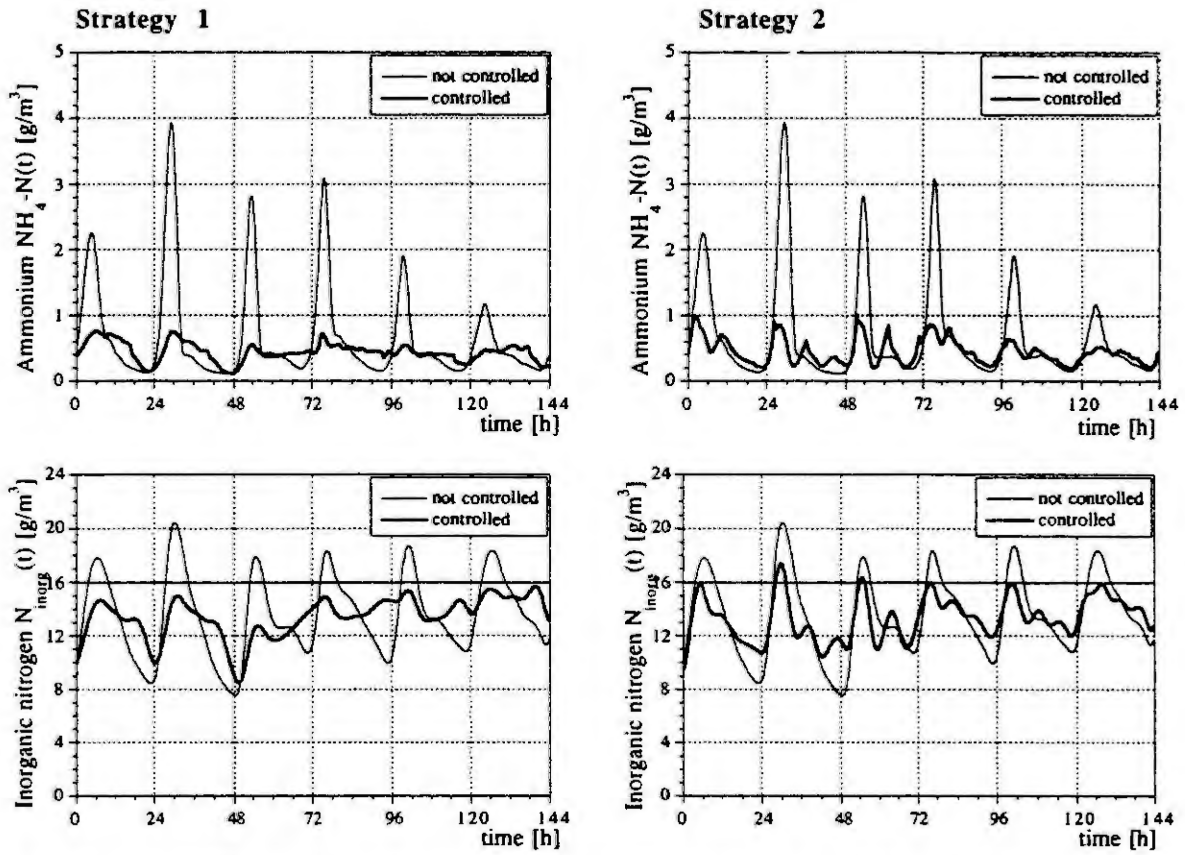


Fig. 6: Controlled effluent concentrations of ammonium and inorganic nitrogen (without nitrite).

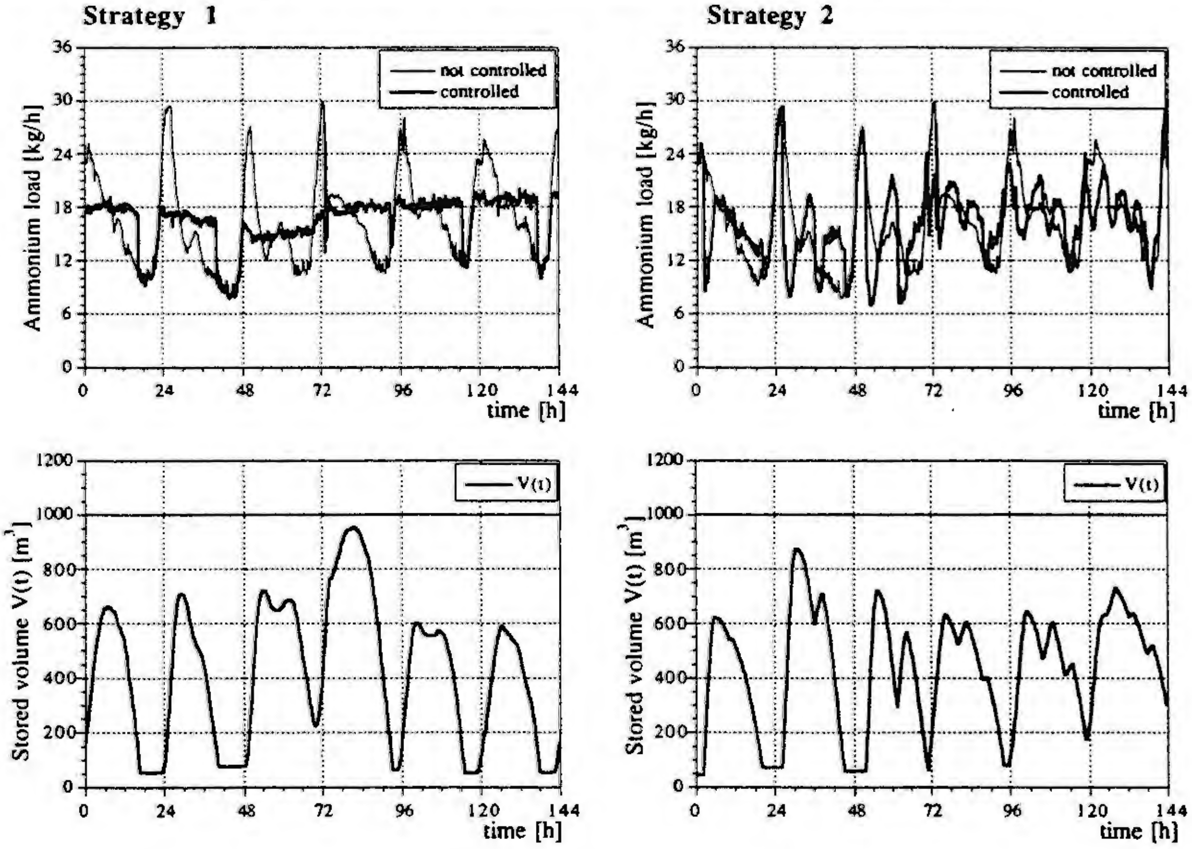


Fig. 7: Controlled influent ammonium load and stored volume V(t) in the equalizing basin.

For strategy 2, replacing the NH_4-N concentration /KSZ94/ by the N_{inorg} concentration in the effluent of the purification stage leads to comparable results. However, in this case wastewater storage already has to be started at very low NH_4-N concentrations (<0.5 g/m^3) to ensure that wastewater volumes comparable to strategy 1 (Fig. 7) are stored. It is doubtful, wether these low concentrations can be determined reliably by modern on-line process analyzers.

In simulation the use of an equalizing basin has turned out to be basically positive. A disadvantage of both strategies is the impossibility to use the volume of the equalizing basin optimally. For example, it is impossible to determine the precise equalizing basin volume required to meet a desired effluent concentration for a given influent load.
The decision at which effluent concentrations or loads wastewater has to be stored or restored anyway depends on the particular plant and the chosen compensation strategy. Morerover it is impossible to anticipate wether an action was necessary or not.
At the moment the IMR works on compensation strategies based on model predictive controllers, avoiding the above-mentioned disadvantages /HSK96/.

7. SUMMARY

By the integration of the modular toolbox KSIM with the simulation and computation environment Matlab/Simulink the user is provided with an universally employable and user-friendly simulation tool. With regard to new developments and knowledge in the field of wastewater purification, for the future a further development is planned.
The experiences made with dynamic simulation of wastewater purification processes have proven different demands and problems to be solved. In this paper the dynamic simulation with KSIM was used to compare two different control strategies for the operation of an equalizing basin at the influent of a wastewater treatment plant. Concerning the simulation results, one has to bear in mind, that the validity of dynamic simulation results is restricted. For this, simulation results often only give qualitative statements.

REFERENCES

/HGGM86/ Henze, M., Grady, C.P.L, Gujer, W., Marais, G.v.R, Matsuo, T.:
(IAWPRC Task Group on Mathematical Modelling for Design and Operation of Biological Wastewater Treatment), Activated Sludge Model No.1, Report, 1986

/HGMM94/ Henze, M., Gujer, W., Mino, T., Matsuo, T., Wentzel, M.C., Marais, G.v.R:
'Activated Sludge Model No. 2', IAWQ Scientific and Technical Reports No.3, IAWQ, London, 1994

/HSK94/ Hoen, K., Schuhen, M., Köhne, M.:
Dynamische Simulation von Kläranlagen. Ein Hilfsmittel für den planenden Ingenieur? Korrespondenz Abwasser 41 (1994), S. 760-770

/HSK96/ Hoen, K., Schuhen, M., Köhne, M.:
Control of nitrogen removal in waste water treatment plants with predenitrification, depending on the actual purification capacity, Wat.Sci.Tech. 33 (1996), No.1, pp. 223-236

/KHS94/ Köhne, M., Hoen, K., Schuhen, M.:
Modelling and simulation of final clarifiers in wastewater treatment plants, Contribution to the 1st IMACS Symposium Mathmod Vienna, 1994

/KSZ94/ Krauth, K., Schwentner, G., Zerrer, H.:
Optimierung der Nitrifikation durch die Zwischenspeicherung von Abwasser am Beispiel der Kläranlage Bietigheim-Bissingen, Korrespondenz Abwasser 41 (1994), S. 751-758

/Math/ The MathWorks Inc.: Matlab/Simulink, 24 Prime Park Way, Natick, Mass. 01760

/Scuh93/ Schuhen, M.:
Modellbibliothek KSIM, IMR-Bericht 14-93, Diplomarbeit UGH Siegen, 1993

/TPN90/ Takács, I., Patry, G.G., Nolasco, D.:
A Generalized Dynamic Model of the Thickening/Clarification Process, Advances in Water Pollution Control Series, 10, IAWPRC Workshop, Elmswood, 1990, pp.487-493

/Vita89/ Vitasovic, Zdenko:
Continuous Settler Operation: A Dynamic Model, In: Dynamic Modeling and Expert Systems in Wastewater Engineering, (ed. by Patry G.G. and Chapman D.), Lewis Publishers, Michigan, 1989

NONLINEAR DYNAMICS MODELLING VIA OPERATING REGIME DECOMPOSITION

Th. H. Göttsche and K. J. Hunt
Intelligent Systems Group, Daimler-Benz Systems Technology Research
Alt-Moabit 96a, D-10559 Berlin, Germany
T. A. Johansen
SINTEF Automatic Control
O.S.Bragstads plass 8, N-7034 Trondheim, Norway

Abstract. This paper describes a method for modelling nonlinear systems using operating regime decomposition. The resulting model structure is known as a local model network (LMN); it consists in the smoothly-weighted combination of a number of simple local models, each of which is valid in a particular operating regime of the plant. A software tool for modelling and identification with this approach has been developed and is described in the paper. A brief survey of applications of the regime-decomposition method is given.

Introduction

Nonlinear dynamical systems can be decomposed into a number of distinct modes of operation with simpler dynamical behaviour. When it is possible to characterise these operating regimes using model variables (the 'scheduling vector'), then it is possible to create a global nonlinear model by the combination of a set of relatively simple local models that captures the relatively simple behaviour within each operating regime.

Within each operating regime the system is described by a simple model; frequently, a low-order linear local model is sufficient for characterisation of local behaviour. The operating regimes are assumed to overlap, and this is captured in the way in which the local models are combined: they are smoothly interpolated using a set of multi-dimensional basis functions which are driven by the scheduling vector. This modelling approach has also been called a Local Model Network (LMN). Full details of the method can be found in the references [1, 2, 3]. The design of nonlinear controllers using local decompositions is described in [4].

This paper considers such a class of nonlinear dynamical systems, and describes a methodology for empirical identification of nonlinear models using plant data.

Following a description of the LMN structure and estimation algorithms, a software tool which embodies the technique is described. Finally, a brief survey of applications of the regime-decomposition method is given.

System Definition

We consider nonlinear dynamic systems having the form

$$\begin{aligned} y(t) &= f(y(t-1),\cdots,y(t-na),u(t-k),\cdots,u(t-k-nb))+e(t) \\ &=: f(\psi(t-1))+e(t) \end{aligned} \tag{1}$$

with the *information vector*

$$\psi^T(t-1) = [y(t-1),\cdots,y(t-na),u(t-k),\cdots,u(t-k-nb)].$$

Here, y is the system output, u is the input, and $k-1$ is an input-output time delay. f is some nonlinear function describing the system behaviour and e is a noise term. The system (1) is known as a NARX model (Nonlinear AutoRegressive, with eXogenous input).

Local Model Networks

We assume that at each time instant the system behaviour is characterised by an *operating point* vector $\tilde{\phi}(t) \in \Phi \subset \mathbf{R}^{n_{\tilde{\phi}}}$. The set of operating points Φ is called the *operating range*.

We define an *operating regime* Φ_i as a subset of the operating set (i.e. $\Phi_i \subset \Phi$) where we assume the system behaves in some uniquely characterisable way (for example, as a linear system). With each

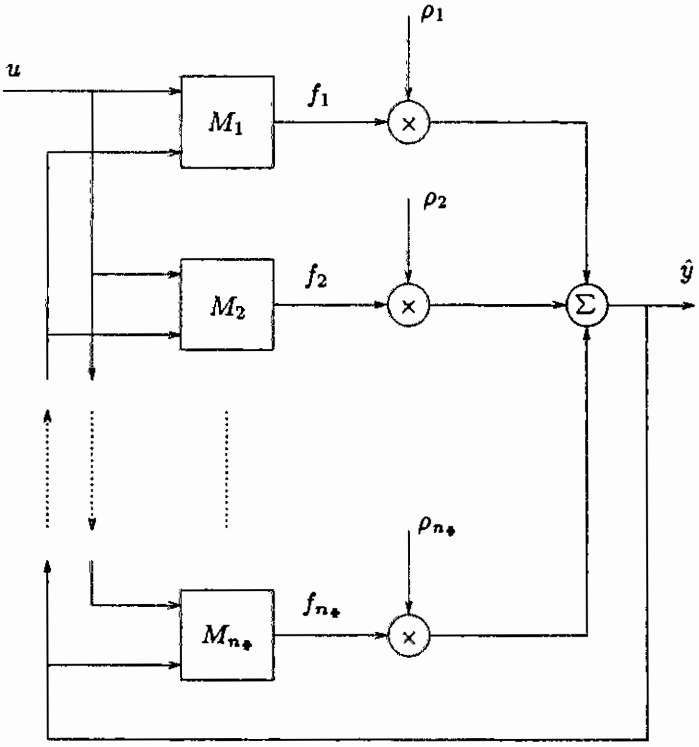


Figure 1: Local model network. The local models are $M_1 \ldots M_{n_\Phi}$, their respective outputs are $f_1 \ldots f_{n_\Phi}$ with
$f_i = [-\hat{y}(t-1) \ldots ; u(t-k) \ldots ; 1] \cdot [a_{i,1} \ldots ; b_{i,0} \ldots ; d_i]^T$. The local models share the common state ψ.

operating regime Φ_i we associate a positive semi-definite *validity function* $\rho_i(\tilde{\phi})$ which determines the validity of the local model given the current operating point $\tilde{\phi}$. The number of validity functions (and operating regimes) is n_Φ. The validity functions are defined such that $\rho_i(\tilde{\phi})$ is typically close to 1 for $\tilde{\phi} \in \Phi_i$ and tends in a smooth fashion to 0 for $\tilde{\phi} \notin \Phi_i$. The validity functions are furthermore constrained to satisfy

$$\sum_{i=1}^{n_\Phi} \rho_i(\tilde{\phi}) = 1 \tag{2}$$

for all $\tilde{\phi} \in \Phi$. Then, for the general nonlinear system (1), given (2) the following is obviously true

$$f(\psi(t-1)) = \sum_{i=1}^{n_\Phi} f(\psi(t-1))\rho_i(\tilde{\phi}(t-1)). \tag{3}$$

Since by definition $\rho_i(\tilde{\phi})$ is near 1 for $\tilde{\phi} \in \Phi_i$ and almost 0 otherwise, we can substitute f in the right-hand-side of (3) with some f_i which is a good local approximation to f in Φ_i. Our approximate system model becomes

$$y(t) = \sum_{i=1}^{n_\Phi} f_i(\psi(t-1))\rho_i(\tilde{\phi}(t-1)) + e(t). \tag{4}$$

In this model the $f_1, f_2, ..., f_{n_\Phi}$ are locally valid models which are smoothly interpolated by the validity functions ρ_i to produce the overall model. The model of equation (4) is referred to as a *local model network*, or LMN for short. The LMN structure is depicted in figure 1.

It should be emphasised that the local models do not have their own local state. Rather, they share a common 'global' state as defined by the information vector ψ.

The LMN in the form described above was first presented and analysed by Johansen and Foss [1, 2, 3]. Closely related approaches are described in the following section. The LMN can be viewed as a generalisation of the standard basis function networks (e.g. radial basis function networks) where in essence the local models are constant output values rather than linear models of the input/output behaviour. This increased structural richness in general results in the requirement for the operating range to be less densely populated with basis functions. Second, the LMN basis functions have as input the operating point vector $\tilde{\phi}$ which generally has a lower dimension than the information vector ψ. Notice the qualitative distinction between $\tilde{\phi}$ and the information vector ψ. $\tilde{\phi}$ could be equal to ψ, but if the system is linear with respect to certain elements of ψ then these can be left out of $\tilde{\phi}$. In principle, $\tilde{\phi}$ may also incorporate external or auxiliary variables which affect the system characteristics.

Typically, we choose to work with linear local models in which case the local models will have the local outputs

$$f_i(\psi(t-1)) = \psi_{re}^T(t-1)\theta_i, \tag{5}$$

where the regression vector ψ_{re} is defined by

$$\psi_{re}^T(t-1) = [-y(t-1), \ldots - y(t-na); u(t-k), \ldots u(t-k-nb); 1] \tag{6}$$

and the local parameter vector is

$$\theta_i^T = [a_{i,1}, \ldots a_{i,na}; b_{i,0} \ldots b_{i,nb}; d_i]. \tag{7}$$

The LMN (4) smoothly combines the local outputs to give

$$y(t) = \psi_{re}^T(t-1) \sum_{i=1}^{n_\Phi} \theta_i \rho_i(\tilde{\phi}(t-1)) + e(t). \tag{8}$$

LMN Properties and Related Approaches

The approximation properties of the local model network (4) have been examined by Johansen and Foss [2]. It was established in [2] that the LMN can uniformly approximate the system (1) on a compact subset of the domain of f given a sufficient number of local models n_Φ. The result is based upon smoothness conditions on the nonlinear function f, and explicitly formalises the intuitive notion that the operating point vector $\tilde{\phi}$ should capture the nonlinearities of the system. Although an existence result, the theorem is constructive and delivers an upper bound on the approximation error.

A number of other modelling approaches are quite closely related to the LMN. Several authors have developed piecewise linear models (without smooth interpolation). These include Skeppstedt *et al* [5], and Billings and Voon [6]. There are many examples of multiple model approaches in the statistical literature. These include the Smooth Threshold AR model of Tong [7] and the state-dependent models of Priestley [8]. The relationship here can be made explicit by examining equation (8). This appears as a linear model with parameters which depend on the operating point (which corresponds to the 'state' in Priestley's terminology). The LMN provides a way of finitely parameterising the state-dependent model.

The LMN is also linked strongly to fuzzy inference systems. Under certain conditions the Takagi-Sugeno model of fuzzy inference [9] is functionally equivalent to the LMN. This functional equivalence is described in detail by Hunt *et al* [10, 11].

Local Controller Networks

An arbitrary controller for the NARX-system (1) is given by:

$$\begin{aligned} u(t) &= C(r(t)\cdots, y(t)\cdots, u(t-1)\cdots) \\ &=: C(\psi_c(t)). \end{aligned} \tag{9}$$

Here, r is a command signal and ψ_c is the controller information vector:

$$\psi_c^T(t) = [r(t)\cdots, y(t)\cdots, u(t-1)\cdots]$$

Given a plant model of the form (4) as a local model network, it is natural to exploit this structure for the controller design. For every operating regime Φ_i the system is characterised by the local model f_i, to a degree fixed by the function ρ_i.

The already given partition of the input-space of the plant can be exploited by designing a local controller C_i for every local model (see [4]), whose validity is given by ρ_i.

The global controller for the whole plant is then defined analogously to the local model network, by interpolating the local controllers with the validity functions. The local controller network (LCN) is then given by:

$$u(t) = C(\psi_c(t)) = \sum_{i=1}^{n_*} C_i(\psi_c(t)) \cdot \rho_i(\tilde{\phi}) \tag{10}$$

Local and Global Estimation Criteria

Given a series of input-output measurements $(u(t), y(t)), t = 1 \dots N$, and a set of model validity functions ρ_i we wish to determine the parameters θ_i in each of the local models (5). Determination of the structure and parameters of the validity functions is discussed in detail in [3, 12, 13]. We assume here that the local models are linear as described by equations (5)-(7). A global criterion for estimation of the parameters of the LMN is

$$J_N(\theta) = \frac{1}{N} \sum_{t=1}^{N} \alpha(t)(y(t) - \hat{y}(t|t-1,\theta))^2, \tag{11}$$

and the vector containing all the estimated model parameters ($\theta^T = [\theta_1^T, \theta_2^T, \dots \theta_{n_*}^T]$) is obtained as

$$\hat{\theta} = \arg\min_{\theta} J_N(\theta). \tag{12}$$

In (11) $\alpha(t)$ are observation weights which can be attached to each measurement, and the one-step-ahead predictor is found directly from (8)

$$\hat{y}(t|t-1,\theta) \quad = \quad \psi_{re}^T(t-1) \sum_{i=1}^{n_*} \theta_i \rho_i(\tilde{\phi}(t-1)). \tag{13}$$

A second possibility, as described in [13, 14], is to locally estimate the parameters of the local models (5) by defining separate locally weighted criteria for each operating regime. For the i-th local model the estimation criterion is

$$J_{i,N}(\theta_i) \quad = \quad \frac{1}{N} \sum_{t=1}^{N} \alpha_i(t)(y(t) - \hat{y}_i(t|t-1,\theta_i))^2, \tag{14}$$

$$\hat{y}_i(t|t-1,\theta_i) \quad = \quad \psi_{re}^T(t-1)\theta_i \rho_i(\tilde{\phi}(t-1)). \tag{15}$$

In this case the estimate of the local model parameter vector θ_i is given by

$$\hat{\theta}_i = \arg\min_{\theta_i} J_{i,N}(\theta_i). \tag{16}$$

For the global criterion it is possible to set $\alpha(t) = 1$ for all t, or to select $\alpha(t)$ to achieve desirable parameter tracking behaviour. For the local criteria, on the other hand, the weights must be chosen to take direct account of the interactions of the validity functions.[1] Our confidence in a given observation regarding its relevance for the i-th local model is directly reflected in the i-th validity function. The local weighting functions should therefore be set as

$$\alpha_i(t) = \rho_i(\tilde{\phi}(t-1)), \tag{17}$$

which results in a set of local criteria

$$J_{i,N}(\theta_i) = \frac{1}{N_i} \sum_{t=1}^{N_i} \rho_i(\tilde{\phi}(t-1))(y(t) - \hat{y}_i(t|t-1,\theta_i))^2. \tag{18}$$

Estimating the parameters on the basis of a global criterion or several local criteria leads to local models with qualitatively different properties. With local criteria, the aim is to find local models that are close to local linearisations of the system, while with a global criterion the aim is to find local models that when

[1] In the global case the validity functions appear directly in the global model predictor $\hat{y}(t|t-1,\theta)$ through definition of the regressor vector—see below.

interpolated give a global model that is close to the system. Hence, with a global criterion, it may not be possible to interpret the local models separately. On the other hand, the global model will typically give better predictions when it is identified with a global criterion, as long as the model structure is parsimonious. The differences between the global and local estimation criteria are discussed in detail in [14].

Integrated Software Environment

In this section we describe the MATLAB based integrated software environment for LMN development, and gain scheduled LCN design, analysis and implementation. The LMN modelling tool is called ORBIT (Operating Regime Based modelling and Identification Toolkit).

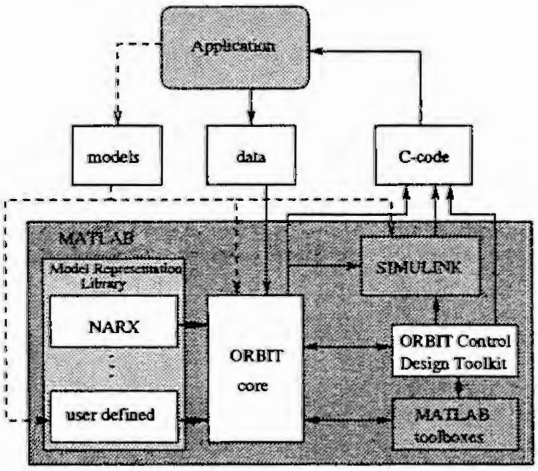


Figure 2: The ORBIT Environment.

An overview of the ORBIT software environment can be seen in Figure 2 and is described in more detail in [15, 16]. ORBIT is implemented in MATLAB. The ORBIT core contains the graphical user interface, parameter and structure identification algorithms, model validation algorithms, model database management as well as interfaces to standard MATLAB functions and toolboxes. ORBIT can support a wide range of model representations, including NARX, NARMAX and non-linear state-space models based on local models. However, only the NARX representation is currently implemented as part of the standard model representation library. The advanced user is free to include customised or general model representations in this library by programming the required MATLAB functions. ORBIT models can be made available as SIMULINK S-functions and blocks for validation and application. Local model parameters can also be interchanged with other MATLAB functions, including many functions in the MATLAB Control Toolbox and Signal Processing Toolbox. An application programmers interface (API) allows other MATLAB programs to access the ORBIT model database. ORBIT is extendible, i.e. its core model representation and functions are documented. Application data and models form the basis of model development in ORBIT. These can be pre-processed and analysed using generic MATLAB and SIMULINK functions before they are made use of in ORBIT.

Integrated in this software environment is the ORBIT Control Design toolkit that supports design of gain scheduled LCNs on the basis of ORBIT models. Local linear controllers are designed using LQG and pole assignment methods. The theory behind these design methods and some of their local and global properties are described in [4].

Applications

An overview of the most recent applications of local model networks is given in [17]. A comparative study of nonlinear models for the longitudinal dynamics of an experimental vehicle was given by Hunt *et al* [18]. Performance of the local model network was very favourable, and the models developed are currently being used as the basis of a nonlinear control design. Nonlinear control of automatic vehicle steering using regime decomposition has been reported by Hunt *et al* [19].

References

[1] T. A. Johansen and B. A. Foss, "A NARMAX model representation for adaptive control based on local models," *Modelling, Identification and Control*, vol. 13, no. 1, pp. 25–39, 1992.

[2] T. A. Johansen and B. A. Foss, "Constructing NARMAX models using ARMAX models," *Int. J. Control*, vol. 58, no. 5, pp. 1125–1153, 1993.

[3] T. A. Johansen, *Operating regime based process modeling and identification.* PhD thesis, Department of Engineering Cybernetics, Norwegian Institute of Technology, University of Trondheim, Norway, 1994.

[4] K. J. Hunt and T. A. Johansen, "Design and analysis of gain-scheduled control using local controller networks," *Int. J. Control*, 1996. To appear.

[5] A. Skeppstedt, L. Ljung, and M. Millnert, "Construction of composite models from observed data," *Int. J. Control*, vol. 55, no. 1, pp. 141–152, 1992.

[6] S. A. Billings and W. S. G. Voon, "Piecewise linear identification of non-linear systems," *Int. J. Control*, vol. 46, pp. 215–235, 1987.

[7] H. Tong, *Non-linear Time Series: A Dynamical System Approach.* Oxford University Press, 1990. Oxford Statistical Science Series 6.

[8] M. B. Priestley, *Non-linear and Non-stationary Time Series Analysis.* Academic Press, London, 1988.

[9] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 15, pp. 116–132, 1985.

[10] K. J. Hunt, R. Haas, and R. Murray-Smith, "Extending the functional equivalence of radial basis function networks and fuzzy inference systems," *Trans. IEEE on Neural Networks*, vol. 7, pp. 776–781, May 1996.

[11] K. J. Hunt, R. Haas, and M. Brown, "On the functional equivalence of fuzzy inference systems and spline-based networks," *International Journal of Neural Systems*, vol. 6, pp. 171–184, June 1995.

[12] R. Murray-Smith, *A Local Model Network Approach to Nonlinear Modelling.* PhD thesis, University of Strathclyde, Glasgow, Scotland, 1994.

[13] R. Żbikowski, K. J. Hunt, A. Dzieliński, R. Murray-Smith, and P. J. Gawthrop, "A review of advances in neural adaptive control systems," *Int. J. Neural Systems*, 1994. Submitted for publication.

[14] R. Murray-Smith and T. A. Johansen, "Local learning in local model networks," in *4th IEE Intern. Conf. on Artificial Neural Networks*, pp. 40–46, June 1995.

[15] T. A. Johansen and B. A. Foss, "ORBIT - operating regime based modeling and identification toolkit." Preprint, submitted for publication, 1996.

[16] T. A. Johansen, "ORBIT - User's guide and reference, version 1.5," Tech. Rep. STF72-A96312, SINTEF, Trondheim, Norway, 1996.

[17] R. Murray-Smith and T. A. Johansen, eds., *Multiple Model Approaches to Modelling and Control.* Taylor and Francis, London, 1996.

[18] K. J. Hunt, J. C. Kalkkuhl, H. Fritz, and T. A. Johansen, "Constructive empirical modelling of longitudinal vehicle dynamics using local model networks," *Control Engineering Practice*, vol. 4, pp. 167–178, February 1996.

[19] K. J. Hunt, R. Haas, and J. C. Kalkkuhl, "Local controller network for autonomous vehicle steering," *Control Engineering Practice*, vol. 4, pp. 1045–1051, August 1996.

DEAD-ZONE ADAPTATION VS. OVERTRAINING PHENOMENON FOR BASIS FUNCTION NETWORKS

M. Heiss
Siemens PSE NLT2 ECANSE
Gudrunstr. 11, A-1100 Vienna, Austria
http://www.siemens.at/~ecanse, E-mail: m.heiss@ieee.org

Abstract. The incorporation of dead-zones in the error signal of basis function networks avoids the networks' overtraining and guarantees the convergence of the normalized LMS-algorithm and related algorithms. The paper shows how different types of dead-zone realizations influence the mean approximation error and assure the convergence of the algorithm. For the case that a dead-zone is not realizable, we show that the basis function network converges to the L_2-optimal approximation of the updated data points, weighted with the occurrence frequency of the operating points (with exponential forgetting).
Keywords: Learning, Dead-Zone, Overtraining, Radial Basis Function Network, LMS-Algorithm.

1 Introduction

Nonlinear controllers often use memoryless nonlinear input-output maps for representing the nonlinearity of the controller. The nonlinear map is designed according to the nonlinearity of the plant. If the plant's nonlinearity is slowly time-varying due to aging effects or due to other influences, an automatic adaptation of the input-output map is desired [1, 2].

It is well known [3, 4, 5] that the weighted ($\widehat{c}_i$) sum of N basis functions $b_i(\mathbf{x})$ can be used to approximate a nonlinear mapping $y(\mathbf{x}) : \mathcal{X}^n \rightarrow \mathcal{Y}$, where $\mathcal{X}^n \subset \mathbb{R}^n$ is the application specific n-dimensional input space and $\mathcal{Y} \subset \mathbb{R}$ is the application specific output space (Fig. 1). With $\widehat{\mathbf{c}} = (\widehat{c}_1, \ldots, \widehat{c}_N)^{\mathrm{T}}$ and $\mathbf{b}(\mathbf{x}) = (b_1(\mathbf{x}), \ldots, b_N(\mathbf{x}))^{\mathrm{T}}$ the approximated mapping can be written as

$$\widehat{y}(\mathbf{x}) = \widehat{\mathbf{c}}^{\mathrm{T}}\mathbf{b}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X}^n. \tag{1}$$

A wide class of basis functions $b_i(\mathbf{x})$ can be used such that (1) is a universal approximator [6, 7, 5] (e.g. Gaussians [8, 9], B-splines [10, 11], triangular functions [12, 13], multiquadrics [14]).

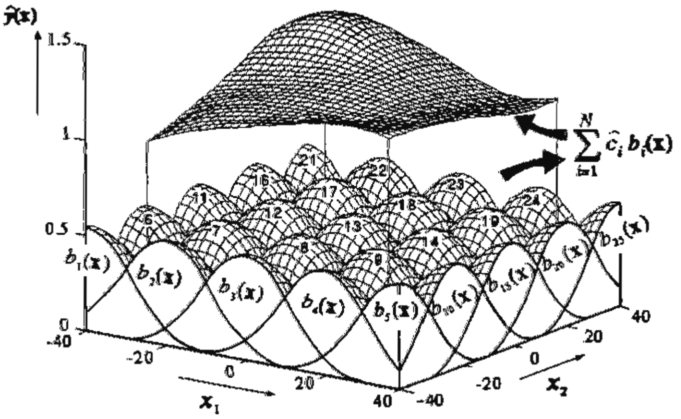


Figure 1: Basis function network approximating a two-dimensional ($n = 2$) mapping with $N = 25$ B-spline basis functions.

Nevertheless, with a fixed number N of basis functions the set of exactly representable mappings $y(\mathbf{x})$ is limited. In general, even given the best possible weights c_i^*, the mapping

$$y(\mathbf{x}) = \mathbf{c}^{*T}\mathbf{b}(\mathbf{x}) + \varepsilon(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{X}^n \tag{2}$$

is not exactly representable and a representation error $\varepsilon(\mathbf{x})$ is remaining. We assume that an upper limit

$$\varepsilon_{\max} \geq |\varepsilon(\mathbf{x})| \quad \forall \mathbf{x} \in \mathcal{X}^n \tag{3}$$

of the representation error is known. The limit $\varepsilon_{\max}$ can be interpreted as a smoothness measure of the desired mapping $y(\mathbf{x})$ if smooth basis functions are used in (1).

If the error[1]

$$e(\boldsymbol{\xi}) = y(\boldsymbol{\xi}) - \widehat{y}_{\text{old}}(\boldsymbol{\xi}) \tag{4}$$

at the actual operating point $\mathbf{x} = \boldsymbol{\xi}$ is known, then (1) can be iteratively updated by the time-discrete learning algorithm

$$\widehat{\mathbf{c}}_{\text{new}} = \widehat{\mathbf{c}}_{\text{old}} + \mu\, e_{\text{dz}}(\boldsymbol{\xi}) \frac{\mathbf{b}(\boldsymbol{\xi})}{\|\mathbf{b}(\boldsymbol{\xi})\|_2^2}, \tag{5}$$

where $0 < \mu \leq 1$ is the learning rate, $\widehat{\mathbf{c}}_{\text{old}}$ and $\widehat{\mathbf{c}}_{\text{new}}$ are the estimated weights before and after the update, and e_{dz} is the error e after applying a dead-zone (see Section 2 Fig. 2). The algorithm is called the *normalized LMS-algorithm* [11]. The denominator term $\|\mathbf{b}(\boldsymbol{\xi})\|^2$ causes the error $y(\boldsymbol{\xi}) - \widehat{y}_{\text{new}}(\boldsymbol{\xi})$ at the location $\boldsymbol{\xi}$ of the update to be zero after the update (a-posteriori error). The learning algorithm (5) can also be used without the denominator term if a reduction in the error is sufficient. In this case the algorithm is called "LMS-algorithm" in the context of adaptive filtering [15, 16], "back-propagation algorithm" in the context of neural networks [17], or simply "gradient descent method" [11].

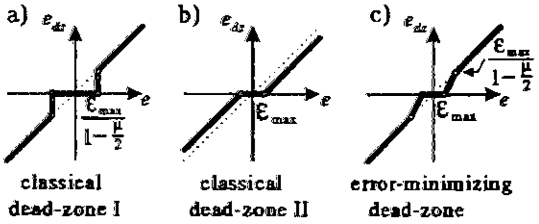
In the following section we show how the characteristic of the required dead-zone influences the convergence and the accuracy of the normalized LMS-algorithm. The dead-zones and the convergence analyses in this paper are also applicable to the LMS-algorithm, if $\mu\|\mathbf{b}(\boldsymbol{\xi})\|^2$ is used instead of μ in all equations.

2 Choice of Dead-Zone Type Influences the Final Approximation Error

It is shown in [18] that every dead-zone fulfilling the convergence condition

$$\left.\begin{array}{l} |e_{\text{dz}}| \leq \frac{2}{\mu}[e(\boldsymbol{\xi}) - \varepsilon(\boldsymbol{\xi})]\,\text{sign}(e(\boldsymbol{\xi})) \\ \text{or} \\ e_{\text{dz}} = 0 \end{array}\right\} \forall e \in \mathbb{R}, \forall \varepsilon(\boldsymbol{\xi}) \leq \varepsilon_{\max}. \tag{6}$$

guarantees the convergence of the basis function network (i.e. a decreasing parameter error $\|\mathbf{c}^* - \widehat{\mathbf{c}}\|_2^2$). If no dead-zone is incorporated ($e_{\text{dz}} = e$), condition (6) is not satisfied and therefore the convergence is not guaranteed (see Section 3). The classical dead-zone I (Fig. 2a), the classical dead-zone II (Fig.2b), and the error-minimizing dead-zone (Fig. 2c) [18] satisfy condition (6).


a)
e_{dz}
e
$\frac{\varepsilon_{\max}}{1-\frac{\mu}{2}}$
b)
e_{dz}
e
$\varepsilon_{\max}$
c)
e_{dz}
e
$\frac{\varepsilon_{\max}}{1-\frac{\mu}{2}}$
$\varepsilon_{\max}$

classical dead-zone I

classical dead-zone II

error-minimizing dead-zone

Figure 2: Comparison of three different dead-zone characteristics (dotted line: $e_{\text{dz}} = e$).

[1] For clarity the indices "old" and "new" are used instead of the usual k and $k+1$.

With $\mu = 1$ the maximal final error is $2\varepsilon_{\max}$ for the classical dead-zone I, and $\varepsilon_{\max}$ for both the classical dead-zone II and the error-minimizing dead-zone. The average error for typical examples is 5% less for the classical dead-zone II than for the classical dead-zone I and even 30% less for the error-minimizing dead-zone than for the classical dead-zone I [18].

3 Overtraining Phenomenon

The dead-zones of the previous section are not realizable if the exact value of the error $e(\boldsymbol{\xi})$ is not known but only some unknown or some non-invertible function $f(e)$ of the error is available. We will analyze in this section whether the LMS algorithm is still reasonably applicable even if the convergence is not guaranteed. For this purpose let us have a different view on the LMS algorithm.

It was shown by [16, 13] that the simple learning algorithm (5) is equivalent to computing the new parameters

$$\widehat{\mathbf{c}}_{\text{new}} = \arg \min_{\widehat{\mathbf{c}} \in \mathbb{R}^n} \left(\frac{\mu}{\|\mathbf{b}(\xi)\|_2^2} \underbrace{(y(\boldsymbol{\xi}) - \widehat{\mathbf{c}}^T \mathbf{b}(\boldsymbol{\xi}))}_{\substack{\text{error after adaptation at} \\ \text{actual operating point}}}{}^2 + (1-\mu) \underbrace{\|\widehat{\mathbf{c}} - \widehat{\mathbf{c}}_{\text{old}}\|_2^2}_{\substack{\text{change of the} \\ \text{parameter vector}}} \right) \tag{7}$$

by minimizing[2] a combination of the local a-posteriori error and the parameter change.

If the learning rate is large, $\mu \approx 1$, then the first term of Eq. (7) is dominant and therefore the local error at the operating point $\boldsymbol{\xi}$ is minimized even if this would cause a large change of the already learned parameters (second term of Eq. (7)).

If the learning rate is very small, $\mu \ll 1$, then the second term of Eq. (7) works like a first order filter which sums up (with exponential forgetting) all the local errors (first term of Eq. (7)). Thus, not only the local squared a-posteriori error $(y(\boldsymbol{\xi}) - \widehat{\mathbf{c}}_{\text{new}}^T \mathbf{b}(\boldsymbol{\xi}))^2$ is minimized but (after several updates at different operating points $\boldsymbol{\xi}$) the global least squares error is minimized. Note that the second term of Eq. (7) sums up the squared errors at each operating point $\boldsymbol{\xi}$. Consequently, each squared error is weighted with the occurrence frequency of its operating point $\boldsymbol{\xi}$.

We can therefore conclude as a rule of thumb:
The basis function network converges within the local neighborhood of the operating points to the operating point occurrence frequency weighted L_2-optimal approximation of the updated points. Additionally, we must take into account that older updates have less weight for the optimization than more recent updates have (approximately exponential forgetting).

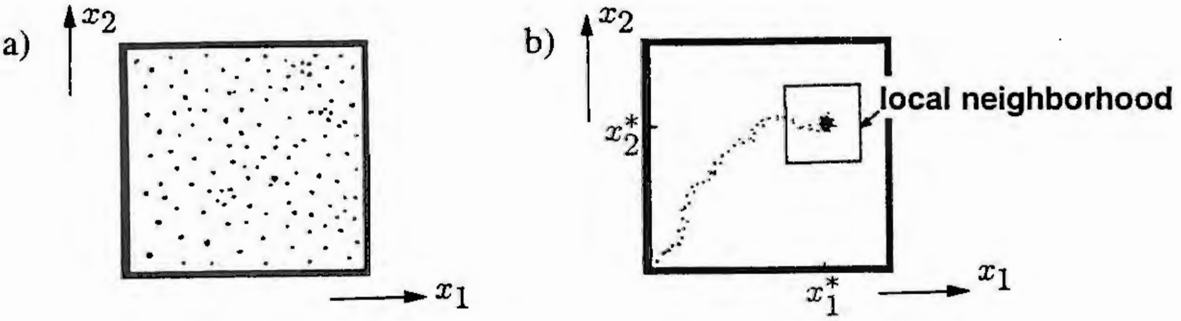


Figure 3: a) evenly distributed operating points (dots)
b) typical undesirable operating point distribution in practical applications.

This behavior is very desirable as long as the operating points $\boldsymbol{\xi}$ are evenly distributed over the whole input space $\mathcal{X}^n$ (Fig. 3a). In most applications the operating points are not evenly distributed and in

[2] Setting the partial derivative $\frac{\partial}{\partial \widehat{\mathbf{c}}}$ of the cost function in Eq. (7) equal to the zero vector, yields Eq. (5).

these cases the algorithm's behavior (without dead-zone) is undesirable as the so called "overtraining phenomenon"[19] occurs.

A simplified example illustrates the overtraining phenomenon: Consider to learn some engine characteristic during a long car ride on a straight flat freeway. Let the speed of the car be controlled by a cruise control. After some initial acceleration the car's engine will stay at some operating point $\xi = (x_1^*, x_2^*)$ for millions and millions of updates (Fig. 3b). We would expect that this particular operating point and its neighborhood are perfectly learned after the ride. As the number of updates at the few initial operating points is vanishing small compared to the millions of updates at the single operating point (x_1^*, x_2^*), the weights of all other points can be assumed to be zero except the single operating point (x_1^*, x_2^*) which is weighted with one. The weighted L_2-optimum is therefore the tangent plane at this single operating point (x_1^*, x_2^*). Thus, the basis function network converges within the local neighborhood to this tangent plane and consequently may destroy a part of the already learned engine characteristic.

4 Conclusion

If the learning algorithm (5) is applied with a dead-zone type satisfying (6) (see Fig. 2 for examples), then the convergence is guaranteed and no overtraining phenomenon occurs. If no dead-zone is realizable then the network converges to the operating point occurrence frequency weighted L_2-optimum (with approximately exponential forgetting).

In the latter case, the convergence behavior depends on the distribution of the operating points and on the curvature of the desired input-output map. As long as the curvature is small, the distortion due to the overtraining effect will be considerably small. If the curvature is large, then the overtraining effect (in other words: the distortion) is substantial and the incorporation of a dead-zone would be recommended in order to avoid the overtraining effect.

Acknowledgement

This work was supported by the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung" under contract P8914-TEC during the author's stay at the Vienna University of Technology. The author is grateful to Prof. F. Paschke, Prof. A. Weinmann and Dr. G. Brasseur for their support.

References

[1] M. Schmitt, *Untersuchungen zur Realisierung mehrdimensionaler lernfähiger Kennfelder in Großserien-Steuergeräten*, VDI-Verlag, Reihe 12, Nr. 246, Düsseldorf, 1995.

[2] H. Tolle and E. Ersü, *Neurocontrol*, Springer-Verlag, Berlin, 1992.

[3] M. J. D. Powell, "The theory of radial basis functions for multivariable approximation in 1990", in *Advances in Numerical Analysis II*, W. Light, Ed. Oxford University Press, Oxford, 1992.

[4] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks", *Science*, vol. 247, pp. 978–982, February 1990.

[5] F. Girosi and T. Poggio, "Networks and the best approximation property", *Biol. Cybernetics*, vol. 63, pp. 169–176, 1990.

[6] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks", *Neural Computation*, vol. 3, pp. 246–257, 1991.

[7] J. Park and I. W. Sandberg, "Approximation and radial-basis-function networks", *Neural Computation*, vol. 5, pp. 305–316, 1993.

[8] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control", *IEEE Trans. on Neural Networks*, vol. 3, no. 6, pp. 837–863, November 1992.

[9] S. Kampl and M. Heiss, "Multiplication-free radial basis function network", in *American Control Conference (ACC 95)*, Seattle, 1995, IEEE, pp. 3782–3785.

[10] S. H. Lane, D. A. Handelman, and J. J. Gelfand, "Theory and development of higher-order CMAC neural networks", *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 23–30, April 1992.

[11] M. Brown and Ch. Harris, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, New York, 1994.

[12] R. M. Sanner and J.-J. E. Slotine, "Stable recursive identification using radial basis function networks", in *American Control Conference (ACC 92)*, Chicago, 1992, IEEE, pp. 1829–1833.

[13] M. Heiss, D. Heiss, and S. Kampl, "Learning of linearly interpolated input-ouput maps (in German)", *Automatisierungstechnik at*, vol. 42, no. 11, pp. 497–506, 1994.

[14] M. Pottmann and D. E. Seborg, "Identification of non-linear processes using reciprocal multiquadric functions", *J. Process Control*, vol. 2, no. 4, pp. 189–203, 1992.

[15] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, New Jersey, 1985.

[16] G. Kubin, "Joint recursive optimality – a non-probabilistic approach to adaptive transversal filter design", *Computers Elect. Engng*, vol. 18, no. 3/4, pp. 277–289, 1992.

[17] D. E. Rumelhart et al., *Parallel Distributed Processing*, vol. 1, MIT-Press, Boston, 1986.

[18] M. Heiss, "Error-minimizing dead-zone for basis function networks", *IEEE Transactions on Neural Networks*, vol. 7, November 1996.

[19] J.-J. E. Slotine and R. M. Sanner, "Neural networks for adaptive control and recursive identification: A theoretical framework", in *Essays on Control: Perspectives in the Theory and its Application*, H. L. Trentelman and J. C. Willems, Eds., chapter 11, pp. 381–436. Birkhauser, Boston, 1993.

A Simulated Annealing Algorithm for Approximating Triangulated Surfaces*

(Extended Abstract)

Martin Kropp[1], **Peter R. Remmele**[2], **and Thomas Roos**[2]

[1]R&D Industrial Measurement Systems, Leica AG
CH-5035 Unterentfelden, Switzerland
Fax: +41 62 723 07 34. Email: krma@pmu.leica.ch

[2]Departement für Informatik
ETH Zentrum, CH-8092 Zürich, Switzerland
Fax: +41 1 632 11 72. Email: {remmele, roos}@inf.ethz.ch

Abstract

Given a set P of n points $p_i = (x_i, y_i, f(x_i, y_i))$ on a surface in $\mathbb{R}^3$. Let f_P be the triangulated surface of P whose projection onto the xy-plane is the Delaunay triangulation of the projected points. Our goal is to approximate f_P by a simplified triangulated surface f_A with $A \subset P$ and with respect to two approximation measures: the quality of the approximation measuring the maximum vertical distance between f_A and f_P and the cardinality $|A|$ of the approximating set. We present a simulated annealing approach based on a fully dynamic Delaunay triangulation. Extensive tests with real data sets from CAD and GIS show that our algorithm is both practical and efficient.

1 Introduction

In many areas of science, real-world surface data is represented by surface models, as e.g. the digital elevation model (DEM) and the triangulated irregular network (TIN). Geographic information systems (GIS) use triangle-based terrain models for simulation, visualization, and analysis of terrain data [22]. Triangulated surfaces can also be found in such diverse fields like finite element methods (FEM), medical imaging, computer aided design (CAD), and computer animation. Larger ranges of interest paired with the need of greater accuracy is the reason for the large amount of data encountered by today's applications. This indicates the need for approximate representations of lower complexity to speed up the computation or to allow even real-time visualization [3, 9, 10]. Multiresolution techniques use triangulations from a hierarchy of approximations of different resolutions depending on the distance from the view point [5, 22, 24].

The dilemma of approximation algorithms is that on the one hand they should simplify the surface by minimizing the amount of data used for its representation while on the other hand, the quality of approximation and the appearance of the surface should be maintained [13]; in GIS this means that specific features such as peaks, rivers, ridges, etc. should be preserved. Thus, an approximation algorithm has to combine the simplification of the surface by reducing the amount of data with an error measure that preserves surface features somehow. From a theoretical point of view, Agarwal and Suri [1] proved the general abstract problem of finding a linear approximation of a bivariate function $f(x, y)$ within a given error bound and minimum complexity to be $\mathcal{NP}$-hard. This is the reason why many approaches to the described problem use (interactive) heuristics to achieve reasonable results [11, 24].

In this paper we investigate the approximation of triangulated surfaces with respect to a given approximation error *and* a desired complexity of the approximated surface. We present a simulated annealing algorithm based on a fully dynamic Delaunay triangulation that allows to simultaneously take both requirements into account.

*The authors gratefully acknowledge the partial support by the Swiss National Science Foundation (SNF) under grant 21-45610.95 and 5003-45360/1.

2 The Algorithm

We are given a set P of n points $p_i = (x_i, y_i, f(x_i, y_i))$ lying on the surface of a bivariate function $f : (x, y) \mapsto f(x, y)$. Let f_P be the triangulated surface of P whose orthogonal projection onto the xy-plane is the *Delaunay triangulation* [23] of the projected points P'. Thereby, the Delaunay triangulation is defined as the triangulation of P' consisting of triangles that contain no point of P' in the interior of their circumcircle. It is well known that the Delaunay triangulation is unique as long as the points of P' are in *general position* (i.e. no four points of P' lie on a circle and no three points on a line) and that it maximizes the minimum angle among all triangulations of P' [12] which, in general, avoids the occurrence of very thin (sliver) triangles. The Delaunay triangulation of a point set in non-general position, e.g. on a grid, is not unique; however, this difficulty can be overcome by enumerating the points in their order of input.

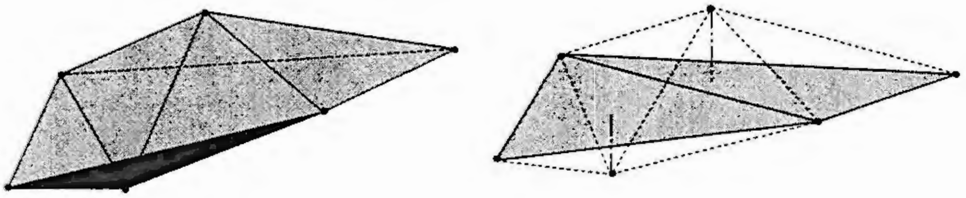


Figure 1: An example of an approximation.

We call f_A an *approximation* of f_P iff $A \subset P$ (see Figure 1). To measure the quality of an approximation, we introduce a *cost measure* c of A with respect to P. This allows us to define an *optimal approximation* f_P^{ε} of f_P as an approximation f_A that minimizes the measure c. In order to design a useful measure c we take two criteria into account: the cardinality $|A|$ of the approximating set and the similarity of f_A with respect to f_P. In our approach, the latter is defined as the maximum pointwise vertical distance ε_A of the surface f_A with respect to f_P.

Fully dynamic Delaunay triangulation

The fast generation of new approximations f_A requires the maintenance of the Delaunay triangulation $DT(A')$ (again A' is the projection of A) with respect to insertions and deletions of points. This goal can be achieved using *fully dynamic* Delaunay triangulation methods. We can efficiently support these operations using the *quad-edge* data structure by Guibas and Stolfi [15].

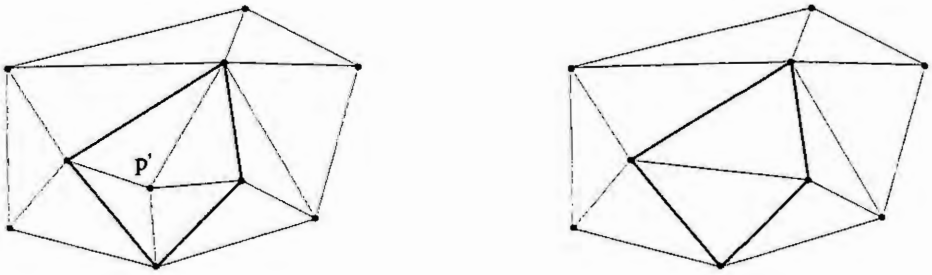


Figure 2: Deleting and adding a point p'.

The initial Delaunay triangulation $DT(P')$ can be computed in $O(n \log n)$ time, e.g. by a *randomized incremental* algorithm [14] or a *divide-and-conquer* approach [15]. The latter work also gives an $O(k)$ time algorithm for the *insertion* of a single point p' into an existing Delaunay triangulation $DT(A')$, where k is the number of Delaunay neighbors of p' in the new triangulation $DT(A' \cup \{p'\})$. In order to locate a point p' in $DT(A')$ in constant time we maintain for each point $q' \in P' - A'$ a reference to the covering triangle of the current triangulation $DT(A')$. A right-to-left transition in Figure 2 shows the insertion of the point p'.

Deleting a point p' from the triangulation $DT(A')$ creates a polygonal star-shaped hole which has to be re-triangulated in order to re-establish the correct Delaunay triangulation $DT(A' - \{p'\})$. Therefore, the time complexity of the deletion is determined by the time complexity of the re-triangulation of the Delaunay neighbors of p' in $DT(A')$; this can be done in $O(k)$ time [2], where k is again the number of Delaunay neighbors of p' in $DT(A')$. A left-to-right transition in Figure 2 shows the deletion of the point p'. Notice that the *average degree* of a point in a Delaunay triangulation is about six [23] (due to the planarity of the triangulation). This implies that both the insertion and the deletion of a point p' can be performed in constant time in the average case.

We can efficiently maintain the maximum vertical distance ε_A of an approximation f_A with respect to f_P by a dynamic update of ε_A whenever A changes by an insertion or a deletion of a point p. We do this by updating the vertical distances of the affected points in $P - A$; for this, for each point $q' \in P' - A'$ a pointer to the currently covering triangle of $DT(A')$ is maintained. By the way, this also allows to locate a point p' in the Delaunay triangulation $DT(A')$ in constant time. It is not too hard to see that the average update time is constant as long as $|A|$ is proportional to $|P|$.

Simulated Annealing

Simulated annealing (SA) is a technique which can be used to solve hard combinatorial or optimization problems [4, 6, 17]. This fact motivated us to apply the SA framework to our problem. Simulated annealing has its origin in physics and simulates the annealing of a heated metal, i.e. the transition to a state of minimum energy (thermal equilibrium). In our setting, we want to find an optimal approximation f_P^c among all feasible approximations f_A. The *configuration space* of the SA algorithm is therefore the set of all subsets $A \subset P$. The feasible *neighbor configurations* of a configuration A are the ones which can be created by adding a point of $P - A$ to A or removing a point from A. Notice that both operations can be efficiently supported using the Delaunay triangulation – as we have seen before.

The SA algorithm simulates the physical process of annealing by decreasing a given starting temperature, by means of a *temperature function* $T : \mathbb{N} \to \mathbb{R}_{\geq 0}$, stepwise and slowly towards zero. For each level of temperature, the algorithm generates random transitions to neighbor configurations, until a global *stop criterion* is reached. Thereby the probability of a transition from configuration A to a neighbor configuration B is given by the following expression:

$$\mathrm{Prob}[A \to B] \;:=\; \frac{1}{n} \cdot e^{\min\{0, c(A) - c(B)\}/T}$$

This formula expresses the ability to escape from a local minimum with some small probability that decreases with falling temperature T. The temperature function T is crucial to the SA algorithm; if T decreases too fast, the algorithm can get stuck in a local minimum. More details concerning the general SA framework can be found in the standard texts [7, 16, 19, 20, 21].

As we have seen, two different optimality criteria come to mind (see also [8]): for a given *maximal vertical tolerance* $\varepsilon > 0$, an optimal configuration minimizes the cardinality $|A|$ among all configurations A with approximation error $\varepsilon_A \leq \varepsilon$. On the other hand, if the cardinality m is specified, the optimal configuration A is the one which minimizes the approximation error ε_A among all configurations A with $|A| \leq m$. So, we have two competing quality measures: reducing the cardinality $|A|$ will increase the approximation error ε_A, and vice versa. We combine both criteria to the desired *cost measure*:

$$c(A) \;:=\; \alpha \cdot e^{p_\alpha \frac{|\varepsilon_A - \varepsilon|}{\varepsilon}} + \beta \cdot e^{p_\beta \frac{||A| - m|}{|P|}}$$

The first term expresses the costs resulting from the approximation error of the approximation f_A while the second term reflects the costs of the cardinality of A. The parameters α and β allow the weighting of the optimization goals. Setting $\alpha = 0$, we optimize for cardinality only, while $\beta = 0$ focuses on the approximation error. The parameters p_α and p_β are used as *penalty factors*.

For each temperature level, the SA algorithm randomly generates a series of configurations. After the system has *almost* reached thermal equilibrium, we decrease the temperature and generate the next series of configurations. This process is repeated until a global *stop criterion* is reached. We use two different stop criteria: a fixed number of temperature levels or a maximal cost difference of a few recent configurations. As a single configuration transition (insertion or deletion) can be performed in constant time in the average case, the time complexity of our SA algorithm is mainly determined by the number of temperature levels and the number of repetitions per temperature level.

3 Implementation / Test

The SA algorithm has been implemented on a two-processor SUN SPARCstation 10 using (Gnu) C++. We extensively tested the SA algorithm with real data sets of different sizes from GIS and CAD: laser tracker data of a turbine (771 points), data generated by a bivariate function (4000 points), laser tracker data of a car door (5133 points), and data of a digital terrain model from photogrammetry (32086 points) were used as test data sets (for details see [18]).

As described above, the cost measure c can be parametrized by its parameters α, β, p_α, and p_β which allow to direct the optimization towards the maximal approximation error ε or the maximal cardinality m. Table 1 gives an extract of the tested parameter settings.

Name	α	β	p_α	p_β	Optimization for
K1	1	1	5	5	Error and cardinality (equal weight)
K2	1	0	5	5	Error only
K3	0	1	5	5	Cardinality only
K6	1	1	4	2	Error and cardinality (double the weight)

Table 1: Parameter settings of the cost function.

We tested different SA algorithms: the SA1 algorithm uses a fast decreasing temperature function in combination with many configuration transitions per temperature level while the SA2 algorithm uses a slowly decreasing temperature function executing only a few configuration transitions per temperature level. These two SA variants also differ in their stop criteria: while SA1 uses a maximal cost difference between a few recent configurations, SA2 computes a fixed number of iterations. The tests showed that the setting of the maximal cost difference is crucial to the running time of the SA1 algorithm.

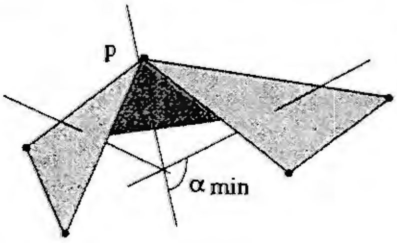


Figure 3: Planarity calculation for point p.

The SA algorithms were compared with a *deterministic* algorithm whose point elimination scheme is based on the *planarity* of each point. Thereby, the planarity of a point p is defined as the minimum angle in space formed by any two adjacent triangles that are incident to p. Figure 3 illustrates the construction of the planarity for the point p. This deterministic algorithm incrementally maintains a list of points sorted according to their planarity. The algorithm continues to remove the point with maximum planarity until the given maximal approximation error ε is reached.

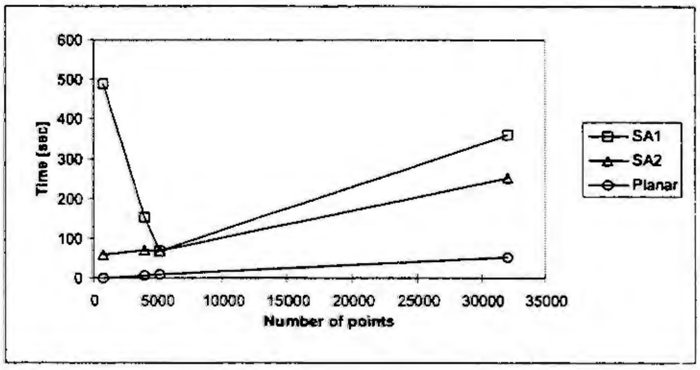


Figure 4: Comparison of the running time.

Figure 4 compares the running time of the algorithms as a function of the size of the data sets using the setting K1 of Table 1. Comparing the approximation quality of the algorithms, we observe that the two SA algorithms achieve much better results than their deterministic counterpart. Figure 5 shows the obtained reduction of cardinality of P (with a desired reduction of 50 %) and the deviation from the given maximal approximation error; thereby, the deviation is given as the fraction between the obtained approximation error and the given maximal approximation error; thus a factor of 1 means that the maximal approximation error was exactly matched. Figure 5 also clearly shows that the deterministic algorithm did not reach the given optimization goals, while the SA algorithms came very close.

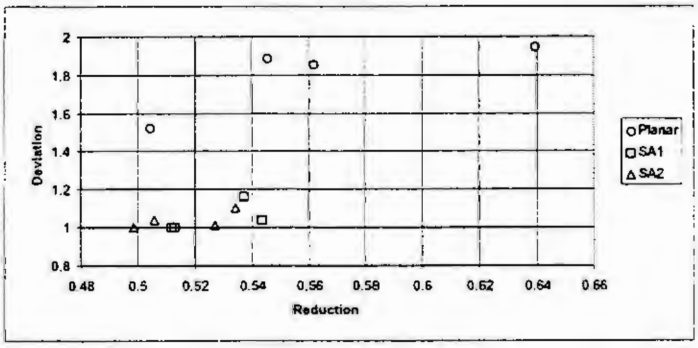


Figure 5: Error deviation versus reduction.

Both SA algorithms produced better results for equally weighted optimization criteria than for one criterion alone. In some cases doubling the weight of the cardinality criterion dramatically reduced the running time. An interesting observation is the difference of spatial distribution of the final point arrangements resulting from the different point-elimination schemes. While the deterministic algorithm (Figure 6, right) is thinning all planar areas of a given surface, the SA algorithms (Figure 6, left) tend to eliminate points equally distributed. Notice that both point arrangements in Figure 6 have the same cardinality of about 50 % of the original set.

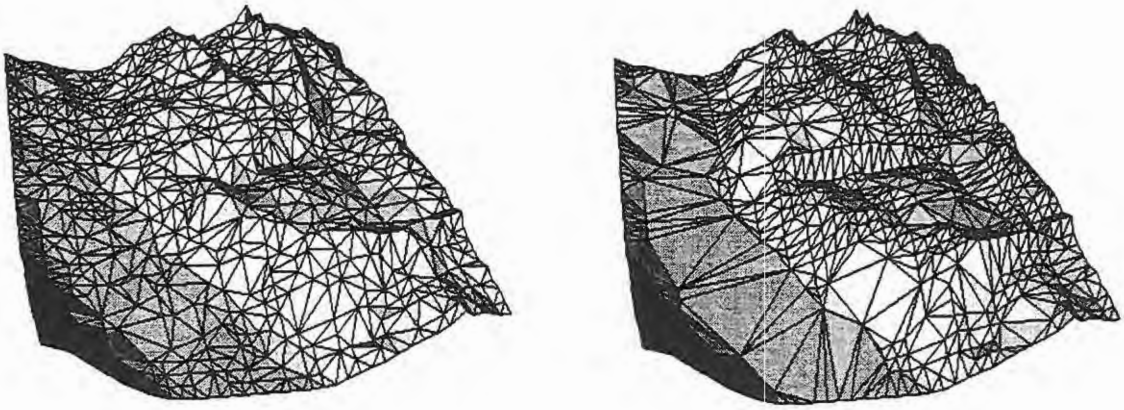


Figure 6: Approximation of a terrain.

Remarks and Future Work

In this paper, we presented a simulated annealing approach to the problem of approximating a triangulated surface. We used a fully dynamic Delaunay triangulation to support the (random) transitions of the SA algorithm. Our approach turned out to be both practical and efficient. We are currently applying the algorithm to the *Rimini* digital elevation model of Switzerland that covers Switzerland by a $250m \times 250m$ grid with approximately 500.000 elevation points. In this context we are grateful to the Bundesamt für Landestopographie for providing us the data.

References

[1] P. Agarwal and S. Suri, *Surface Approximation and Geometric Partitions*, Proc. 5th ACM-SIAM SODA, pp. 24–33, 1994

[2] A. Aggarwal, L.J. Guibas, J. Saxe, and P.W. Shor, *A Linear Time Algorithm for Computing the Voronoi Diagram of a Convex Polygon*, Proc. 19th ACM-STOC, pp. 39–45, 1987

[3] J. Bloomenthal, *Polygonalization of Implicit Surfaces*, Computer Aided Geom. Design, Vol. 5, No. 4, pp. 341–355, 1988

[4] E. Bonomi and J.L. Lutton, *The N-City Traveling Salesman Problem*, Statistical Mechanics and the Metropolis Algorithm, 1984

[5] P. Borrel et al., *Multiresolution 3D Approximations for Rendering Complex Scenes*, Modeling in Computer Graphics: Methods and Applications, B. Falcidieno and T. Kunii (Eds.), Springer, 1993

[6] V. Černy, *Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm*, J. Opt. Theory Appl., Vol. 45, pp. 41–51, 1985

[7] N.E. Collins et. al., *Simulated Annealing - An Annotated Bibliography*, American Journal of Mathematical and Management Sciences, pp. 209–307, 1988

[8] L. De Floriani et. al., *Hierarchical Structure For Surface Approximation*, Computer and Graphics, Vol. 8, No. 2, pp. 183–193, 1984

[9] M. DeHaemer and M. Zyda, *Simplification of Objects Rendered by Polygonal Approximations*, Computer and Graphics, No. 15, pp. 175–184, 1992

[10] H. Delingette, *Simplex Meshes: A General Representation for 3D Shape Reconstruction*, Proc. Int. Conf. on Computer Vision and Pattern Recognition, pp. 856–859, 1994

[11] D.H. Douglas, *Experiments to Local Ridges and Channels to Create a New Type of Elevation Model*, Cartographica, No. 23, pp. 29–61, 1986

[12] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer, 1987

[13] B. Falcidieno and C. Pienovi, *A Feature-Based Approach to Terrain Surface Approximation*, Proc. 4th Symp. on Spatial Data Handling, pp. 190–199, 1990

[14] L. Guibas, D.E. Knuth, and M. Sharir, *Randomized Incremental Construction of Delaunay and Voronoi Diagrams*, Proc. 17th Intern. Colloquium on Automata, Languages and Programming ICALP'90, LNCS 443, Springer, pp. 414–431, 1990

[15] L.J. Guibas and J. Stolfi, *Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams*, ACM Transactions on Graphics, Vol. 4, pp. 74–123, 1985

[16] B. Hajek, *Cooling Schedules for Optimal Annealing*, Mathematics of Operations Research, Vol. 13, No. 2, pp. 311–329, 1988

[17] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, *Optimization by Simulated Annealing*, IBM Research Report RC 9355, 1982

[18] M. Kropp, *Oberflächenapproximation mittels Triangulation und deren Anwendungen*, Diploma Thesis, FU Hagen and ETH Zürich, 1996

[19] P.J.M. van Laarhoven, *Theoretical and Computational Aspects of Simulated Annealing*, D. Reidel Publishing Company, 1988

[20] P.J.M. van Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications*, D. Reidel Publishing Company, 1987

[21] S. Nahar et al., *Experiments with Simulated Annealing*, 1985

[22] R. Pajarola, P. Stucki, K. Szabo, and P. Widmayer *ViRGIS - Virtual Reality und Geographische Informations-Systeme*, Bulletin des Schweizerischen Elektrotechnischen Vereins, No. 17, pp. 25–28, 1996

[23] F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction*, Springer, 2nd Ed., 1988

[24] L. Scarlatos and T. Pavlidis, *Hierarchical Triangulation Using Cartographic Coherence*, CVGIP: Graphical Models and Image Processing, Vol. 54, No. 2, pp. 147–161, 1992

Modelling Nonlinear Characteristics using Hierarchical Delaunay-Networks

T. Ullrich and H. Tolle
Darmstadt University of Technology, Landgraf-Georg-Strasse 4, D-64283 Darmstadt
E-Mail: thul@rt.e-technik.th-darmstadt.de

Abstract

Real-time control of nonlinear plants requires efficient methods to approximate nonlinear characteristics such as process models or control strategies. Artificial Neural networks can be applied to represent such characteristics but often this approach does not meet constraints of limited computational resources. Automotive control systems are an example of application areas, where real-time processing based on relatively simple hardware is necessary. Delaunay-Networks are an efficient means for real-time function approximation. They consist of a number of interpolation nodes, the Delaunay triangulation of which is used to define finite elements in the input space. Within each finite element, a linear model of the represented function is applied. This approach yields short network response times but a relatively large storage capacity is required for the representation of the triangulation. This is particularly true if more than two variables comprise the network's input space. The paper at hand introduces the concept of *Hierarchical Delaunay-Networks* (HDN), an approach that allows for memory expense to be traded-off against response time. A simplified engine torque model is used to illustrate this method.

1 Introduction

When analytical modelling of nonlinear plants is intractable or too time-consuming, systems that *learn* from data are a promising approach. Artificial Neural Networks (ANN) are generally applicable to this kind of problems, however, in application areas where real-time operation on relatively simple hardware is necessary, the implementation of ANNs often is infeasible. *Interpolation networks* which are based on mathematical principles will be shown to be an alternative.

Interpolation networks consist of a set of *interpolation nodes* which are connected to define *finite elements* in the input space. Within each finite element, a simple, e.g. a *linear* model of the represented function is applied. Typically, lattice-like networks with hypercubes as finite elements are used [1]. However, the lattice-structure does not allow to adjust the distribution of the nodes to the local complexity of the underlying function. Thus, these models do not comply with the *principle of parsimony* which demands that the model should represent a given set of data with the fewest possible parameters (nodes). Parsimonious models are computationally inexpensive and yield a high generalisation quality. Overparameterised models, on the other hand, are prone to overfitting, i.e. to modelling measurement noise or the peculiarities of the given samples.

One approach to parsimonious interpolation networks is to allow the nodes to be arbitrarily distributed. The node density can then be adjusted to the local complexity of the represented function. Instead of hypercubes, it is then appropriate to use *simplices* as finite elements. A simplex in the n-dimensional input space is defined by $n+1$ nodes each of which comprises a *position* in the input space and an *attribute*, i.e. an estimate of the approximated function's value at the node position. Thus, simplices allow the network response to be computed by local *linear* interpolation.

Queries to a simplex-based interpolation network are processed in a two step procedure. At first, given a *query point* $\mathbf{x}$ the simplex which contains $\mathbf{x}$ is selected (see figure 1 left). The vertices of that simplex are called *active nodes*. These $n+1$ active nodes define a hyperplane which is used to compute the network response (figure 1 right).

The rest of this paper is organized as follows. In section 2 simplex-based networks with simplices obtained form the *Delaunay triangulation* of the nodes are introduced. These *Delaunay-Networks* (DN) have some desirable properties for function approximation. Further information on this approach can be found in [2] and [3]. Section 3 discusses the further development of this method to scalable *Hierarchical Delaunay-Networks* (HDN). The basic idea of HDN is to memorize the Delaunay triangulation of a subset of the nodes only. Queries to the network are processed by first selecting the simplex of the stored triangulation that contains the current query point (1st hierarchical level). Then a local reconstruction of the Delaunay triangulation is carried out in the vicinity of the current query point and a simplex for output interpolation is selected (2nd hierarchical level). This is achieved by applying *topological transitions* [4] and *containment tests* in an iterative manner. HDNs are *scalable*, i.e. their memory

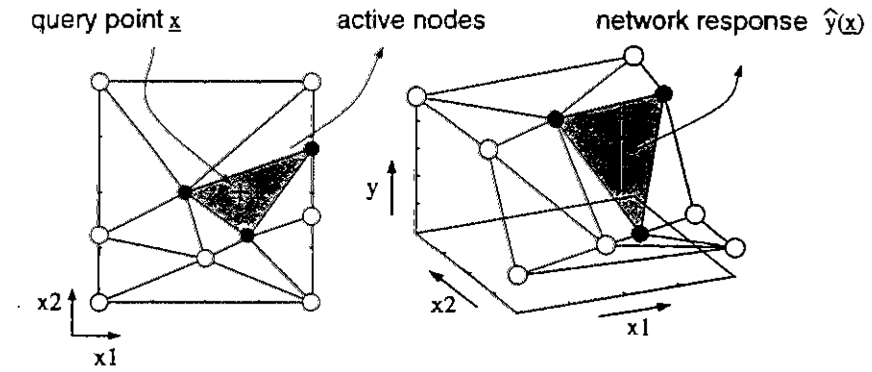

query point $\underline{x}$
active nodes
network response $\hat{y}(\underline{x})$
y
x2
x1
x2
x1

Figure 1: A simplex-based interpolation network with 2D input space.

requirements and response time can be adapted to comply with the limits of a given application:

1. **Trade-off between storage capacity and response time:** The higher the number of nodes that are included in the memorized triangulation is, the lesser topological transitions need to be carried out in the second level, yielding faster network response and higher memory expense.

2. **Trade-off between maximum response time and model accuracy:** The maximum response time can be guaranteed not to exceed a certain limit by prescribing a maximum number of topological transitions *per query* . This means that the network response can be computed from a non-Delaunay simplex in some situations and this leads to a loss of model accuracy temporarily.

In section 4 the HDN approach is illustrated by means of a simplified engine torque model. In this example, the input space comprises only two variables, but the method can also be applied to multidimensional modelling problems. A three-dimensional example will be discussed at the conference.

2 Delaunay Networks

Given a set of N nodes arbitrarily distributed in the n-dimensional input space, the definition of a set of simplices connecting $n+1$ nodes respectively, is not unique. At maximum, it is possible to construct as many as $s_{max} = \binom{N}{n+1}$ simplices. However, those simplices used in an interpolation network have to satisfy two constraints. Firstly, they must be non-intersecting to avoid ambiguity of the network response. Moreover, the union of all simplices must cover the network's entire input space in order to avoid *extrapolation*. This is desirable because extrapolation generally leads to unreliable responses and the implementation of extrapolation routines involves computational overhead.

A certain type of simplices which are particularly useful for function approximation are *Delaunay simplices* [5]. They are characterized by the empty-circle property:

Definition 2.1 *A simplex T_i consisting of $n+1$ nodes in $\mathbb{R}^n$ is a Delaunay simplex if and only if its embedding n-dimensional hypersphere does not contain any other node of the network.*

Figure 2 (left) illustrates this definition for a set of four nodes in $\mathbb{R}^2$. The circumcircles of the triangles in part a contain the fourth node, respectively[1]. The triangles in part b, however, posess the empty circle property and define the *Delaunay triangulation* of the given set of nodes.

Delaunay simplices can be shown to satisfy the two constraints; they are non-intersecting and cover the entire input domain provided that the node distribution is properly chosen[2]. Moreover, Delaunay simplices minimize the worst case approximation error that arises from the local linear modelling. This property is proven in [6], and can be motivated intuitively by means of a one-dimensional example: If a function $\hat{y}(x)$ with bounded second derivative $\hat{y}'' \leq 2 \cdot c$ is to be approximated by a linear model on a finite interval $[a, b]$, the hardest modelling problem is to represent a quadratic function $\hat{y} = c \cdot x^2 + b \cdot x + a$,

[1] In the 2D case, simplices are triangles and their embedding hyperspheres reduce to circumcircles.

[2] The convex hull of the set of nodes must cover the entire input domain and the nodes have to be in general position.

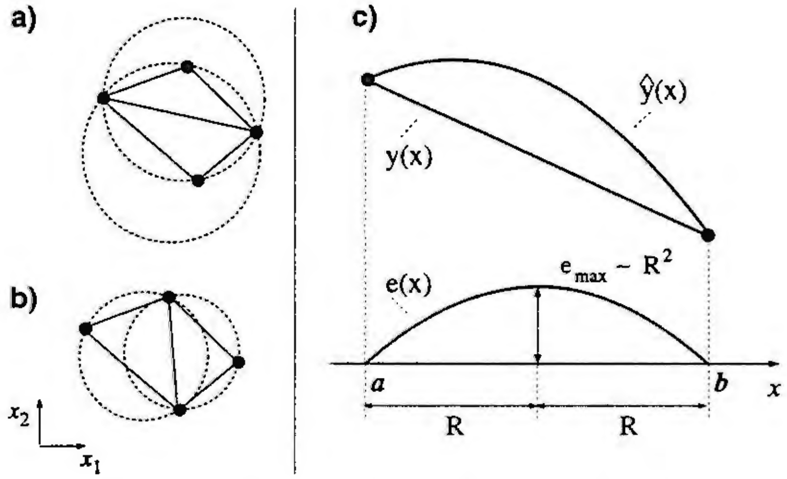


Figure 2: **a)** and **b)** The Delaunay definition. **c)** Minimization of the worst case approximation error.

since the second derivative is then equal to the maximum of $2 \cdot c$ on the entire interval. In this case the maximum approximation error occurs in the centre of the interval and is proportional to the square of its half width, see figure 2 c. In higher-dimensional cases, the linear model is applied within the convex hull of simplices. The diameter of the circumspheres of these simplices play the role of the interval width then. As a result of the empty circle property, the circumspheres of Delaunay simplices have minimum diameters. Hence, they minimize the maximum approximation error.

2.1 Containment Tests

To process a query to a Delaunay network, the simplex the convex hull of which contains the query point x needs to be determined (see figure 1). This problem can be solved efficiently by means of *Barycentric Coordinates* (BC). By definition [7], the BC of a query point x with respect to a simplex T, are $n+1$ weights which move T's center into x. Formally, this is expressed by the set of linear equations

$$C(T) \cdot \mathrm{b} = \mathrm{x}, \qquad (1)$$

where $C(T)$ denotes a matrix of the (cartesian) coordinates of T's vertices and b is the vector of BC, $\mathrm{b} = (b_1, \ldots, b_{n+1})^T$. An important property of the BC is, that their signs describe on which side of T's edges the query point x is located. If x is located inside T's convex hull, $x \in CH(T)$, all $n+1$ BC are positive, whereas if x is located beyond one of T's edges[3], the respective BC is negative. BC provide an efficient means for selecting the active nodes, since they can be computed easily for low-dimensional problems by calculating determinants (i.e. solving (1) by applying Cramer's rule).

3 Hierarchical Delaunay Networks

The proposed methods allow the implementation of interpolation networks which yield very short response times; results obtained on a T805 processor were reported in [3]. However, the disadvantages of DN are the relatively large strorage capacity needed for the representation of the triangulation and the fact, that inserting or deleting nodes is a computationally complex task[4]. Node insertion and deletion is essential, however, when a DN is constructed automatically on the basis of a set of training data.

Hierarchical Delaunay Networks (HDN) cope with these difficulties by memorizing the triangulation for a subset of $\widehat{N} \leq N$ nodes only. Hence, memory requirements can be reduced. The Delaunay triangulation for this subset is computed off-line. Additional nodes can be inserted and deleted again without the need to update the topological database. This results in a considerable speed-up of network construction procedures.

[3] In the n-dimensional case, the edges are $(n-1)$-dimensional planes.

[4] If an additional node is inserted, some simplices are likely to loose the empty-circle property. In case of node deletion, those simplices which had the deleted node as a vertex, are destroyed. In either case, the triangulation needs to be recomputed and the database needs to be updated.

3.1 Topological Transitions

To compute the response of a HDN, it is necessary to reconstruct the Delaunay triangulation locally, i.e. in the vicinity of the current query point $\mathbf{x}$. Starting with that simplex of the memorized triangulation that contains $\mathbf{x}$, the triangulation is iteratively refined under consideration of all N nodes. These refinements are carried out by applying *topological transitions* [4] and *containment tests* (see section 2.1).

Given the simplex T that contains the current query point $\mathbf{x}$, the node list is searched for a node inside the circumsphere of T. If no such node can be found, T is a global Delaunay simplex and can be used for computing the network response. Otherwise, the local Delaunay triangulation for the set of $(n+2)$ nodes[5] is computed. From $n+2$ nodes, $\binom{n+2}{n+1} = n+2$ simplices can be constructed. One of theses candidates is identical to T and hence needs no further investigation. The remaining $n+1$ simplices T_i are defined as follows:

$$T_i = T \backslash \{p_i\} \cup \{p\} \,. \tag{2}$$

Here, p denotes the node detected in T's circumsphere, p_i, $\quad i = 1, \ldots, n+1$ are the vertices of T. The i-th candidate simplex T_i is obtained if the i-th vertex of T is replaced by p. The task to be solved now, is to determine that candidate T_j with the following properties:

1. T_j is a *local* Delaunay simplex, i.e. its circumsphere does not contain the remaining node.
2. T_j contains the query point $\mathbf{x}$.

The first requirement is met, if the i-th barycentric coordinate of the location of the node p with respect to the simplex T is positive. This result is quoted here without proof, the reader is referred to [8] for a comprehensive discussion. The second criterion can be checked by the containment test explained in section 2.1. After the candidate T_j which posesses both properties, is found, the procedure is repeated with T_j taking T's place. This iteration is continued until one of the following conditions hold true:

1. T is a global Delaunay simplex, i.e. there is no node within its circumsphere.
2. A prescribed maximum number of t_{max} iterations has been performed.

In either case, the currently existing simplex T is used for output interpolation. It can be guaranteed that T contains the query point $\mathbf{x}$. If the procedure was stopped because of the prescribed maximum number of iterations, T might not be a global Delaunay simplex; in section 4 the impact of the limit t_{max} on the worst case response time as well as on the model accuracy is investigated. Figure 3 summarizes the algorithm.

```
Find node p∈CS(T)
if (no such node exists) OR (max no. of iterations reached)
  compute network response using simplex T
else
    for i=1, ..., n+1
      construct candidate T_i according to eq. (2)
      if (p_i∉CS(T_i)) AND (x∈CH(T_i))
        set T = T_i and start from beginning
```

Symbols:

x	query point
T	simplex containing x
CH(T)	convex hull of T
CS(T)	circumsphere of T

Figure 3: Algorithm for the iterative local reconstruction of the Delaunay triangulation. The procedure starts with a simplex T which is part of the memorized triangulation and contains the query point $\mathbf{x}$.

3.2 Selection of Nodes for Off-line Triangulation

The HDN approach requires the selection of an appropriate subset of $\widehat{N} \leq N$ nodes and the computation of the Delaunay triangulation for this subset. $\widehat{N}$ serves as a trade-off parameter for response time and memory expense. Thus, it has to be determined according to the application-specific requirements and resources. After specifying *how many* nodes are included in the memorized triangulation, it is necessary

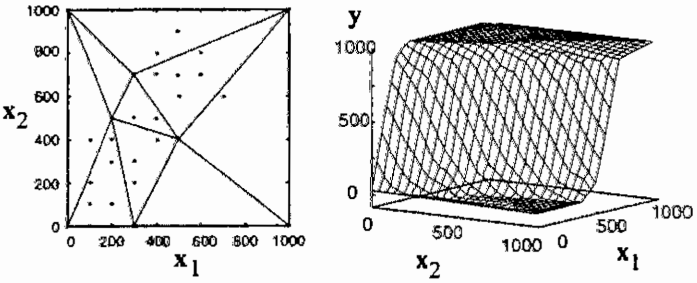
[5] n+1 nodes are the vertices of T, the additional node is located inside T's circumsphere.

to determine *which* nodes should be included. An appropriate selection can be done automatically, if the fact is considered that incremental construction techniques typically lead to *fractal growth* of the network: The node distribution after a few node insertions qualitatively is similar to that of the final network. Hence, it is recommended to select the *first* $\widehat{N}$ nodes that were inserted into the network during its construction[6]. This choice assures that the circumspheres of all simplices of the memorized triangulation contain approximately the same number of nodes and therefore the number of topological transitions will be relatively independent from the positions of the query points.

4 Results

To illustrate the approach and to investigate its performance, we constructed a HDN model of the nonlinear bivariate function

$$y(x_1, x_2) = 1000 \cdot \left[1 + e^{(2-\frac{1}{50}x_1+\frac{1}{100}x_2)}\right]^{-1}. \tag{3}$$



This equation simulates the steady-state torque characteristics of a combustion engine as a function of the throttle valve position and the engine speed. The figure illustrates a HDN model of this function with $N = 32$ nodes. In the depicted example, $\widehat{N} = 8$ nodes are part of the memorized triangulation. The right diagram shows the interpolated surface, i.e. the model's input/output relationship.

In figure 4 the impact of the number of nodes $\widehat{N}$ which are included in the memorized triangulation on the memory expense and response time is illustrated. The more nodes are included, the higher is the required storage capacity and the lower are the network's response times, since fewer topological transitions need to be carried out. The timings were obtained on a T805 microprocessor running at 30 MHz. The query points used in this simulation were derived from real-world signals, measured in a test vehicle. Apparently, there is a large difference between the *average* and the *maximum* response time. This is due to the fact, that the query point moves slowly in stationary operating modes. Once a global Delaunay simplex is reconstructed, it can be used in successive query cycles. However, in transient conditions, the Delaunay triangulation needs to be reconstructed in another area of the input space. This requires a large number of topological transitions and leads to the maximum response time.

In the simulations discussed so far, the number of topological transitions was not limited, i.e. the local reconstruction algorithm came up with a global Delaunay simplex for each query point. It is worthwile investigating, how a limited number of topological transitions per query affects the network's performance (see figure 4). The maximum response time decreases linearly with the reduced number t_{max} of topological transitions. In some situations, the network response is computed using a non-Delaunay simplex then. Since Delaunay simplices yield the minimum worst case approximation error, the network responses obtained with a certain limit t_{max} are compared to the responses with unbounded t_{max}. The lower diagram in figure 4 (right) shows the *maximum absolute* deviation. It is important to note, that this maximum error occurs only *temporarily*, since the iterative refinement of the simplices is continued in successive query cycles.

Figure 4 (right) suggests that t_{max} can be set to approximately 75% of the value needed for reconstructing global Delaunay simplices in every query cycle without significant loss of accuracy. In the example discussed here, at maximum 13 transitions are necessary; setting t_{max} to 10 yields only minor errors but reduces the maximum response time by 229 μsec.

5 Summary

Finite element based interpolation networks are a computationally efficient alternative to artificial neural networks for real-time modelling and control. When simplices are used as finite elements, the node

[6] Algorithms for data-driven construction of Delaunay networks are discussed in the accompanying paper by Brown and Ullrich.

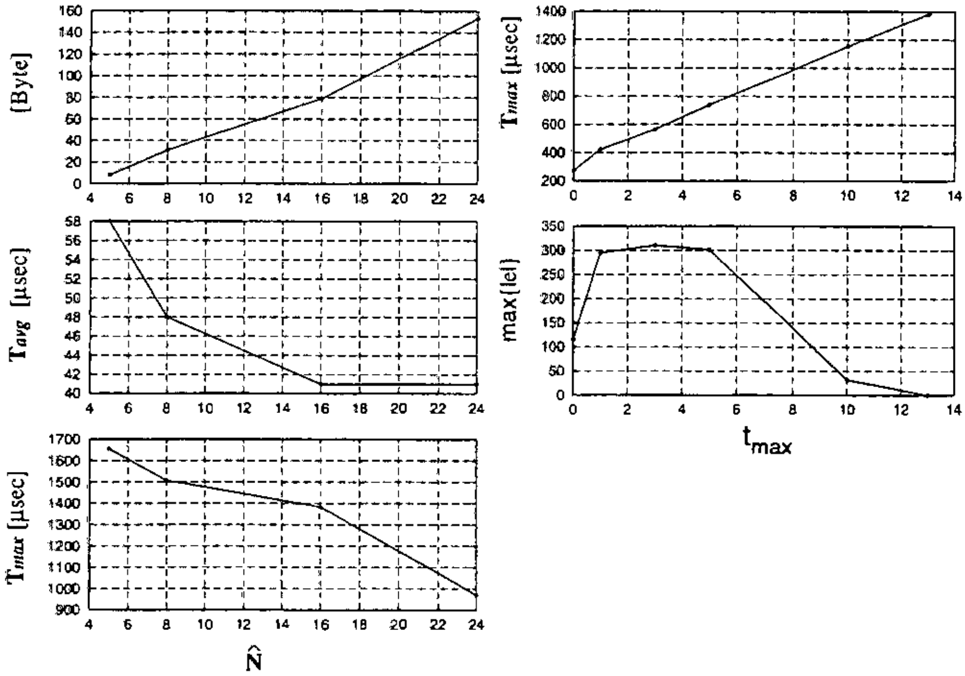


Figure 4: **Left**: Impact of the parameter $\hat{N}$ on memory expense and average (T_{avg}) and maximum (T_{max}) response time. **Right**: Impact of the parameter t_{max} on the maximum response time and model accuracy.

distribution can be adjusted to the local complexity of the underlying function. Hence, parsimonious models can be built. The Delaunay triangulation is the appropriate structure for such networks since it minimizes the worst case approximation error. The further development of this concept to hierarchical Delaunay networks was described in the paper. The HDN approach allows for memory expense to be traded-off against the network's response time and thus the model can be adapted to meet application-specific constraints.

Acknowledgements

This work is sponsored by the German National Science Foundation (DFG) under grant To 75/22-1. Furthermore, the author is grateful to Dr. T. Roos, Federal Institute of Technology at Zurich, for the helpful discussions on computational geometry.

References

[1] M. Brown and C. Harris. *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall, Hemel-Hempstead, 1994.

[2] S. M. Omohundro. Geometric learning algorithmns. Technical Report 89-041, International Computer Science Institute, Berkeley, California, USA, 1989.

[3] T. Ullrich and H. Tolle. Delaunay networks for modelling of non-linear processes. In *IASTED/ISMM International Conference Modelling and Simulation*, pages 319–322, Pittsburgh, USA, April 1996.

[4] L. Guibas, J. Mitchell, and T. Roos. Voronoi diagrams of moving points in the plane. In *17th International Workshop on Graphtheoretic*, pages 113–125, Fischbachau, Germany, 1991.

[5] A. K. Cline and R. L. Renka. A storage-efficient method for construction of a thiessen triangulation. *Rocky Mountain Journal of Mathematics*, 14:119–139, 1984.

[6] S. M. Omohundro. The delaunay triangulation and function learning. Technical Report 90-001, International Computer Science Institute, Berkeley, California, 1989.

[7] I. N. Bronstein and K. A. Semendajew. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, 21 edition, 1982.

[8] G. Albers, J. S. B. Mitchell, L. J. Guibas, and T. Roos. Voronoi diagrams of moving points. *International Journal of Computational Geometry & Applications*, 1996.

Comparison of Node Insertion Algorithms for Delaunay Networks

[1]M. Brown and [2]T. Ullrich,
[1]ISIS Research Group, Dept. Electronics and Computer Science,
Southampton University, UK,
Email: mqb@ecs.soton.ac.uk
[2]Control Systems Theory & Robotics Dept.,
Darmstadt University of Technology, Germany,
Email: thul@rt.e-technik.th-darmstadt.de

Abstract

This paper compares three different node insertion strategies which can be used to incrementally construct Delaunay triangulation models. Delaunay networks may be used to efficiently store low dimensional nonlinear models, and can therefore be used in a wide range of real-time applications. However, there are no direct node selection methods, and it can be shown that the network's generalisation abilities are strongly affected by the triangular partitioning of the input space. The three iterative, constructive node insertion algorithms (maximum error, local weighted error and one-step-ahead optimal search) are compared using two data sets, and conclusions are drawn about the quality of the extracted triangulation and the algorithms' computational costs.

Introduction

Due to their universal approximation capabilities, Artificial Neural Networks (ANNs) can be used to successfully model and control a wide range of nonlinear plants. However, the ANNs have some undesirable properties which prevent them from being widely applied in industry. The construction and training of ANNs is often difficult, their implementation can be resource-demanding and it is non-trivial to verify and validate the information extracted from the data. For real-time applications, where computer power and storage capacity is a question of cost, alternative methods of function approximation need to be developed.

Interpolation networks based on *interpolation nodes* are an alternative. The state of the art in tbis field are lattice-based networks where the nodes are located on a multidimensional grid, yielding hypercubes as finite elements [4]. This produces a very regular and predictable generalisation pattern, although the lattice structure suffers from the *curse of dimensionality*, i.e. the required storage capacity grows exponentially with respect to the input space dimension.

This drawback can partially be overcome by allowing arbitrary positions for the interpolation nodes. Instead of using hypercubes, it may be appropriate to define *simplices* as finite elements. A simplex (triangle in 2D, tetrahedron in 3D) connects $n+1$ nodes in the n-dimensional input space, thus allowing the network output be computed by *piecewise linear interpolation*. Higher order (quadratic) surfaces are possible, although the piecewise linear elements defined on simplices is by far the most popular approach.

The construction of Delaunay networks

Delaunay Networks (DN) have been proposed to efficiently implement simplex-based, finite element interpolators, [8], in contrast to earlier work [1]. Delaunay simplices have the property of empty circumspheres and can be shown to minimise the maximum approximation error [6, 7], and the input/output relationship of a DN can be described by:

$$y = \phi^T(\mathbf{x}) \cdot \mathbf{w} \tag{1}$$

where the network's response, y, for a query point, $\mathbf{x}$, is given by the inner product of the basis function vector, $\phi(\mathbf{x})$, and a weight vector, $\mathbf{w}$, which contains the attributes[1] assigned to the interpolation nodes.

[1] The *attribute* (or weight) of a node is an estimate of the modelled function's value at the node's position.

For any query point in the n-dimensional input space, only $n + 1$ basis functions have non-zero values. These basis functions correspond to the nodes which are vertices of the simplex whose convex hull contains $\mathbf{x}$. It is essential to note, that the network's output is *linearly* dependent on the weight vector whereas it is a *nonlinear* function of the nodes' positions, which determine the basis functions' shapes.

When constructing a DN to model a given set of training data, i.e. a set of t query points and desired responses, $\{\mathbf{x}_i, \widehat{y}_i\}_{i=1}^{t}$, the network's weights, $\mathbf{w}$, can be efficiently tuned by linear Least Squares (LS) optimisation [5]. To obtain parsimonious models[2], it is necessary to adjust the distribution of the nodes to the locally varying complexity of the underlying characteristic. This is a nonlinear optimisation problem to which heuristic strategies must be applied.

To obtain a suitable partition of the input space, it is generally necessary to use a construction algorithm which incrementally builds up a "good" model at each iteration. Most ANNs use gradient descent-type iterative parameter update rules, although these would cause the Delaunay triangulation to "flip" at certain points, leading to a very non-smooth performance function. In addition, it is difficult to estimate a suitable number of nodes. Therefore, iterative model building approaches which incrementally construct a suitable partition are appropriate.

Forwards selection algorithms choose a set of different possible refinements from a base network and evaluate how well they model the data, with respect to the performance criteria. The best refinement is included in the current model, and the process is repeated until either the designer, or a termination criteria, decide that the model is acceptable. It is not a trivial decision to determine when the network is a suitably good model of the data, as the network should try and reproduce the underlying relationships contained in the data without being influenced by either the noise or the intrinsic modelling error (bias) of the particular algorithm. In practice, several models of differing sizes may be constructed and then their generalisation abilities should be assessed.

Forwards selection algorithms have been applied to input variable and knot selection in more conventional lattice-based finite element networks [2]. In this paper, three such algorithms are applied to a DN and their performance and computational costs are compared on two test data modelling problems. These one-step-ahead iterative model building approaches have the potential to produce a suitable partition of the input space, possibly taking into account the complexity of the underlying, unknown function and the input data density. This will be discussed further when the individual algorithms are described.

Node insertion algorithms

Three heuristic model building algorithms are now described which can be used to iteratively insert nodes into a Delaunay network. After each insertion, the current network's weight vector, $\mathbf{w}$, is tuned by LS optimisation.

The *Maximum Error* (ME) model building algorithm simply inserts a node at the input value for which the current residual output error is maximum. In reality, this is not a forwards selection algorithm as there is only one refinement candidate produced which is always included in the model. However, forwards selection-type algorithms can be obtained which evaluate either the r points with the largest errors, or randomly select r points where the probability of selection depends on the size of the output error.

In practice, the use of the ME algorithm can have problems, as the point of maximum error doesn't always lie in the region where the optimal node insertion should occur. Indeed, the point of maximum error often initially lies on the edge of the data domain (when performing simple LS data fitting), and this can often initially bias this approach to node insertion, see figure 1.

Instead of selecting the point for which the output error is largest, the *Locally Weighted Error* (LWE) model building algorithm computes the Summed Square output Error (SSE) for every simplex, T_i. $\mathrm{SSE}(T_i)$ is determined by the output errors $\widehat{y}(\mathbf{x}) - y(\mathbf{x})$ that occur for query points $\mathbf{x}$ inside the convex hull H_i of the simplex T_i.

$$\mathrm{SSE}(T_i) = \sum_{\mathbf{x}\in H_i} (\widehat{y}(\mathbf{x}) - y(\mathbf{x}))^2 \tag{2}$$

An additional node is inserted at the position

$$\mathbf{p} = \frac{\sum_{\mathbf{x}\in H_j} \mathbf{x} \cdot (\widehat{y}(\mathbf{x}) - y(\mathbf{x}))^2}{\mathrm{SSE}(T_j)} \tag{3}$$

[2] A parsimonious network models the given data to a certain level of accuracy with the fewest possible nodes.

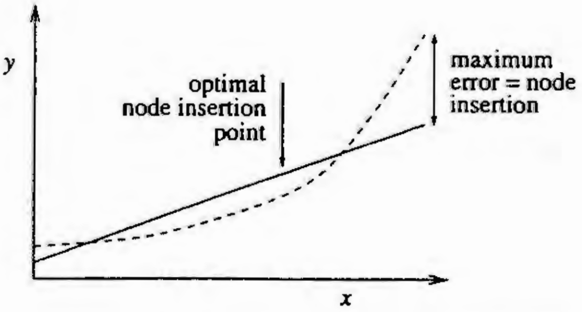


Figure 1: An example of undesirable behaviour with the Maximum Error algorithm.

where $T_j = \arg\max(SSE(T_i))$. The node is inserted inside the simplex with the largest SSE and the query points inside that simplex are weighted according to the (normalised) error. It is worthwhile noting that this node selection algorithm, unlike the other two, does not necessarily place the nodes at the input data points. Instead, nodes may be inserted between two data clusters, although they *are* always inserted inside the simplex with the maximum error, due to the convex calculation in equation 3.

The combination of reducing the output error and the local weighting scheme ensures that this algorithm tries to model the complexity of the underlying function as well as the input data density distribution. However, convergence can sometimes be a little erratic, especially when the first few nodes are inserted, due to the input space triangulation partitioning changing significantly when a single point is added.

The one-step *Optimal Search* (OS) forwards selection algorithm evaluates every possible data point as a candidate for node insertion, and includes the one which reduces the Mean Squared output Error (MSE) the most for the current network. It is a very simple but computationally intensive approach to model building, although it does have the advantage that, in some cases, significant information about the underlying functional relationships contained in the training data can be extracted from the final network's structure. This is possible because the network's output is a linear function on each triangular segment, and if some of the inputs are *affine*[3], nodes should not occur along these input boundaries, rather they should be placed on the hyperplane boundaries describing the remaining inputs. Affine systems are very important in modelling and control applications, and their detection is an important area of research.

Examples

In this section, the three construction algorithms are applied to two data sets describing real-world problems. In both cases, the network's input space is two-dimensional, which allows the network's structure and the interpolated surface to be visualised and interpreted easily. It must be pointed out, however, that the DN approach is not efficiently applicable to high-dimensional problems, although it is certainly possible to extend the method to three- and possibly four-dimensional cases, but for higher dimensions the required storage capacity as well as the computational cost grows considerably. Moreover, in higher dimensions the method is prone to failure due to round-off errors. For high-dimensional modelling problems, it is advisable to exploit structural information, e.g. by constructing an ANOVA additive decomposition [3].

In this example, a set of data representing a steam model was generated. The inputs (temperature and pressure) lay on a (33×37) grid and there was no measurement noise on the output (steam density). This gave a set of 1221 data points for which an accurate model could be expected.

The models produced by the three refinement algorithms are shown in figure 2. The ME algorithm suffered from the previously mentioned problem, where nodes were inserted at the point of maximum error but this failed to significantly change the underlying model structure, hence nodes continued to be inserted at adjacent points. The model shown contains 26 nodes, but the ME algorithm was continued for up to 100 nodes and the same behaviour was observed. In earlier tests using the ME algorithm, the weight values were regularised and this reduced the effect of this phenomena, but it still existed in certain cases. The LWE and the OS algorithms produced MSE values of 0.00023 and 0.00025 with 17 and 13 nodes, respectively. Both produce good fits to the data and the number of nodes is small which improves

[3] An model is affine with respect to an input u if it can be expressed in the form $y = f(\mathbf{x})\,u$.

generalisation and reduces the implementation cost. However, the OS network placed all its nodes along one axis which shows that the second input is included as an affine variable. This interpretation is only possible because the refinement procedure was halted (by inspecting the MSE) before the network began to model "noise", and thus overfit the data.

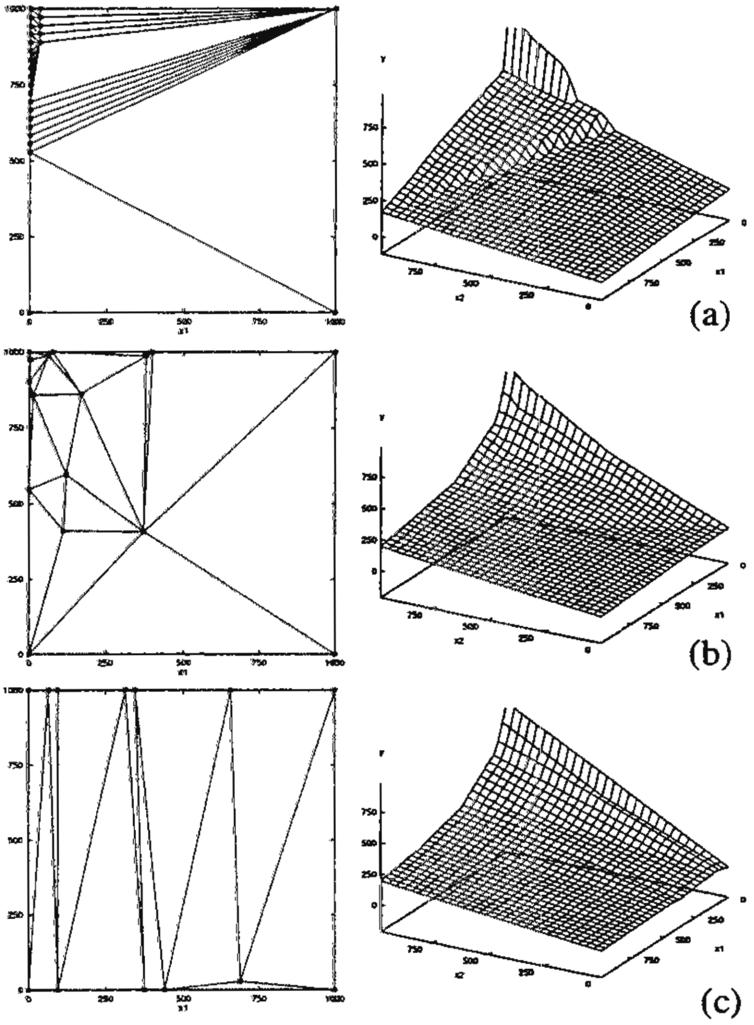


Figure 2: A comparison of the three different refinement strategies for the steam model data (a) Max Error (b) Locally Weighted Error (c) Optimal Search.

The three construction strategies were furthermore applied to a data set describing the steady-state torque of a combustion engine as a function of the two inputs throttle valve position and engine speed. The training set comprised 194 samples acquired on a test rig.

All three algorithms performed acceptably with the ME, LWE and OS networks producing MSE values of 0.0022, 0.0024 and 0.0023 using 26, 23 and 14 nodes, respectively (see figure 3). Again the network produced by the OS algorithm is smaller but the time taken to construct it was substantially longer than the other networks, see table 1. In all the tests performed so far, the ME and LWE construction algorithms take a similar time but the LWE procedure generally produces more reliable network structures. The triangulation of the input space produced by the OS algorithm is partially interpretable, as a set of nodes are placed along the ridge where the surface changes significantly, thus indicating that the two regions separated by this ridge are substantially different.

Discussion

The computationally simple ME algorithm leads to short construction times. However, because of its previously noted problems, it fails to build an accurate model of the steam data. As for the engine data,

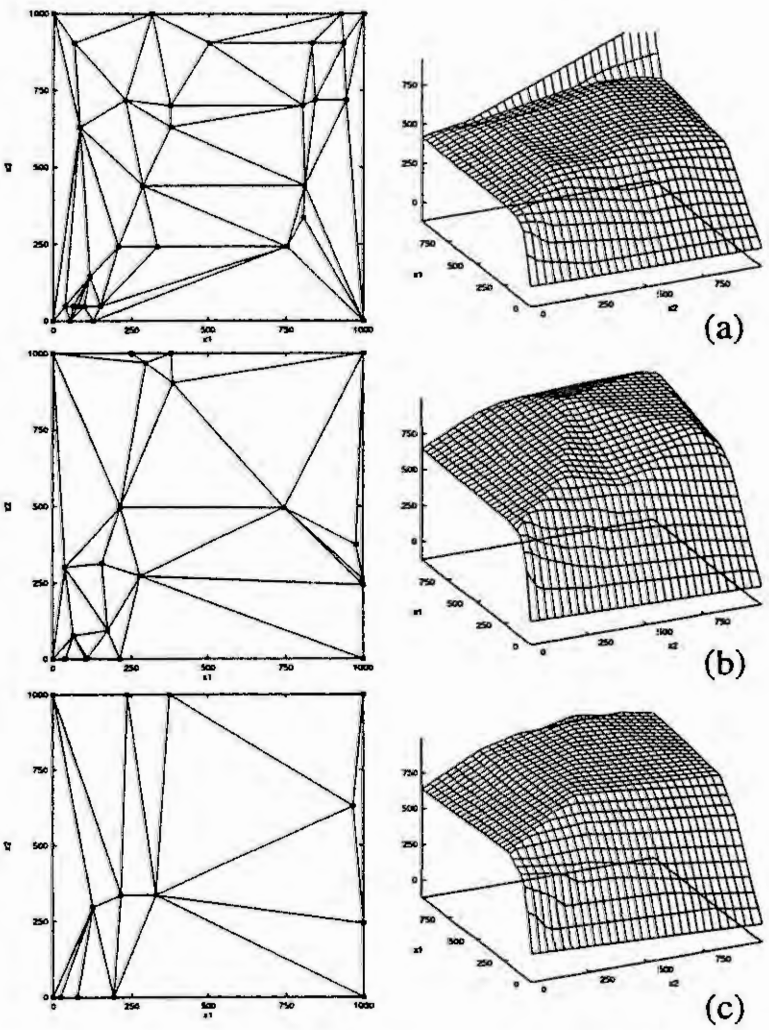


Figure 3: A comparison of the three different refinement strategies for the engine torque data (a) Max Error (b) Locally Weighted Error (c) Optimal Search.

Problem	final MSE $\cdot 10^{-3}$			nodes			comp. time [min:sec]		
	ME	LWE	OS	ME	LWE	OS	ME	LWE	OS
Steam	5.7	0.23	0.25	26	17	13	0:9	0:5	39:30
Engine	2.2	2.4	2.3	26	23	14	0:2	0:2	1:20

Table 1: Results of the three construction strategies on the steam and engine test data. The reported computation times were measured on a Sun Ultra 1-140 workstation.

the ME strategy produces a reasonably accurate model, but needs more nodes than the other methods.

The LWE method is also a computationally cheap approach to model building. When large training data sets, like the steam data which comprises 1221 samples, are processed, it is even faster than the ME algorithm. The computational cost of the LWE method is *output-sensitive*, i.e. its complexity mainly depends on the size of the network being constructed rather than on the size of the training set[4]. Generally, it is a fast construction procedure that produces accurate models and it is therefore recommended instead of the ME algorithm.

The OS strategy is a computationally expensive method and its complexity strongly depends on the size of the training set. For the steam data set, it takes nearly 40 minutes to construct the network, although this method has the ability to model the underlying function's structure, especially for affine models. It is worthwhile noting that the extracted information about the underlying function will only be interpretable if the data modelling procedure is halted at an appropriate time, i.e. before overfitting occurs.

In both examples investigated here, the OS algorithm achieves the same level of accuracy as the LWE algorithm with considerably fewer nodes. The OS method is thus recommended if the training set is not too large and the size of the final network is of the utmost importance because of limited computational resources of the target system. A possible way to speed up the OS algorithm would be to produce a smaller, filtered training data set (preserving the data spread) and use these centres as possible candidate refinements in the forwards selection algorithms.

Acknowledgements

The authors gratefully acknowledge the financial support of EPSRC (OSIRIS project GR/K55110) and the German National Science Foundation (DFG grant To 75/22-1). This work was started while Martin Brown was visiting Prof. Tolle's research group funded by the ISIA Graduiertenkolleg at Darmstadt University of Technology.

References

[1] C. Berger. Modeling dynamic systems using finite elements. In *IFAC World Congress*, volume 1, pages 333–336, Sydney, Australia, 1993.

[2] M. Brown, K.M. Bossley, and C. J. Harris. Neurofuzzy algorithms for model identification: Structure and parameter determination. In *IMACS/IEEE-SMC Multiconference on Computational Engineering in Systems Applications (CESA) - Symposium on Control, Optimization and Supervision*, volume 2, pages 1061–1066, Lille, France, 1996.

[3] M. Brown, K.M. Bossley, D.J. Mills, and C.J. Harris. High dimensional neurofuzzy systems: Overcoming the curse of dimensionality. In *Proc. Int. Joint Conf. of the 4th Int. Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Engineering Symp*, volume 2, pages 2139–2146, Yokohama, Japan, 1995. IEEE/IFES.

[4] M. Brown and C. Harris. *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall, Hemel-Hempstead, 1994.

[5] R. W. Farebrother. *Linear Least Squares Computations*. Marcel Dekker, New York, 1988.

[6] S. M. Omohundro. The Delaunay triangulation and function learning. Technical Report 90-001, International Computer Science Institute, Berkeley, California, 1989.

[7] T. Ullrich. Modelling nonlinear characteristics using hierarchical Delaunay networks. In *2nd IMACS Conference on Mathematical Modelling*, Vienna, Austria, 1997.

[8] T. Ullrich and H. Tolle. Delaunay networks for modelling of non-linear processes. In *IASTED/ISMM International Conference Modelling and Simulation*, pages 319–322, Pittsburgh, USA, April 1996.

[4] Of course, the size of the training set influences the computational effort of the LS (weight) training in all three algorithms.

Time-Suboptimal Control Design of Singularly Perturbed Systems by Reduced Order Feedback Design

S.A.Mikhailov, P.C. Müller
University of Wuppertal, Gauss str. 20, D-42097 Wuppertal

Abstract. In the paper a new method for feedback control of singularly perturbed systems is developed; it is based on the separation of slow and fast motions. The advantage is an important reduction of computational amount and a simplified implementation of the control. The disadvantage is the suboptimality but the error is in order of the small parameter ε.

1 Introduction

Problems of optimal control of singularly perturbed systems have been intensively studied(see the surveys of the literature on singular perturbation in control theory [3],[4]). These systems frequently occur in applications. Examples are: drives, actuators, robots and electronic circuits. The small parameter ε in these systems may represent small masses, small time constants, or the inverse of large stiffnesses or large gains. Consider a system described in state-space form by the set of equations

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 + B_1u \tag{1.1}$$

$$\varepsilon\dot{x}_2 = A_{21}x_1 + A_{22}x_2 + B_2u \tag{1.2}$$

where ε is a small positive scalar, $x_1 \in R^n$ is the slow state vector, $x_2 \in R^m$ is the fast state vector, $u \in R^k$ is the vector of control variables, and $\dot{()} = d/dt$. The control function is subject to the constraints

$$u(t) \in U \tag{1.3}$$

where U is a compact set in R^k.

The problem of time-optimal synthesis is considered, where the state variables at the initial moment of time can take arbitrary values in the state space:

$$x_1(0) = x_1^0,\ x_2(0) = x_2^0. \tag{1.4}$$

The task is to steer the initial state x_1^0, x_2^0 to the origin

$$x_1(T) = x_2(T) = 0 \tag{1.5}$$

in minimum time, taking into account the restrictions (1.3).

It should be mentioned tbat the answer of the considered synthesis problem is the switching surface in the state space. The control u must be defined as a function of state co-ordinates $u = u(x_1, x_2)$. Exact analytical solutions for optimal feedback control exist only for specific problems, sucb as linear systems with integral quadratic cost criteria usually for slow problems only. Numerical solutions are possible for optimal programs or open-loop controls, but it is very difficult to determine the numerical solutions for feedback controls if tbe dimension of the system is high. Therefore, we construct the approximate solution of the time-optimal problem by means of small parameter technique.

The problem of open-loop time-optimal control is considered in a number of studies [1]-[3]. We generalize these results to time-optimal feedback control.

The time-optimal synthesis in singularly perturbed systems has at least two special properties.

1. The switching surface is singularly perturbed (Fig.1)

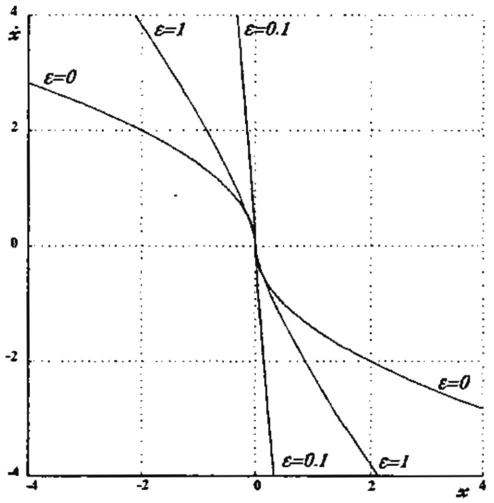


Fig. 1. Switching locus for various ε .

To illustrate this property, time-optimal synthesis for the following system is considered

$$\varepsilon\dddot{x}+\ddot{x}=u(x,\dot{x},\ddot{x},\varepsilon),\ \ |u|\leq 1,\ \ T\Rightarrow min_u \tag{1.6}$$

where $u(x,\dot{x},\ddot{x},\varepsilon)$ is the synthesizing function for the optimal feedback control. Fig. 1 shows the projection of the switching curve (two optimal trajectories which lead to zero) on the plane $x,\dot{x}$ for various ε. It is clear that this switching curve cannot be constructed by regular expansions in nonnegative powers of ε in a neighborhood of the point $\varepsilon=0$. It is said that such a function is singular in ε [4].

2. The application of the synthesis obtained in the so-called reduced system ($\varepsilon=0$) will lead

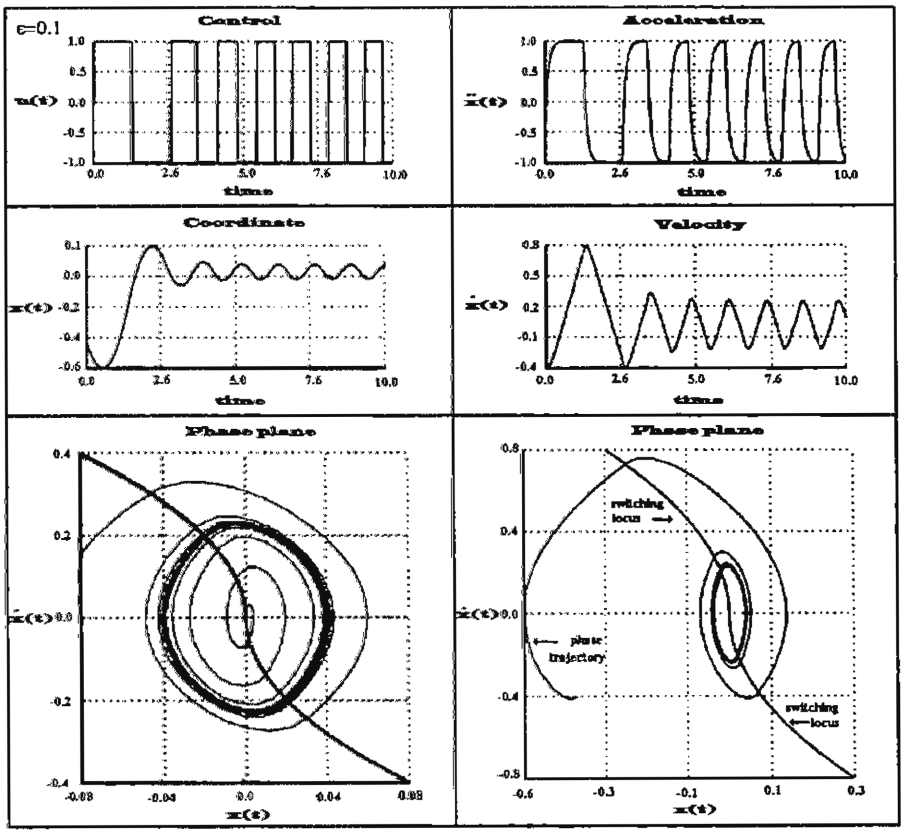


Fig. 2 The results of numerical simulation .

to a limit cycle in the original system. To demonstrate this property consider the example (1.6). For the reduced system $\ddot{x} = u(x, \dot{x})$ the switching curve (locus) has the explicit form $\dot{x} = -sgn(x)\sqrt{2|x|}$. If this synthesis is applied to the original system (1.6), the results given in Fig. 2 are obtained.

Fig. 2 shows the switching curve for the reduced system $\ddot{x} = u$ and the projection of the phase trajectory ($\varepsilon = 0.1$) on the phase plane $x, \dot{x}$. For highly accurate systems, the limit cycle is inadmissible. Therefore, we need to correct the feedback given by the reduced system in order to obtained a more precise control algorithm for steering the state trajectory to the origin.

2 The decoupling of the slow and fast variables

To alleviate the difficulties described above, we decouple motions in the system (1.1), (1.2). The Riccati transformation is used to obtain the uncoupled equations for the slow and fast variables [6]:

$$\begin{aligned} x_1 &= y_1 + \varepsilon D_2 y_2, \\ x_2 &= D_1 y_1 + (I_2 + \varepsilon D_1 D_2) y_2 \end{aligned} \tag{2.1}$$

where D_1, D_2 are determined from

$$A_{21} + A_{22}D_1 - \varepsilon D_1 A_{11} - \varepsilon D_1 A_{12} D_1 = 0, \tag{2.2}$$

$$A_{12} + \varepsilon(A_{11} + A_{12}D_1)D_2 - D_2(A_{22} - \varepsilon D_1 A_{12}) = 0. \tag{2.3}$$

By means of linear transformations of the state co-ordinates (2.1) we separate the fast and slow motions in the original system (1.1),(1.2) and obtain the following block-diagonal form of the governing equations

$$\dot{y_1} = F_1 y_1 + G_1 u, \tag{2.4}$$

$$\varepsilon \dot{y_2} = F_2 y_2 + G_2 u \tag{2.5}$$

with

$$F_1 = A_{11} + A_{12}D_1, \quad G_1 = B_1 - D_2 B_2 + \varepsilon D_2 D_1 B_1, \tag{2.6}$$

$$F_2 = A_{22} - \varepsilon D_1 A_{12}, \quad G_1 = B_2 - \varepsilon D_1 B_1. \tag{2.7}$$

The system (2.4),(2.5) is equivalent to the original equations (1.1),(1.2) but it is simpler. It consists of two subsystems of order n - slow mode and order m - fast mode interacting by means of control. Note that the system for the slow variable is regularly perturbed. Looking for the solution of (2.2) and (2.3) in the form of expansions in power series of ε,

$$D_1 = D_1^0 + \varepsilon D_1^1 + \cdots, \quad D_2 = D_2^0 + \varepsilon D_2^1 + \cdots,$$

and assuming that matrix A_{22} is regular, we obtain the solution up to the first order of ε:

$$D_1 = -A_{22}^{-1}A_{21} + \varepsilon A_{22}^{-2}A_{21}(-A_{11} + A_{12}A_{22}^{-1}A_{21}) + O(\varepsilon^2) \tag{2.8}$$

$$D_2 = A_{12}A_{22}^{-1} + \varepsilon(A_{11}A_{12}A_{22}^{-1} - A_{12}A_{22}^{-2}A_{21}A_{12} - A_{12}A_{22}^{-1}A_{21}A_{12}A_{22}^{-1})A_{22}^{-1} + O(\varepsilon^2).$$

The decomposition into separate slow and fast subsystems suggests that separate slow and fast control laws are designed for each subsystem, and then combined into a composite control of the original system. These ideas have produced numerous two-time-scale designs in linear state feedback, output feedback, observers and optimal control. It should be mentioned that this procedure for the system decomposition has been extended to linear time-varying systems [4].

3 New control algorithm

In this section we propose a composite control algorithm. It consists of two steps. The first step is the time-optimal design for the slow subsystem (2.4). The second step consist in a Lyapunov type control for the fast system (2.5).

First step. If the control constraints are decoupled, $U = \{u|\ |u_i(t)| \leq u_{i0}\}$, we have

$$u_1^* = -SGN(G_1^T \lambda_1)u_0 \tag{3.1}$$

where λ_1 is the adjoint vector. This control leads to a minimal time T in which the initial state $y_1(0) = y_{10}$ is transferred to $y_1(T) = 0$. This control will be applied in the time interval $[0, T - \varepsilon\tau]$ to the dynamical system leading to the states

$$y_1(T - \varepsilon\tau) = O(\varepsilon), \quad y_2(T - \varepsilon\tau) = O(1). \tag{3.2}$$

The time $\varepsilon\tau$ is sufficiently small such that $y_1(T - \varepsilon\tau)$ is in a small neighborhood of the origin $y_1 = 0$. τ will be determined in step 2.

Second step. In this step the fast variables y_2 will be driven approximately to the origin $y_2 = 0$ within the interval $[T - \varepsilon\tau, T]$. The control is designed by a Lyapunov procedure. Assuming $Re\lambda_i(F_2) < 0$, what essentially means asymptotic stability of A_{22}, then there are symmetric, positive matrices P_2, Q_2 such that

$$F_2^T P_2 + P_2 F_2 = -Q_2 \tag{3.3}$$

holds. Using a Lyapunov function $v = y_2^T P_2 y_2$ and minimizing pointwise its time-derivative then a feedback control

$$u_2^*(t) = -SGN(G_2^T P_2 y_2) u_0 \tag{3.4}$$

is designed for $[T - \varepsilon\tau, T]$. The time τ is determined by the requirement of

$$||y_2(T)|| = \varepsilon ||y_2(T - \varepsilon\tau)|| \tag{3.5}$$

$$\tau = p_{max}/q_{min} \ln(p_{max}/p_{min}\varepsilon^2)$$

where $p_{min} = \lambda_{min}(P_2)$, $p_{max} = \lambda_{max}(P_2)$, $q_{min} = \lambda_{min}(Q_2)$.

Thus the composite control algorithm consists of two phases. First, we solve the time-optimal problem for the slow mode. This problem is easier than the original one because we construct the synthesizing function in the space of slow variables, and the synthesis for the slow variables is regularly perturbed. Second we consider the terminal boundary layer and determine the Lyapunov type control in accordance with the formulas (3.4),(3.5). The system state is close to the origin and the control error is of the order of the small parameter ε:

$$x_1(T) = O(\varepsilon), \quad x_2(T) = O(\varepsilon). \tag{3.6}$$

4 Example

Now we consider an example which illustrates the proposed procedure.

$$\varepsilon \dddot{x} + \ddot{x} = u, \quad |u| \leq 1 \tag{4.1}$$

The third order differential equation (4.1) may be rewritten in the state space form

$$\dot{x}_{11} = x_{12}, \quad \dot{x}_{12} = x_2, \quad \varepsilon \dot{x}_2 = -x_2 + u. \tag{4.2}$$

The task is to steer x_{11}, x_{12} and x_2 to the origin in minimum time. In this example x_{11}, x_{12} are the slow variables and x_2 is the fast variable. The separation of slow and fast modes can be made by means of the transformation

$$x_{11} = y_{11} + \varepsilon^2 y_2, \quad x_{12} = y_{12} - \varepsilon y_2 \quad x_2 = y_2$$

After transformation, we obtain the uncoupled equation for the slow variables y_{11}, y_{12} and the fast variable y_2

$$\dot{y}_{11} = y_{12} - \varepsilon u, \quad \dot{y}_{12} = u, \quad \varepsilon \dot{y}_2 = -y_2 + u. \tag{4.3}$$

The system (4.3) has one remarkable property. By the transformation

$$y_{11} = \alpha - \varepsilon\beta, \quad y_{12} = \beta$$

the slow part can be reduced to a nonperturbed form

$$\dot{\alpha} = \beta, \quad \dot{\beta} = u, \quad \varepsilon \dot{y}_2 = -y_2 + u.$$

Time-optimal synthesis (3.1) for the slow variables (α, β) has the following form:

$$u_1^* = -sgn(\beta - sgn(\alpha))\sqrt{2|\alpha|}).$$

We implement this control until $\sqrt{x_{11}^2(t) + x_{12}^2(t)} > \varepsilon$. In the ε - vicinity of the origin the Lyapunov-type feedback (3.4) $u_2^* = -sgn(x_2)$ is used. Fig. 3 shows the state trajectory on the phase plane of the slow variables (x_{11}, x_{12}) and time history for the fast variables y_2.

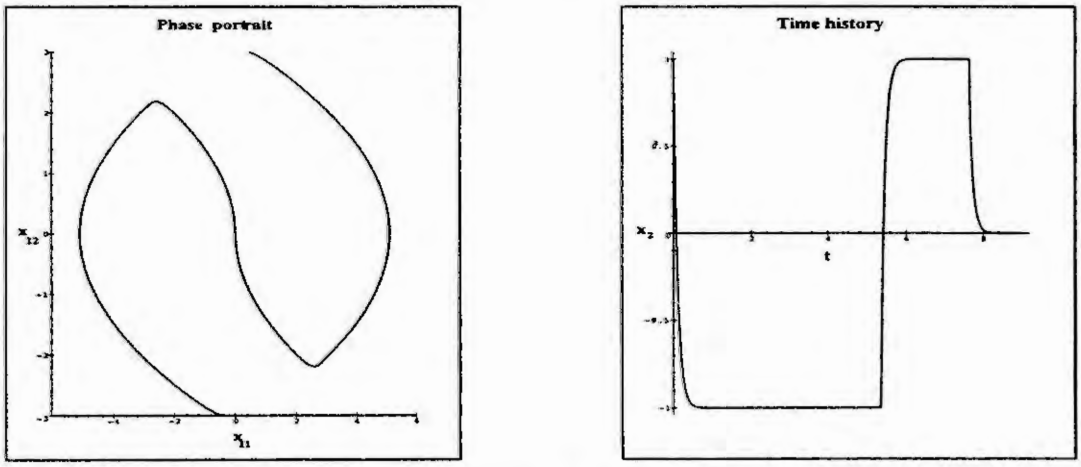


Fig. 3. The results of numerical simulation.

We can compare this composite design with another control algorithm for the same problem [5]. Feedback design in the terminal boundary layer is defined by the expression (3.6) [5], where $u_{1,2}^* = \mp 1$, $\tau_0 = \ln 2$. The values $\hat{\alpha}, \hat{\beta}$ are equal to

$$\hat{\alpha_{1,2}} = \mp(\varepsilon \ln 2)^2/2, \quad \hat{\beta_{1,2}} = \pm\varepsilon \ln 2 \tag{4.4}$$

In this example the switching locus has a complex analytical form and consists of two curves S_+, S_- (see Fig.3).

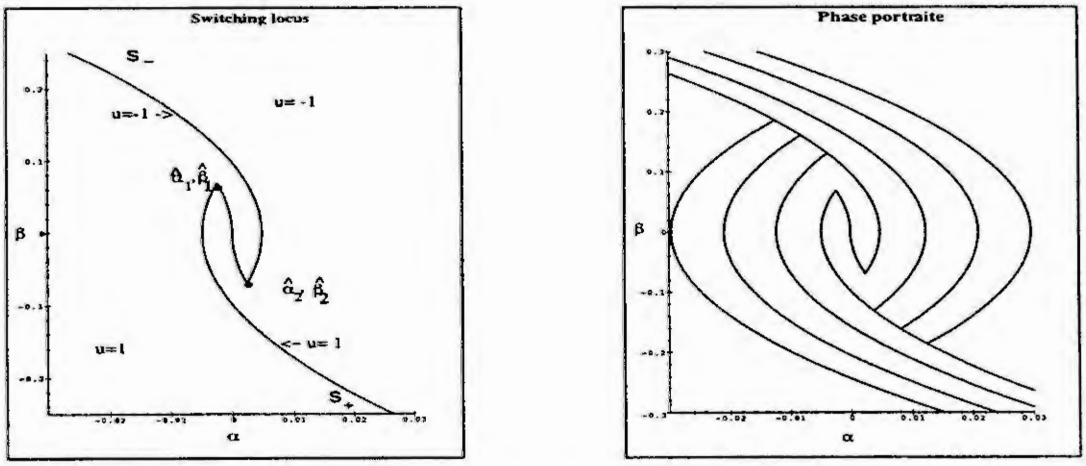


Fig. 4. The switching locus and phase portrait.

The switching locus divides the phase plane into two parts. For the phase points above the switching locus, the feedback control is $u(\alpha, \beta) = -1$; for the phase points below the switching locus, $u(\alpha, \beta) = 1$. In order to test the developed synthesis, the system (4.2) has been numerically simulated. Fig. 4 shows the switching locus for the slow mode and projections of the state trajectories on the plane (α, β) for various initial conditions. In the Fig. 5 the state trajectory for the fast variable x_2 is shown. The results of numerical simulations show that the composite control algorithm [5] can steer the phase point with arbitrary initial conditions to the origin in a period of time which is slightly longer than the slow mode minimum-time.

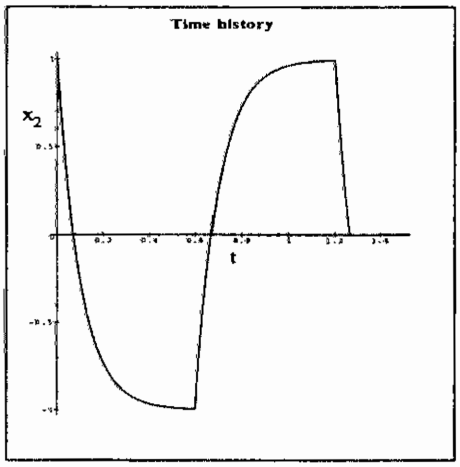


Fig. 5. The typical behaviour of the fast mode.

5 Conclusions

A composite algorithm has been developed for near-time-optimal feedback control of singularly perturbed linear systems. The main problems concerned with these systems are the difficulties with high-order systems, singularly perturbed switching surfaces (locus) and limit cycles. These difficulties have been successfully overcome by means of decoupling the slow and fast variables, and the construction of a composite algorithm. The control goal is the reduction of dimensionality to the dimension of the slow part. It should be pointed out that the synthesis for the slow part is regularly perturbed. Simulation results have shown the effectiveness of the composite algorithm, compared with the reduced order control.

REFERENCES

1. Collins, W.D. (1973) Singular perturbations of linear time-optimal control. *In Recent Mathematical Developments in Control, D. J. Bell, Ed. New York: Academic.*

2. Halanay, A., Mirica, St. (1979) The time-optimal feedback control for singular perturbed linear systems . *Rev. Roum. Math. Pures et Appl.,* 24, pp. 585-596.

3. Kokotovic, P.V., Khalil, H.K., and O'Reilly, J. (1986) *Singular perturbation methods in control: analysis and design.* Academic Press. 371 pp.

4. O'Malley, R.E. (1991) *Singular Perturbation Methods for Ordinary Differential Equations.* Springer-Verlag, Berlin. p. 225.

5. Mikhailov, S.A., Müller, P.C. Near-time-optimal feedback control of mechanical systems with fast and slow motions. IUTAM Symposium on Interaction Between Dynamics and Control in Advanced Mechanical Systems, April 21-26, 1996, Eindhoven, Netherlands.

6. . Smith, D.R. (1987). Decoupling and order reduction via the Riccati transformation. SIAM Review, vol. 29, no. 1, pp. 91-113.

PHASE MARGIN DESIGN OF PHASE LEAD CONTROLLERS USING AUTOTUNING

A.F. Boz and D.P. Atherton
School of Engineering, University of Sussex, Brighton BN1 9QT, U.K.
email address: atherton@sussex.ac.uk

Abstract: The paper discusses the problem of autotuning for phase lead controllers. Emperical formulae are derived to give the appropriate compensated gain crossover frequency from the autotuning parameters for different transfer functions when classical phase lead design is implemented. This enables the parameters of the controller to be set by autotuning.

1 Introduction

In many engineering situations it may not be possible to derive good models of the process to be controlled. This may be due to the difficulties of mathematical modelling or parameter identification, or in some cases lack of time and finance to undertake the work. Also many controllers have to be tuned, or retuned, on site relatively quickly by inexperienced personnel. Thus, it is important to have quick and easy methods to tune the parameters of standard type controllers. One of these methods is the autotuning approach. In recent years the autotuning approach to tuning PID controllers, initially suggested by Astrom and Hagglund[1], has proved particularly succesful in the process control industries. The procedure used is to replace the proportional gain used in the Ziegler-Nichols method by a relay which has the major advantage that it controls the limit cycle amplitude. Then from measurements of the amplitude and frequency of the resulting limit cycle, the gain margin(or critical point) of the process can be estimated. The accuracy of the estimation depends on the process dynamics since the approximate describing function, denoted DF, method is used in the resulting analysis. When good tuning rules for a simple, controller can be deduced from this test, then it has several advantages in that (*i*) no mathematical model of the process is required, (*ii*) tuning can be done 'in-situ' by process operators, (*iii*) the time required, depending on the process time constants, may not be significant and (*iv*) retuning can be done if the process parameters change with time.

For many plants such as servomechanisms, where integration exists in the plant transfer function, then compensation is often achieved using phase lead rather then PID control. This also means that when performing an autotuning experiment on these plants, since they are good low pass filters, DF analysis may be expected to be quite accurate, and better estimates for the critical frequency, ω_c, and gain, K_c, for use in the controller design[2], will be available than for the PID controller situation. Therefore in this paper the problem of tuning the parameters of a phase lead controller based on autotuning is investigated and results are given for doing this for different forms of plant transfer functions. Basically the approach assumes one of three forms of standard third and fourth order transfer function, although the results to be presented will show that often if one errs on the type of transfer function the designs are not too far out from their required values. In general, if one knows something about the structure of a system, then it is not too difficult to fit reasonably well one of the standard forms of the transfer functions considered. In reference [2] it was shown how the parameters of a phase lead controller could be obtained to provide a given phase margin by making an use of sets of graphs presented for each of the three types of transfer function. Here by using approximation to graphical data simple formulae are produced for obtaining the parameters of the phase lead controller.

2 Autotuning procedure

The basic idea behind the autotuning method is illustrated in Fig 1. A relay replaces the controller in the loop and the amplitude, a, and frequency, ω_o, of the resulting limit cycle are measured. Using its describing function, $4h/a\pi$, to approximate the relay gain, where $\pm h$ are the relay output levels, then it

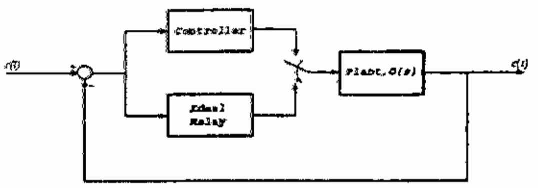


Figure 1: Relay autotuning procedure

is easily shown that

$$K_c = a\pi/4h \tag{1}$$

and

$$\omega_c = \omega_o \tag{2}$$

so that the critical point of the plant transfer function can be estimated from the test. It is now required to determine the parameters of the phase lead controller, from ω_c and K_c, so that the feedback loop has a given phase margin when the controller is used. To do this three commonly used transfer functions are considered, namely

$$G_1(s) = K_v/s(1+sT_1)(1+sT_2) \tag{3}$$

$$G_2(s) = K_v/s(1+sT_1)(1+sT_2)^2 \tag{4}$$

$$G_3(s) = (K_v/s(1+sT_1))e^{-sT_d}. \tag{5}$$

If the normalised frequency, $s' = sT_1$, is used then these transfer functions may be written in the frequency normalised form

$$G_1(s') = K/s'(1+s')(1+\rho s') \tag{6}$$

$$G_2(s') = K/s'(1+s')(1+\rho s')^2 \tag{7}$$

$$G_3(s) = (K/s'(1+s'))e^{-\rho s'}. \tag{8}$$

They are dependent on two parameters only, namely $K = K_v T_1$ and ρ equal to T_2/T_1 for $G_1(s')$ and $G_2(s')$, and T_d/T_1 for $G_3(s')$. Thus it is seen that the frequency normalised plants have frequencies, ω', related to those, ω, of the original plants by $\omega' = \omega K/K_v$ and that for these equivalent frequencies the gains are the same, thus

$$\omega'_c = \omega_c K/K_v \tag{9}$$

and

$$K'_c = K_c \tag{10}$$

It is easily shown that ρ and K can be found from

$$\rho = 1/(\omega'_c)^2 \tag{11}$$

$$KK'_c = 1 + (\omega'_c)^2 \tag{12}$$

for $G_1(s')$,

$$\rho^2 + 2\rho = 1/(\omega'_c)^2 \tag{13}$$

$$KK'_c = \omega'_c\sqrt{1+(\omega'_c)^2}(1+\rho^2(\omega'_c)^2) \tag{14}$$

for $G_2(s')$, and finally

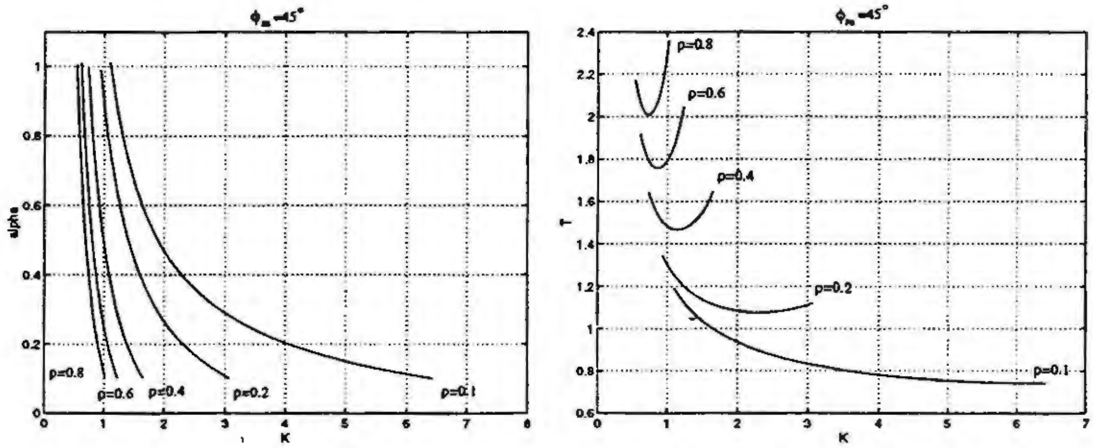
$$\rho^2 + 2\rho = 2/(\omega'_c)^2 \tag{15}$$

$$KK_c' = 2(\rho^2+2\rho+2)/\rho(\rho+2)^2 \tag{16}$$

for $G_3(s')$. Thus when a plant with one of the above transfer functions is autotuned the values of ω_c and K_c are measured, and if its K_v is known, which can normally be found by another simple test, the corresponding values of ω_c' and K_c' can be found from Eqns (9) and (10). Using the appropriate pair of Eqns (11) and (12), (13) and (14), or (15) and (16) the values of ρ and K for the corresponding normalised plant can be found. Then the required parameters, α and T, of the normalised phase lead controller

$$G_c(s') = (1+s'T)/(1+\alpha s'T) \tag{17}$$

to provide a given phase margin for closed loop control of the normalised plant can be calculated. Typically one can use the classical design procedure described in Dorf [3] and to enable this to be quickly implemented curves such as those shown in Fig 2 (a) and (b), which give α and T for a phase margin of 45°, were presented in reference [2] using a MATLAB program which implemented the classical design procedure. The required parameters for the phase lead controller for the given plant are then α and TK/K_v.



a) α

b) Normalised time constant,T,

Figure 2: α and T versus K for $\phi_m = 45^o$ phase margin

3 A new technique for tuning the phase lead controller

From the above MATLAB program the value of the centre frequency of the phase lead controller, which corresponds to the gain crossover frequency of the compensated normalised system, can also be evaluated and the results are shown in Fig 3 for a phase margin of $\phi_m = 45^o$ for $G_1(s')$. Although consideration is with the normalised design in this figure and throughout this section for ease of notation the prime is dropped from ω. Also since the open loop gain of the compensated system is unity at ω_m, one has

$$|G(j\omega_m)G_c(j\omega_m)| = \left|\frac{K\sqrt{1+\omega_m^2T^2}}{\omega_m\sqrt{1+\omega_m^2}\sqrt{1+\omega_m^2\rho^2}\sqrt{1+\omega_m^2\alpha^2T^2}}\right| = 1. \tag{18}$$

Also for the phase lead controller $T^2 = 1/\omega_m^2\alpha$ and using this in Eqn 18 yields a quadratic equation for α, namely

$$[\omega_m^2(\omega_m^2\rho^2+\rho^2+1)+1]\alpha^2+[\omega_m^2\rho^2(\omega_m^2+1)-(K^2/\omega_m^2)+\omega_m^2+1]\alpha-(K^2/\omega_m^2)=0. \tag{19}$$

Eqn 19 depends upon ρ and K which are known after autotuning and use of Eqns 11 and 12 and ω_m which can be found from appropriate curves, such as Fig 3, for the known ρ and K for $\phi_m = 45^o$. However, the storing of significant amounts of tabular data as would be required to represent the curves

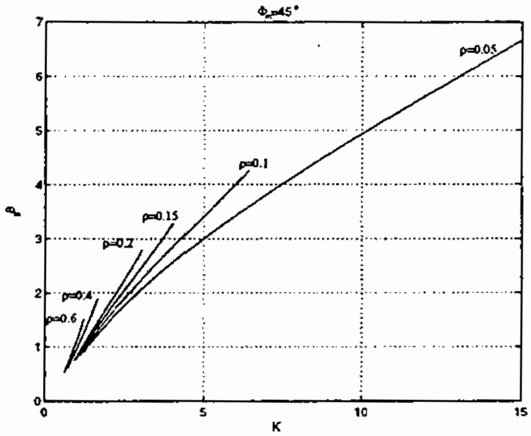


Figure 3: New location of ω_m for $G_1(s')$ for $\phi_m = 45°$

of Fig 3 for different phase margins and also interpolation routines are not appropriate for a real time autotuner. Thus a mathematical relationship for the 'ω_m curves' is desirable. It is clearly seen from Fig 3 that the ω_m curves for different values of ρ are almost linear, especially when ρ is greater then 0.1 . Therefore an approximation to linearise the ω_m curves is proposed for $G_1(s')$. The phase lead gives a maximum phase lead of ϕ at the frequency ω_m where

$$\sin\phi = (1-\alpha)/(1+\alpha) \qquad (20)$$

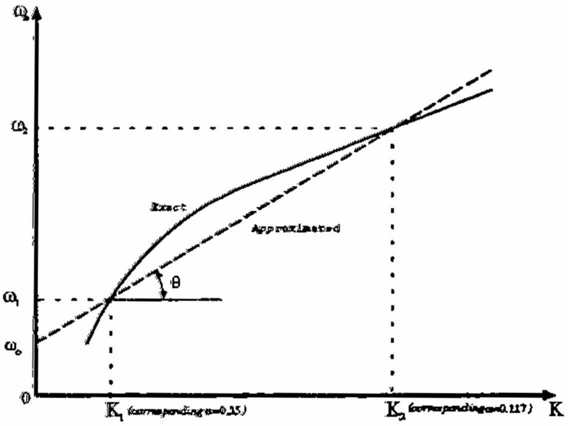


Figure 4: Approximation of the system's new location frequency of ω_m

By examination of the curves in Fig 3 a good linear approximation was found for most of them using values of $\alpha = 0.117$ and $\alpha = 0.35$ as illustrated in Fig 4. It is now required to find the value of ω and K, denoted with subscripts 1 and 2 at these points. Thus $\alpha = 0.35$ yields
$\sin\phi = (1-0.35)/(1+0.35)$, which gives $\phi = 28.78°$.
The compensated system's phase margin is given by

$$\phi_m = 180° + \angle G_1(j\omega) + 28.78°$$

where $\angle G_1(j\omega) = -90° - \arctan\omega - \arctan\rho\omega$ and ω_1, which is the location of the ω_m frequency for $\alpha = 0.35$ and can be easily found from

$$\tan(118.78 - \phi_m) = (\omega_1 + \rho\omega_1)/(1-\rho\omega_1^2) \qquad (21)$$

Also the uncompensated system gain is equal to $-10\log_{10}(1/\alpha)dB$ at ω_1, therefore

$$-10\log_{10}(1/\alpha)dB = 20\log_{10}(g)dB$$

where g is the uncompensated system gain and is found from $g = K/\omega_1\sqrt{1+\omega_1^2}\sqrt{1+\rho^2\omega_1^2}$. Thus K_1 for $\alpha = 0.35$ is given by

$$K_1 = 0.591\omega_1\sqrt{1+\omega_1^2}\sqrt{1+\rho^2\omega_1^2} \tag{22}$$

Similarly for $\alpha = 0.117$, ω_2 and K_2 can be found from

$$\tan(142.2 - \phi_m) = (\omega_2 + \rho\omega_2)/(1 - \rho\omega_2^2) \tag{23}$$

$$K_2 = 0.342\omega_2\sqrt{1+\omega_2^2}\sqrt{1+\rho^2\omega_2^2} \tag{24}$$

Thus the linear equation for ω_m can be written as

$$\omega_m = ((\omega_2 - \omega_1)/(K_2 - K_1))K + \omega_o \tag{25}$$

where $\omega_o = (\omega_1 K_2 - \omega_2 K_1)/(K_2 - K_1)$. For the plant $G_2(s')$ using the same technique but for different values of α, the equations obtained for ω_1, K_1, ω_2 and K_2 are

$$\tan(106.87 - \phi_m) = (\omega_1(-\rho^2\omega_1^2 + 2\rho + 1))/(1 - \omega_1^2(\rho^2 + 2\rho)) \tag{26}$$

$$K_1 = 0.7416\omega_1(1 + \rho^2\omega_1^2)\sqrt{1+\omega_1^2} \tag{27}$$

$$\tan(140.34 - \phi_m) = (\omega_2(-\rho^2\omega_2^2 + 2\rho + 1))/(1 - \omega_2^2(\rho^2 + 2\rho)) \tag{28}$$

$$K_2 = 0.3606\omega_2(1 + \rho^2\omega_2^2)\sqrt{1+\omega_2^2} \tag{29}$$

The value of α is now, however, obtained from an equation similar to Eqn 19 which is

$$X\alpha^2 + X\alpha - (K^2/\omega_m^2) = 0 \tag{30}$$

with $X = \omega_m^2(2\rho^2 + 2\omega_m^2\rho^2 + \omega_m^4\rho^4 + \omega_m^2\rho^4 + 1) + 1$.

And finally for $G_3(s')$ the equations are

$$\tan(104.02 - \phi_m) = (\omega_1(1 + \rho) - \rho^2\omega_1^3 1/4)/(1 - \omega_1^2(\rho + \rho^2 1/4)) \tag{31}$$

$$K_1 = 0.7810\omega_1\sqrt{1+\omega_1^2} \tag{32}$$

$$\tan(142.08 - \phi_m) = (\omega_2(1 + \rho) - \rho^2\omega_2^3 1/4)/(1 - \omega_2^2(\rho + \rho^2)1/4) \tag{33}$$

$$K_2 = 0.3435\omega_2\sqrt{1+\omega_2^2} \tag{34}$$

and α is found from

$$(1 + \omega_m^2)\alpha^2 + (1 + \omega_m^2 - (K^2/\omega_m^2)\alpha - K^2/\omega_m^2 = 0 \tag{35}$$

4 Examples

Several examples are considered in this section.

Example 1

Here the design of a phase lead controller for the plant transfer function $G_1(s) = 1.2/s(1+2s)(1+0.2s)$ is considered. The design procedure has been applied for required phase margins, ϕ_m of 45° and 55° respectively. The calculated exact values of ω_c and K_c are $\omega_c = 1.581$ rad/sec, $K_c = 4.581$ and K_v is taken as 1.2 . Using Eqns 11 and 12 gives $K = 2.3992$ and $\rho = 0.1001$. For $\phi_m = 45°$, Eqns 21, 22, 23, 24 and 25 yield ω_1, K_1, ω_2, K_2 and ω_m respectively as $\omega_1 = 1.9436$ rad/sec, $K_1 = 2.5605$, $\omega_2 = 3.9346$ rad/sec, $K_2 = 5.8710$ and $\omega_m = 1.8467$ rad/sec.

Inputting the ω_m, ρ and K into Eqn 19 yields $\alpha = 0.3701$ and T' is 0.8901 . It is also known that $T = T'K/K_v$, therefore $T = 1.7797$, and the transfer function of the controller is

$$G_c(s) = (1 + 1.7797s)/(1 + 0.6587s)$$

for $\phi_m = 45^o$ phase margin. For this controller the calculated system phase margin $\phi_m = 45.34^o$, which is near to the desired value.
Using exact analysis of a relay system to obtain the critical point of the plant yields $\omega_c = 1.517$ rad/sec and $K_c = 4.3067$. Then from Eqns 11 and 12 $K = 2.4382$ and $\rho = 0.1053$ and the resulting compensated system phase margin is $\phi_m = 45.81^o$.
Similarly using the exact values of ω_c and K_c for a required phase margin, $\phi_m = 55^o$ gives $\omega_1 = 1.4549$ rad/sec, $K_1 = 1.5355$, $\omega_2 = 2.9065$ rad/sec, $K_2 = 3.1820$ and $\omega_m = 2.2164$ rad/sec. α is 0.1889 and $T' = 1.0381$, thus $T = 2.0755$. The transfer function of the controller is

$$G_c(s) = (1 + 2.0755s)/(1 + 0.3921s)$$

and the resulting system phase margin is $\phi_m = 54.08^o$.
Using relay system analysis gives $\omega_c = 1.517$ rad/sec and $K_c = 4.3067$ and the resulting compensator gives a compensated system phase margin $\phi_m = 55.63^o$.

Example 2
Here a plant with the transfer function $G_2(s) = 1.2/s(1+2s)(1+0.2s)^2$ is considered. In this example, it is assumed for the design that the transfer function is $G_1(s)$, which it is not. Exact values of the critical gain and frequency are $K_c = 2.2864$, $\omega_c = 1.091$ rad/sec and K_v is taken as 1.2 .
Eqns 11 and 12 yield $K = 2.2215$ and $\rho = 0.2451$ and using Eqns 21, 22, 23, 24, 25 and 19 respectively for a required phase margin $\phi_m = 45^o$, α and T are found as $\alpha = 0.1441$ and $T = 2.2513$. The transfer function of the controller is

$$G_c(s) = (1 + 2.2513s)/(1 + 0.3244s)$$

The compensated system's phase margin is $\phi_m = 44.97^o$. For the same phase margin requirement relay analysis yields $\omega_c = 1.1$ rad/sec, $K_c = 2.22$, $K = 2.0659$, $\rho = 0.2788$, $\alpha = 0.1305$ and $T = 2.258$. In this case the compensated system's phase margin is $\phi_m = 46.27^o$. Carrying out the same procedure with the exact values for K_c and ω_c and the formulae for $G_2(s)$ results in a controller giving a phase margin $\phi_m = 45.21^o$.

Example 3
The transfer function of the plant is $G_3(s) = (1.2/s(1 + 2s))e^{-0.4s}$. Exact critical point values are $K_c = 2.1498$, $\omega_c = 1.082$ rad/sec and K_v is 1.2 .
Again assuming the results correspond to a $G_1(s)$ type transfer function, eqns 11 and 12 yield $K = 2.0419$ and $\rho = 0.295$ and using Eqns 21, 22, 23, 24, 25 and 19 respectively for a required phase margin $\phi_m = 45^o$, α and T are found as $\alpha = 0.1176$ and $T = 2.3230$. The transfer function of the controller is

$$G_c(s) = (1 + 2.323s)/(1 + 0.2732s)$$

The compensated system's phase margin is $\phi_m = 43.68^o$. This compares with $\phi_m = 44.11^o$ obtained using the formulae for $G_3(s)$.

5 Conclusion

A procedure has been given which enables the parameters of a phase lead controller to be set at approximately the same values as would be achieved using classical phase lead design using an autotuning approach.

References

[1] Astrom, K.J. and Hagglund, T., *Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins*, Automatica, Vol 20, No 5, 1984, 645-651

[2] Atherton, D.P., *Autotuning of Phase Advance Controller*, 4^{th} IEEE CCA, Albany, New York, Sept. 1995, 148-149.

[3] Dorf, R.C., *Modern Control Systems*, Addison-Wesley, 6^{th} Edition, 1992, 498.

THE RELATIONSHIP BETWEEN CONTROL REQUIREMENTS, PROCESS COMPLEXITY AND MODELLING EFFORT IN THE DESIGN PROCESS OF RIVER CONTROL SYSTEMS

Prof. Dr.-Ing. Bernd Cuno
Fachhochschule Fulda
Fachbereich Elektrotechnik
Marquardstraße 35
36039 Fulda
Germany

Dipl.-Ing. Stephan Theobald
Universität Karlsruhe
Institut für Wasserbau und Kulturtechnik
Kaiserstraße 12
76128 Karlsruhe
Germany

Abstract. The design effort for a control system is mainly determined by two aspects: complexity of the process to be controlled and the control requirements. The paper describes a method to design the structure and to optimize the parameters of the control of a given river reach under cosideration of the control requirements with a minimum of design effort as well as realization effort. The design is including water level control and flow control of the river.

Introduction

The operation of barrages in rivers (including hydraulic power plants) has to consider the following - often contrary - requirements: safety of navigation, intensive use of hydro power, flood protection, covering the water demand of the industry, use of the river for irrigation, minimizing the number of actuator operations (turbines, weirs). The main goal of the operation should be a constant and steady discharge within narrow water level tolerances.

Today the automatic control of a river reach or a cascade of reaches is of increasing importance. The above mentioned requirements have to be fulfilled by the control system with increasing demands in control performance and a high reliability of operation.

On the other hand there is the complex, highly nonlinear, unsteady and locally distributed hydraulic behaviour in the reach as well as the variable operation conditions of the barrage - depending on water quantity and availability - with changing discharges via turbines and weirs. The main influence on the hydraulic behaviour is determined by the operation of locks, the demand driven production of electric energy, outflow respectively inflow of the reach, rain, snow break, etc. Fig. 1 shows a river reservoir consisting of a reach and a barrage with weirs, turbines, and locks. The controlled variable is the water level (usually the headwater level), the manipulated variable is the discharge of the barrage. The water level should be controlled in a way that the most economic and complete exploitation of the water power is achieved. For the sake of reliability each reach is controlled by a local controller. In the case of a chain of reaches the reference value of the water level control is given by a central water management system considering global effects as power modulation by the needs of energy production and traffic by ships, weather, etc.

The control of a barrage is usually realized by industrial process control systems with highly reliable components and partly redundand structures.

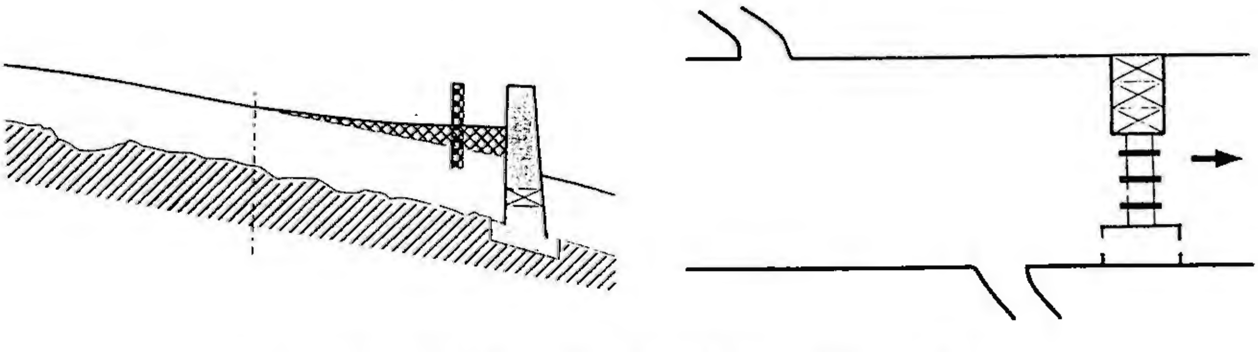


Fig. 1: River reach with barrage (consisting of turbines, weirs, and locks)
a) longitudinal section with gauge b) Top view with additional inflow and outflow

It is a well known fact that many automated installations of river reaches or hydro-electric power plants are suffering from latent stability problems caused by flow or water level oscillations. During operation the control loop may change from stable to oscillatory and limit cycle behaviour. The oscillations when encountered are usually subdued by opening the water level control loop and changing to pure flow control, power control or even manual operation. The manual as well as the automated operation of a chain of reaches may result in extreme flow oscillations - especially at the end of the chain - caused by the amplification of natural given flow changes downstream from barrage to barrage [1], [2].

Design procedure

The increasing requirements on dynamic performance, economic, and secure operation have stimulated the development of more and more sophisticated control strategies allowing fully automated operation. In practice up to now the control parameters are mostly tuned by trial-and-error methods or by rules-of-thumb. This procedure is quite time consuming and does not necessarily result in a best possible control performance. On the other hand many of the published theoretical proposals are based on oversimplified or even wrong system models, are too complex to be technically realized or show a high sensitivity against changes in operation mode or process dynamics.

The economic goal of the project engineer is to design, realize, and set up the control system with a minimum of effort. This effort is determined mainly by two aspects: complexity of the process to be controlled, control requirements given by the user, and operation conditions (e.g. given by laws). A way to minimize this effort is to base the design of the control structure on models and control algorithms as simple as possible. Fig. 2 illustrates the scheme of the control design procedure for a river reach. The structure and parameter design for more or less sophisticated control functions is done by rules-of-tumb, by simulation, or by powerful optimization methods. Models of different accuracy are available for design purposes.

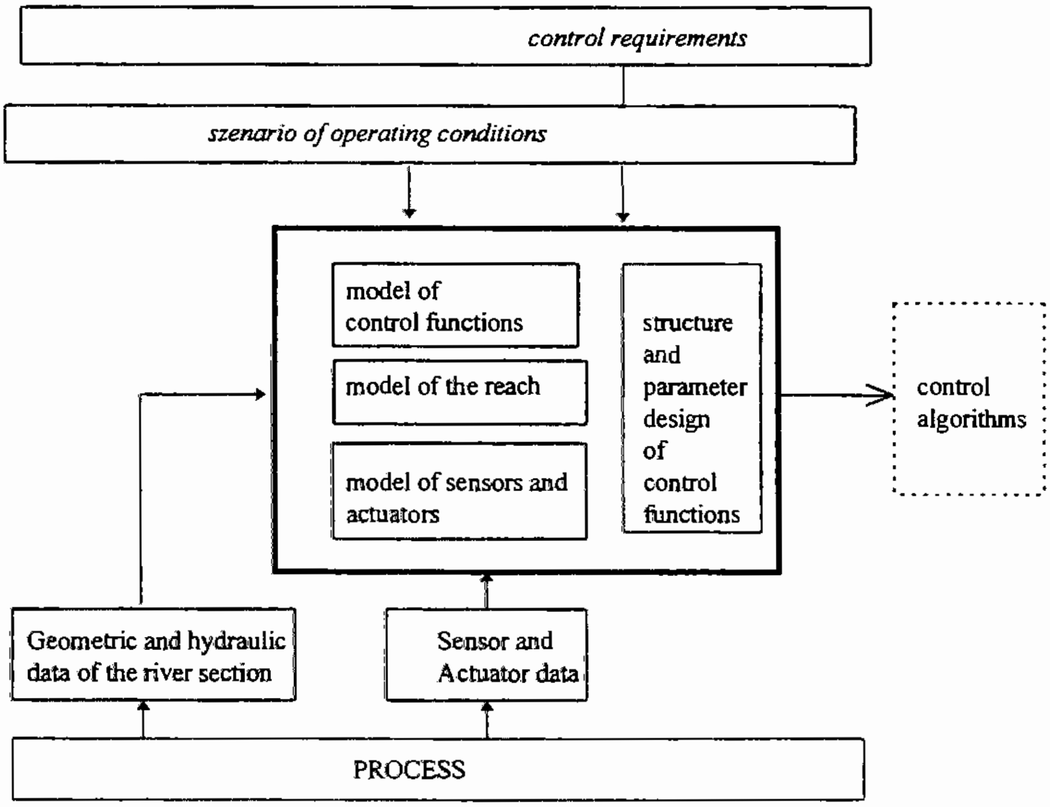

control requirements
szenario of operating conditions
model of control functions
model of the reach
model of sensors and actuators
structure and parameter design of control functions
control algorithms
Geometric and hydraulic data of the river section
Sensor and Actuator data
PROCESS

Fig. 2: Scheme of control design

As hydraulic systems are of a very complex nature, the synthesis of control structure as well as the optimization of its parameters prerequisite a mathematical model describing the process to be automated. Mathematical models of river reaches can be developed either theoretically by deriving the physical relationships of hydraulics, or empirically by experiments on the process or by a combination of both ways. Although an extensive number of scientific tools exist, modelling is determined by the skill and art of the designing engineer. The main problem is to find the right balance between simplicity and adequacy of the model for its intended use. Finally it has to be checked if the model agrees with the real process. This process model-validation is done by cross-checking model and real process behaviour.

The control functions of flow control are designed on the basis of the relevant actuator data. Fig. 3 shows the flow control loop by way of the example of a Kaplan turbine: the flow of the turbine Q_T is determined by the turbine opening a and by the operating head (difference between headwater level h_{OW} and tailwater level h_{UW}). As the real flow Q_T can not be measured directly, it is computed by a static observer: The result Q_T' is filtered and compared with the set value Q_T^*. Depending on the control error the flow controller manipulates the runner blade or gate opening of the turbine. As the positioning is done by a constant-speed reversible motor with three states, drive upward (opening), stop and drive downward (closing) the controller must be similarly arranged in its output signal. The simplest controller for this function is the three-state controller. A hysteresis reduces unnecesary on-off-switching. As the actual discharges of the actuators can not be measured directly they have to be calculated using measurable information. The discharge of each turbine is computed e.g. using the well known static characteristics of the hill (shell) diagram of a Kaplan turbine. Linearization araound the actual operation point (α^0, h^0) yields

$$Q_T(s) = k_{k\alpha}\,\alpha(s) + k_{Qh}\,h_F(s)$$

with the runner blade position α and the operating head h_F.

The discharge of the weirs is computed using well known formulas depending on weir type and discharge situation, e.g. free overfall and partially downed fishbelly flaps.

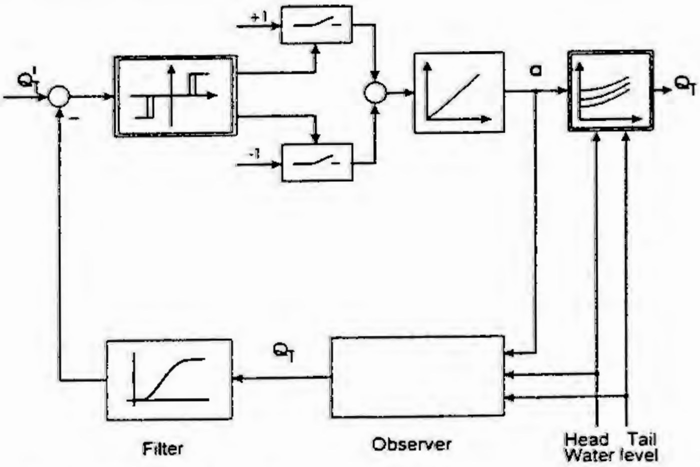


Fig. 3: Flow control loop (by example of a Kaplan turbine as actuator)

Starting with the most simple design step the water level controller is assumed to be a linear PI-controller with constant parameters. These parameters are calculated by rules-of-thumb based on models of the underlying flow control loops and on models of the reach. One can show that the time constants of the slave flow control loop are significantly smaller than the time constants of the master level control loop. That means that for the sake of simplicity the dynamics of the flow control loop can be neglected in a first design step. Furthermore a linear characteristic is assumed; that means the transfer function of the flow control loop is $G_Q(s) = 1$. In the past few years a variety of linearized lumped parameter model of river reaches have been published (see [3]-[6]). The interesting system performance (water level performance as a result of inflow and outflow of the reach) have been approximated by transfer functions of the following structure

$$G_{in,out}(s) = \frac{H(s)}{Q_{in,out}(s)} = \left[G_V(s) + G_W(s)\right]e^{-sT_t}.$$

whereas H(s) is the Laplace transform of the water level to be controlled and Q(s) is the Laplace transform of the inflow (Q_{in}) rsp. outflow (Q_{out}) . The first term of the transfer functions $G_{in,out}(s)$ describes the essential volume variations (storage capacity) of the reach and can be given by

$$G_V(s) = \frac{1}{A\,s}.$$

with the water surface A.

The second term of the transfer function models formation, propagation, reflection, and damping of waves and is described around each operation point by a linearized transfer function model

$$G_W(s)=\frac{a_0+a_1 s+a_2 s^2+a_3 s^3+...+a_n s^n}{1+b_1 s+b_2 s^2+b_s s^3+...+b_n s^n},\quad a_i=f(Q^0),\quad b_i=f(Q^0).$$

The water surface A and the coefficients a_i, b_i of the polynominals in the Laplace operator s are varying with the discharge (quasi stationary flow) Q^0 of the reach. Both rational terms of the transfer function G(s) are combined in series with the transfer function of a pure delay, whereas the Dead Time T_t is assumed to be the wave propagation time T_L. We found that none of the published models can be recommended. Stimulated by the results of a fundamental theoretical as well as experminental modeling [1] an analysis of reservoir performance revealed that not only one but two time delays are dominating the process of a river race [7], [8]

$$G_{in,out}(s)=\frac{H(s)}{Q(s)}=G_V(s)e^{-sT_R}+G_W(s)e^{-sT_L},\quad T_R=f(\Delta V/\Delta Q,Q^0),\quad T_L=f(\dot{Q},Q^0).$$

with the retention time constant T_R and the wave propagation time T_L. This model structure has been evaluated by identification methods, sophisticated model matching methods [10], and simulation runs on the basis of detailled nonlinear distributed parameter models.

As the water level is mainly determined by the volume conditions of the reach, the measured value of the water level is filtered to damp all effects not caused by hydraulic volume conditions. We choose second order filtering whereas the filtering time constant is choosen so that waves alone do not trigger actuator action. Because of the damping via filtering we neglect the effects of wave propagation in the following design phases ($G_W(s) = 0$). For the usual case that the water level is controlled near the barrage we get:

$$G_{in}(s)=\frac{1}{As}e^{-sT_R}\quad,\quad G_{out}(s)=\frac{1}{As}$$

with the retention time constant T_R and the impounded water surface A.

The ideal solution for water level control can be realized by a feedforward structure: if the entering flow in a race can be measured or calculated by an observer, the effects of changing inflow to the water level can be considered by the following transfer function:

$$G_A(s)=\frac{H(s)}{Q(s)}=G_W(s)\cdot e^{-sT_R},\quad T_R=f(\Delta V/\Delta Q,Q^0),\quad T_L=f(\dot{Q},Q^0).$$

To damp inflow variations during its flow through the reach we approximate the pure time delay by

$$G_A(s)\approx\frac{\prod_{j=1}^{m}(1+sT_{Vj})}{\prod_{i=1}^{n}(1+sT_i)}e^{-sT_t}\quad \text{with } T_\Sigma=\sum_{i=1}^{n}T_i-\sum_{j=1}^{m}T_j+T_t\,.$$

If the sum of time constants equals the retention time constant, the volume conditions of the reach are fulfilled and as a consequence the water level remains constant (after transients). The best possible structure of $G_A(s)$ depends on the hydraulic conditions of the reach; we normally take m = 0, n = 2 and $T_1 = T_2 = T_t = T_R/3$.

As the retention time is not known exactly, the inflow can not be measured exactly, and because of unknown inflows and outflows within the reach, the water level will deviate from its reference value. For this purpose we add an water level controller of PI-type to the control scheme. By means of an analysis of the control structure we get the overall control function and are able to optimize the parameters of the PI-controller (e.g. by method of pole position or symmetrical optimum). Fig. 4 shows the structure of the resulting control loops. Many automated river reaches run successfully with the control structure described above.

But in various installations water level oszillation can be observed. A thorough analysis of the control loops reveals that these oscillations arise due to nonlinearities in the flow control loops. Thus, in a more detailled design step the flow control loops are modeled by a backslash nonlinearity. This model can be derived from the structure given in Fig.3 neglecting the hysteresis of the nonlinear control function [9]. By means of the theory of nonlinear control an easily implementable robust control measure can be found to ameliorate the control performance significantly: compensation of the backslash nonlinearity by the nonlinear function of a sensitivity zone. The width of the sensitivity zone is chosen and adapted in a way that the reaction of the level controller to small disturbances of the water level during regular operation is weak and strong vice versa. With this method continuous oscillations are eliminated and a calm reaction on water level disturbances is obtained during normal operation conditions with reduced wear of actuators and a good control performance. Caused by the nonlinearities it becomes difficult to design the control parameters by rules-of-thumb. That's why an iterative redesign step is introduced using siumlation on the basis of the lumped paramter model of the reach and of the control nonlinearity.

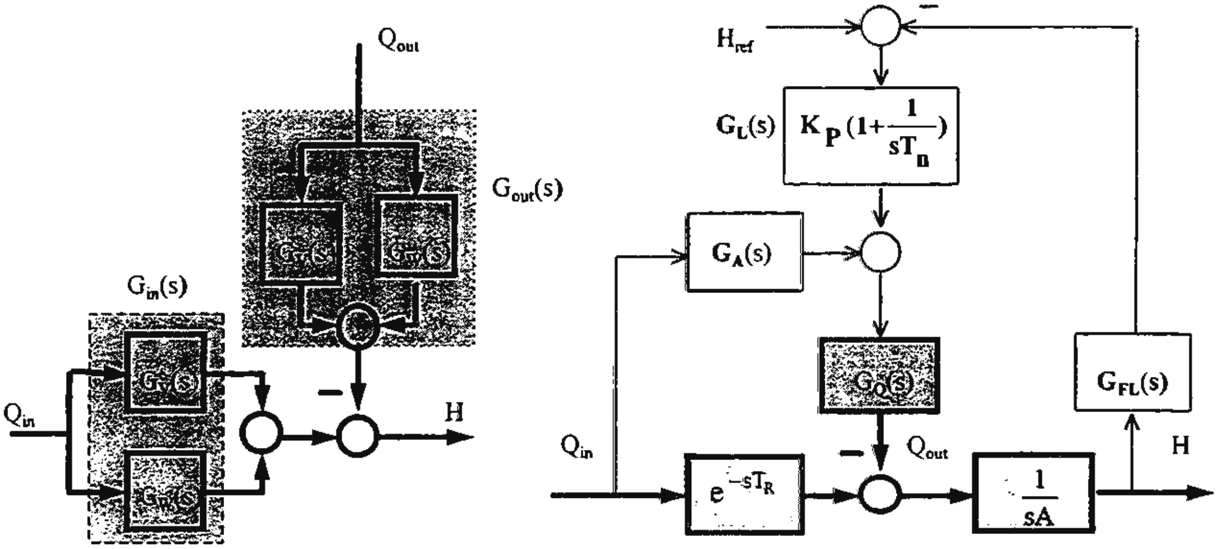


Fig. 4: Model structure and control loop with flow, antizipation, and water level control

The design of plants with high control requirements and/ or complex dynamics normally has to be based on simulation runs using detailled models. A one-dimensional parameter distributed hydraulic-numerical (HN)-model is used to represent the hydraulic process. The basic equations of the model are the classical equations of Saint Venant for 1-D open channel flow, the continuity equation expressing conservation fluid mass and the dynamic equation. The geometry of the landscape is being discretizised in form of cross-sections. At the boundary, where the hydro power plants and barrages are usually situated, unsteady boundary conditions of the discharge, of the water level as well as of the rating curves may be chosen. Special elements, like weir, gate, storage basin, bridge, siphon and looped or mashed systems, can be considered as well. Time (t) and place (x) are the independent variables in this equation system, the dependent variables are the discharge Q(x,t) and wetted area A(x,t) respectively water depth y(x,t). This equation system is solved by using the Preismann implicit method. The original control algorithms of the real industrial realization platform are integrated into the simulation model.

Even more advanced design steps use optimization methods to automatically optimize the time-variant parameters of the resulting nonlinear control structure on the basis of detailled simulation runs. The design process is supported by global and local optimization methods which use calculated performance indices of the simulated automated process as input and produce control parameters corrections as output [11]. The final parameter sets are achieved by numerous simulation runs. In this design phase the control performance is tested intensively under different operation conditions.

Application of the design procedure

The design procedure has been successfully applied during the installation of a new automation system at the hydro power plant of Bad Säckingen/ Rhine. An initial set of control parameters has been found by rules-of-thumb on the basis of the simple lumped parameter model. Using the simulation of the detailed nonlinear distributed parameter model the control parameters have been fine tuned. To calibrate the detailled HN-model field investigations at unsteady discharge were performed in the reach of Säckingen. At 14 stations the water levels in the reach were measured simultaneously, varied by waves generated at the hydro power plant. These results were found to be suitable to perform a numerical verification in order to test the HN-model. Previously the calibration of the model was performed based on fixed water levels made at constant discharges. The fact that the results of the investigations made in the field and in the model were nearly identical proves that the HN-model is suitable for quantifying the hydraulic behaviour. In a final design step parameter computer aided optimization has been used to find paramter sets which fulfill the high performance criteria of the customer. The simulated process was subjected to a variety of disturbances starting from different operation points. The nonlinear control is adapted to the various operating conditions by gain-scheduling. The simulation results as well as first experiences during field tests show that the new control system guarantees a very good performance of the power plant. Fig. 5 shows the controlled water level H during a simulation run over 24 hours. For a given inflow into the reach varying between 1.000 and 1.580 m^3/s the water level shows variations within a tolerance of 5 cm. A rapid rise of inflow of about 200 m^3/s is followed by a maximum peak of 7 cm.

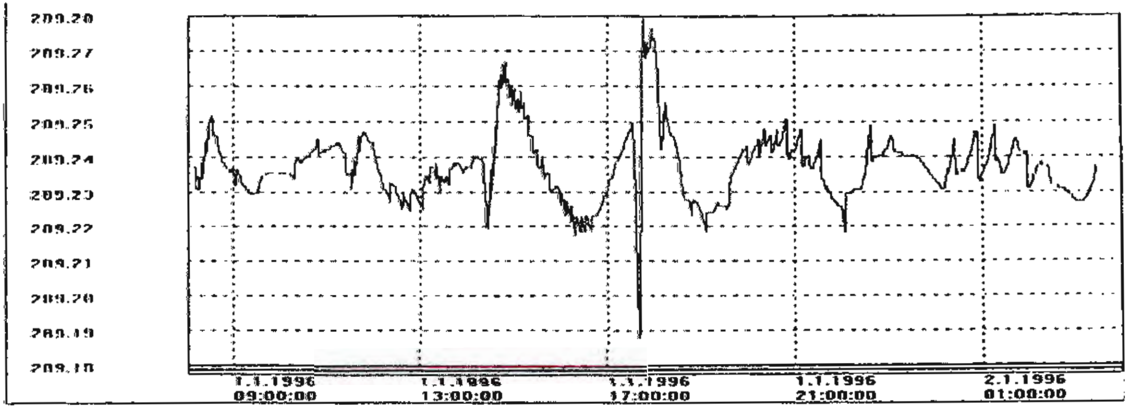


Fig. 5: Head water level of the power plant Bad Säckingen during strong discharge variations

Conclusion

A design procedure has been described which allows a cost effective design, realization, and operation of river reaches. It uses several classes of models, control algorithms and simulation methods to consider different classes of control requirements and process complexity. Because of its flexibility the available design toolbox is suitable for various processes and requirements of use. A case study shows good results. The further development of the design tools will concentrate on water and traffic management for cascades of river reaches using modern control technology as fuzzy and neural control.

References

[1] Neumüller, M. and Bernhauer, W.: Stauregelung und Abflußregelung von Laufwasserkraftwerken mit automatischen Verfahren. Wasserwirtschaft 66, Heft 9 (1976), pp. 253.

[2] Theobald, S. and Nestmann, F.: Control of a sequenence of barrages by numerical simulation. Proceedings of the Conference Modelling, Testing, and Monitoring for Hydro Power Plants, Budapest, Juli 1994.

[3] Dang van Mien, H. and Norman-Cyrot, D.: Nonlinear state affine identification methods: applications to electrical power plants. Automatica, Vol. 20 (1984).

[4] Dang van Mien, H. and Klein, F.: Robust control for hydraulic power plants. Proceedings First IEEE Conference on Control Applications, September 13-16, 1992, Dayton, Ohio.

[5] Detering, M., Langemeyer, A. and Kons, L.: Neue Möglichkeiten der Regelung von Laufwasserkraftwerken. Elektrizitätswirtschaft, Jg. (1996), Heft 14, pp. 946.

[6] Kochs, H.-D., König, D., Peterson, J. and Rogalla, M.: Einsatz wissensbasierter Techniken zur Teilautomatisierung des Schwellbetriebs von Laufwasserkraftwerken. Elektrizitätswirtschaft, Jg. 94 (1995), Heft 11, pp. 633.

[7] Cuno, B.: Ableitung mathematischer Modelle für Flußstauhaltungen. AEG, Frankfurt 1985.

[8] Cuno, B. and Kirchberg, K.-H.: Entwurf des Automatisierungssystems für eine Stauhaltungskette. AEG, Frankfurt 1985.

[9] Föllinger, O.: Nichtlineare Regelungen I: Grundlagen und harmonische Balance. R. Oldenbourg Verlag, München. 1993.

[10] Marenbach, P., Battenhausen, K.D. and Cuno, B.: Selbstorganisierende Generierung strukturierter Prozeßmodelle. Automatisierungstechnik at 43 (1995), pp. 277.

[11] Walcher, U.: Aufbau und Test eines Programmpakets zur simulationsgestützten Optimierung von Automatisierungsfunktionen für Flußsysteme. Diplomarbeit am Fachbereich Elektrotechnik der Fachhochschule Fuda, Fulda 1996.

THE MATHEMATICAL MODELLING OF HEAT GAIN IN BUILDINGS THROUGH TRANSPARENT INSULATION

D. Constales and R. Van Keer
University of Gent, Department of Mathematical Analysis
Galglaan 2, B-9000 Gent
E-mail: Denis.Constales@rug.ac.be, rvk@cage.rug.ac.be

Abstract. By using transparent insulation on the exterior of a wall, rather than an opaque one on its interior, the short-wave solar radiation is allowed to be absorbed at the exterior wall surface, so that the ensuing heat gains are conducted towards the building's interior.

To simulate the transient response of a transparently insulated wall to the varying meteorological and interior influences, a detailed model of both its optical and thermal behaviour has been realised.

The optical and thermal models are linked through the heat gains term in Fourier's equation and through the flux contribution at the absorbing layer, but also through the influence of temperature on the infra-red heat transport in the transparent insulation layers, which is modelled using a temperature-dependent conductivity.

The model described in this paper is used with the finite-element method to provide a simulation tool for the transient thermal behaviour of test cells and test buildings with transparent insulation, in order to study the rational energy potential of such constructions under the climatic and economic circumstances in Flanders.

Transparent insulation for residential buildings

Transparent insulation (TI) is an attempt to insulate without shutting out the short-wave solar gains; these are 'trapped' in the wall by an absorbing surface which lies *beyond* the insulation layer, so that the absorbed heat only has to cross zones of low thermal resistance in order to reach the inside and contribute to its heating.

There are various technical realisations of TI: aerogels, glass or plastic capillaries, honey-comb structures, etc. Suitable TI is of low thermal conductivity (for some structures, this also implies a low internal convectivity), low short-wave absorptivity and high long-wave absorptivity: thus the solar radiation can cross the TI without much attenuation and be absorbed, with little loss back to the outside of the infra-red radiation generated by the heated absorber. From the outside, a TI-wall is black; from the inside, it is completely closed.

In practice, TI proves useful during winter, for then there is a need for interior heating, and the peak values of solar radiation can still be considerable — e.g. the PASSYS measurements (cf. later) include solar radiation of $476\mathrm{W/m}^2$ on a sunny February day. The TI then reached an internal mass temperature of 65°C for an outside temperature of 1°C.

Indeed, the solar heat gains can be so considerable as to cause thermal deterioration to the TI under summer conditions, so a realistic TI wall must also include fail-safe shielding devices and automatic controls to operate it.

Even the simplest TI installations are quite expensive, since they are non-standard realisations and require protective external glazing. Consequently, TI would typically be installed on the South-facing wall of a building that has already been designed with excellent energy saving features; TI can then help improve these even further.

The VLIET project on TI

The research agency IWT[1] of the Flemish Government manages the VLIET[2], a group of research projects involving the Flemish Universities and industrial partners, and aimed at developing techniques of rational energy usage specifically adapted to the climate, economy etc. of Flanders.

One of these projects involves the construction and management of a test building situated on the campus of the KUL (Catholic University of Leuven), into which three slots have been provided for the

[1] Flemish Institute for the promotion of scientific and technological research in industry.

[2] Flemish Impulse Programme for Energy Technology

testing of TI: one holding a straightforward realisation, and the other two having a ventilated cavity in front of the absorber surface, one of these also having an extra layer of opaque insulation behind the absorber. The testing of TI is, of course, only one of many sets of experimental measurements performed this building.

Our project on TI is also funded by the VLIET, and is closely linked to this test building project: it is an interdisciplinary effort involving the Building Physics Lab at KUL and the Department of Mathematical Analysis at the RUG (University of Ghent), with the intention to perform a full evaluation of the rational energy usage potential of TI in Flanders.

This requires the completion of the following tasks:

1. the development of physical models for the heat and mass (e.g., hygric) transfer in TI, which should be general in order to describe all reasonable configurations of TI;
2. the mathematical expression of these models, using the finite element method (FEM);
3. the numerical integration of the resulting equations, using specifically developed and documented software;
4. the obtention of detailed relevant experimental data concerning the transient internal and external heat flow in TI, for climatic conditions prevailing in Flanders;
5. the validation and refinement of the models by comparison with the measurements – this stage implies a cycling through all of the previous ones, including the obtention of new experimental data;
6. the development of simplified models that can realistically be added into existing transient heat flow simulation packages such as ESP-r (cf. [4]) and MBDSA;
7. the performance of extensive simulations of TI applications to buildings situated in Flanders, with a detailed economic analysis of its impact.

This project started on June 1st, 1995. Delays in the planning permission of the test building have prevented us from gathering experimental data in the fall and winter of 1995, but we shall be able to obtain them in late 1996 and early 1997. We have worked on a preliminary basis with measurements that were obtained by the PASSYS project in 1991, cf. [1].

Modelling transient heat transfer in TI configurations

As our approach relies on the FEM, we can start directly from Fourier's equation for heat diffusion: we consider the equation

$$c\rho\frac{\partial u}{\partial t} = \operatorname{div}(k \operatorname{grad} u) + f(x,t),$$

in which u is the unknown temperature and f the volumetric heat gain (in our case, from the absorption of the incident short-wave solar radiation by the TI and the glass).

This equation must hold in all solid zones (glass plates, TI, wall), with suitable boundary conditions at the zone limits. Refer to Fig. 1 for the TI-module used by PASSYS.

In detail, this means:

1. that the heat flux $-k(\partial u/\partial x)$ at the outside glass surface must equal the net heat flux due to convection and radiation:

   $$F = F_c + F_r = h(T_o - u) + (F_o - \epsilon\sigma u^4),$$

   where T_o is the outside temperature, u the glass temperature at the limit, ϵ the long-wave emissivity, F_o the incident infra-red radiation flux (either measured or derived from meteorological data), and h the convective coefficient, which depends on the air velocity v, obeying e.g. an approximate empirical relation such as

   $$h = \begin{cases} 5.8 + 4v, & v \leq 5m/s, \\ 7.14v^{0.8}, & v > 5m/s, \end{cases}$$

in which v and h are expressed in the corresponding SI units, m/s resp. W/m^{2}K. Clearly, this boundary condition is non-linear and time-dependent.

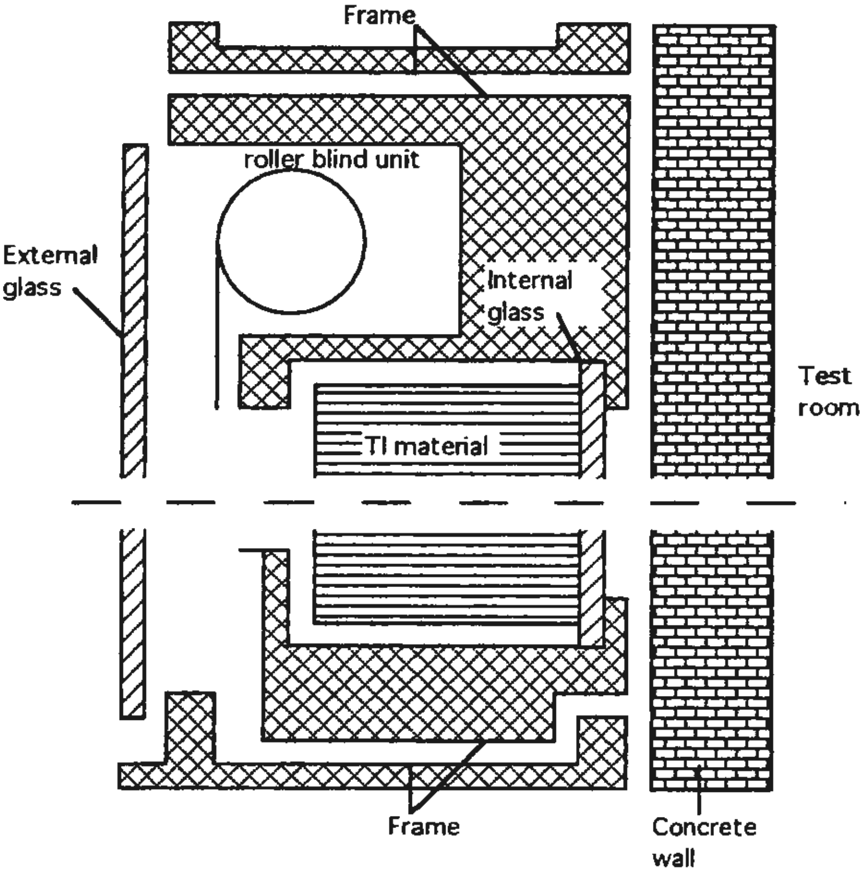


Fig. 1. Cross-section of the PASSYS TI-module.

2. Inside the glass zones, Fourier's equation holds with constant c, ρ and k, but the short-wave gain term f is time-dependent, since it is proportional to the incident solar radiation. To compute the proportionality factor, the solar radiation intensity must be determined at each point in the glass. Since the absorbing surface is not perfect, and reflections must also be taken into account, it consists of two components, the main one travelling to the right and another one travelling to the left. The effects of specular reflection and volumetric absorption can be conveniently expressed in matrix form: if I_l, I_r resp. I'_l, I'_r are the left- and rightbound radiation intensities before and after, specular reflection is expressed through the relationship

$$\begin{pmatrix} I_l \\ I_r \end{pmatrix} = \frac{1}{1-\rho} \begin{pmatrix} 1-2\rho & \rho \\ -\rho & 1 \end{pmatrix} \begin{pmatrix} I'_l \\ I'_r \end{pmatrix}$$

(this follows from energy conservation and from the definition of the reflection coefficient ρ). Similarly, volumetric absorption with differential absorptivity a is expressed by

$$\begin{pmatrix} I_l \\ I_r \end{pmatrix} = \begin{pmatrix} \exp(-aw) & 0 \\ 0 & \exp(aw) \end{pmatrix} \begin{pmatrix} I'_l \\ I'_r \end{pmatrix}.$$

At the absorbing layer the ratio between I'_r and I'_l is known — it's the layer's absorptivity — and at the outside surface I_r is known, being the measured incident solar radiation. These two conditions allow one to compute all I_l and I_r values, the solar gain contribution $f(x,t)$ in the glass and, with suitable parameter values being chosen, the same term in the TI.

The effective formulas are more complicated, though, because diffuse and direct light have to be taken into account separately, and the angle of incidence of the direct radiation influences the results.

The computations are therefore averaged over varying angles to account for the diffusivity, especially that of the TI, and carried out for a set of angles of incidence, so that interpolation can be used during the simulation runs.

3. In the air cavities, mixing is so fast that a homogeneous temperature u_a can be assumed. Its value can be obtained mathematically (cf. [9]), or be computed as an extra unknown temperature from the relationship

$$wc_a(u_a)\rho_a(u_a)\frac{du_a}{dt} = F_{a,l} - F_{a,r},$$

where the $F_{a,*}$ are the flux values at the left, resp. right side of the air layer. These fluxes themselves are of convective origin, so $F_{a,l} = h(u_l - u_a)$ resp. $F_{a,r} = h(u_a - u_r)$, where $h = (k/w)Nu$, and Nu is the Nusselt number for the air cavity; it is always at least 1, since h comprises the conductive value k/w and a supplementary advective term that may be neglected for sufficiently thin cavities. An approximate empirical formula is e.g.

$$Nu = \begin{cases} 1, & Ra \leq 1000 \\ 1560w^2\left(\frac{|\Delta u|}{L}\right)^{1/4}, & 1000 \leq Ra \leq 10^6, \end{cases}$$

where Ra is the Rayleigh number, and the occuring values should be expressed in SI units.

4. Across the internal air cavities there is also a radiative heat exchange. If ϵ_l, resp. ϵ_r are the emissivities and α_l, resp. α_r the absorptivities, and all reflections are taken into account, the radiative flux to the right is given by

$$F_r = \frac{\epsilon_l \sigma u_l^4 + (1-\alpha_l)\epsilon_r \sigma u_r^4}{1-(1-\alpha_l)(1-\alpha_r)},$$

with a similar formula for the flux to the left obtainable by exchanging all indices l and r; σ is Stefan's constant, and all temperatures are expressed using the Kelvin scale. The net radiative transfer is then given by the difference of the fluxes:

$$\frac{\alpha_l \epsilon_r u_r^4 - \alpha_r \epsilon_l u_l^4}{1-(1-\alpha_l)(1-\alpha_r)}$$

from the right surface to the left.

5. We must also write out Fourier's heat equation for the TI. The solar gain term $f(x,t)$ is computed as for the glass zones, but now the long-wave radiative transfer inside the TI must also be expressed. This is difficult to deduce theoretically, as it depends on the structure of the TI, which is highly anisotropic for capillaries and honey-comb structures. Consequently, approximate empirical relationships must be introduced.

We use an approximation in which the conductivity k is supposed to depend linearly on the temperature u:

$$k = k_0 + k_1 u,$$

where the values k_0 and k_1 (which strongly depend on the TI's type and thickness) are obtained experimentally, e.g. by placing the TI in a guarded hot plate device[3] to measure the conductivity at different temperatures, and interpolating a least-squares line on the results. For the PASSYS test cell's polycarbonate honey-comb TI the relevant measurements are mentioned in [5].

The heat equation then adopts the following form:

$$c\rho\frac{\partial u}{\partial t} = \mathrm{div}\,(k_0 + k_1 u)\,\mathrm{grad}\, u + f(x,t).$$

[3] cf. also the ASTM C-177 standard and the Belgian norm NBN B.62-203.

Clearly, under the current assumptions knowledge of u is equivalent to knowledge of the u-variable conductivity k; switching to k as the unknown simplifies the equation to

$$c\rho\frac{\partial k}{\partial t} = \operatorname{div}\operatorname{grad} k^2 + k_1 f(x,t),$$

which belongs to a class of non-linear differential equations discussed e.g. in [2], and for which the heuristic approach to stability conditions often seems to be verified (cf. also [3]).

6. At the inner surface, there is again a convective and radiative boundary condition, much as at the outside, but the convective coefficient h is no longer time-dependent; it just corresponds to the convection into the test room. Since this room is quite deep (5m) and has no forced internal air movement, a value of $h = 5.8\mathrm{W/m^2K}$ seems appropriate. For the radiative term we assume that the test room is a blackbody at its air temperature. Then the test room air temperature can be obtained by integrating the incoming flux.
7. Whenever two solid layers meet, the boundary conditions are those of perfect thermal contact, i.e. continuity of the temperature and of the heat flux.

Summarising, one obtains a set of partial differential equations, some of which are non-linear, which are coupled through non-linear boundary conditions. The FEM is very convenient when it comes to express these conditions, since the boundary values of the flux appear quite naturally through a partial integration during the process of transforming the heat equation; also, perfect thermal contact is expressed by having a single value represent the temperature immediately left and right to the boundary between the contacted layers, and the condition on the flux leads to a convenient cancellation out of terms.

Software implementation and first results

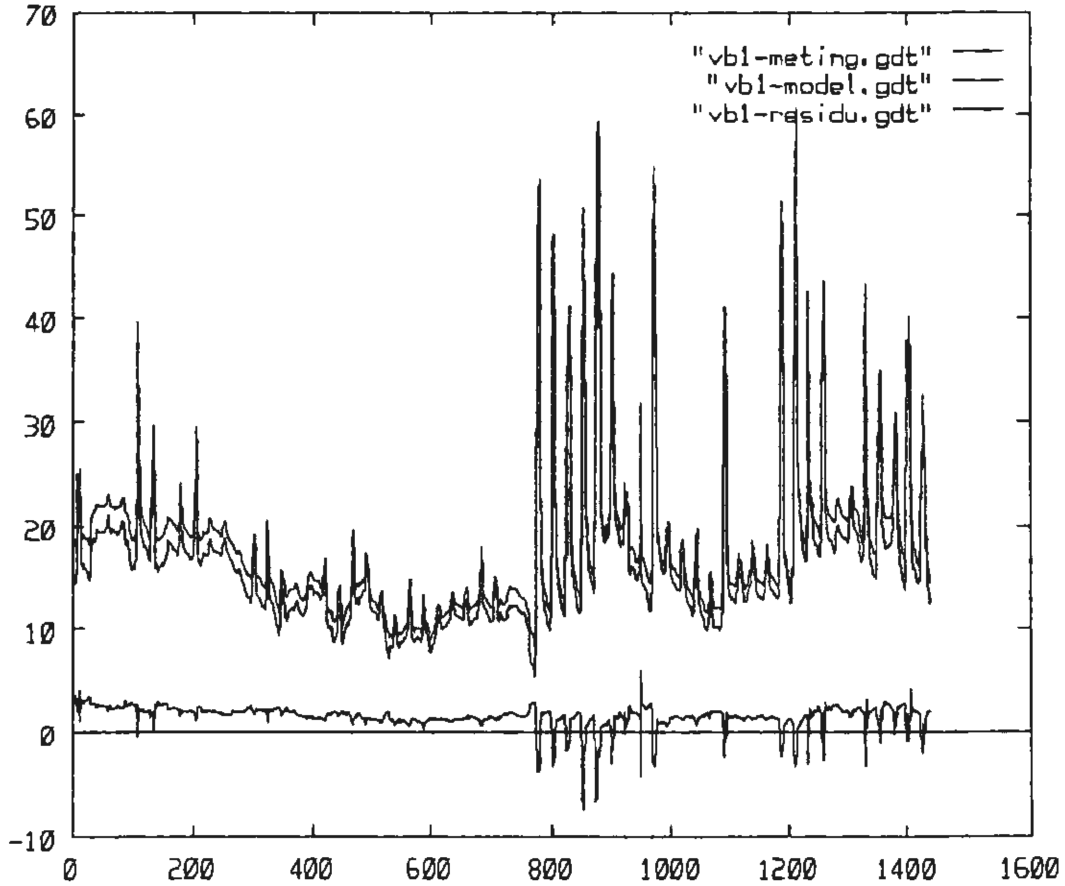


Fig. 2. Mass temperature vs. time (in hours) in the middle of a TI-component.

The validation phase of the model has started, and Fig. 2 provides a comparison between the predicted values and the measurements of the mass temperature in the middle of a TI component — the simulation of this temperature is important because overheating the TI may damage it seriously.

The models for TI are implemented using a software toolbox which we have built on top of the public-domain library meschach (cf. [8]) for dense, band and sparse matrix arithmetic. For ease of portability, everything is being written in ANSI GNU C, and compiles without modification on hardware platforms reaching from Intel 80386 Linux to SparcStations. The code is extensively documented using the literate programming tool noweb (cf. [7]).

Summary

This finite-element model for the transient thermal behaviour of transparent insulation provides an efficient means for extensive simulation of general configurations, and matches the experimental measurements to a level of accuracy that will be further improved when more extensive data has been gathered from out own test building.

References

[1] The PASSYS Services Summary Report. European Commission, Directorate General XII for Sciences, Research and Development, 1992.

[2] Ames, W.F., Nonlinear Partial Differential Equations in Engineering. Academic Press, New York, Vol. I, 1965; Vol. II, 1972.

[3] Ames, W.F., Numerical Methods for Partial Differential Equations, 3rd ed. Academic Press, 1992.

[4] Clarke, J.A., Energy Simulation in Building Design. Adam Hilger Ltd., Bristol and Boston, 1985.

[5] Guy, A. (ed.), PASSYS Test Components Descriptions. European Commission, Directorate General XII for Sciences, Research and Development, 1991.

[6] Incropera, F.P. and De Witt, D.P., Fundamentals of Heat and Mass Transfer, 3rd. ed. Wiley, 1990.

[7] Ramsey, N., The noweb Hacker's Guide. Department of Computer Science, Princeton University, 1994.

[8] Stewart, D.E. and Leyk, Z., Meschach: Matrix Computations in C. In: Proceedings of the Centre for Mathematics and its Applications, The Australian National University, 32 (1994).

[9] Van Keer, R. and Handlovicová, A., On a Mathematical Model for the Heat Transmission through Transparent Isolation Materials in Buildings. Math. Modelling of Systems, 1, No. 1 (1995), 127–137.

PARALLEL SIMULATION OF HEAT TRANSFER PROCESSES

M. Holzinger and F. Breitenecker
Technical University Vienna, Wiedner Hauptstr. 8-10, A-1040 Wien
mholz@osiris.tuwien.ac.at

Abstract. The purpose of this paper is to investigate achievable speedups regarding continuous simulation times using parallel model structures and the language PVM-FORTRAN in computation instead of conventional, sequential procedures. This will be demonstrated by studying a particular subject — the heating of a room including and the surrounding walls and the outer region. Parallelization is done by splitting the physical space into two ranges of space-integration which results in necessary data exchange between the submodels.

1 Introduction

We consider the temperature distribution within a finite range as a scalar field

$$u(x,y,z,t): \mathrm{M} \subseteq \mathbf{R}^4 \to \mathbf{R}$$

For the inner region which is filled with air, we assume

$$\frac{\partial u}{\partial t} + z \cdot \sqrt[3]{z} \cdot \left(v_x \cdot \frac{\partial u}{\partial x} + v_y \cdot \frac{\partial u}{\partial y} + v_z \cdot \frac{\partial u}{\partial z} \right) = \tau_L \cdot \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right]$$

to describe the variation of the temperature field. Note that the right side of the equation contains the Laplacian operator to describe heat conduction (with a factor which represents the thermal conductivity of air), whereas the term on the left side describes heat transfers caused by convection — the velocity components are held constant and the vertical temperature distribution depends on height (the so-called "Priestley approach"). It should be clear that the model presented in this report is not able to describe all physical phenomena, for instance turbulent flows.

As for the walls, the right term on the left side disappears because of the lack of conductive streams in this media. In this case we obtain the ordinary instationary heat conduction equation with a modified temperature coefficient. The whole system can on the one hand be regarded as an initial/bounded value problem with the external temperature given. On the other hand, it seems merely natural to separate the problem into two bounded value problems (room/walls) with necessary data exchange on their common borders.

2 Parallel Aspects

If we have got several processors at our disposal and want to make optimal use of their capacity we should first have a look at the parallel structures of a model. To measure achievable accelerations in continuous simulation, we define *speedup*

$$S_n = \frac{\text{calculation time on 1 processor}}{\text{calculation time on } n \text{ processors}}$$

and *parallel efficiency*

$$E_n = \frac{S_n}{\text{number } n \text{ of processors}}$$

Regarding distributed memory models, we now give a classification of parallel structures - our problem fits the second of the following three groups of parallelisms:

1. Model independent parallelisms: for example the repeated execution of a single program such as given in the variation of initial values in an initial value problem. Each processor is provided with the whole set of data and communication amount is very low. Problems of this type can be parallelized very efficiently.

3. Algorithmic parallelisms: we try to split the source-code and distribute parts of the whole algorithm. In this case, the data flow between the processors is very high and one has to be clear that in worst case the calculation time exceeds the sequential solution.

3 Discretization and Implementation

The local partial derivatives were approximated by finite differences [4]; that guarantees error terms of second order for the second central quotients. As for the velocity gradient we obtain only first order because we did not use central differences but mean values of the forward and backward quotients. This choice was determined by considerations of symmetry. Thereby we get a system of ordinary differential equations that can be solved by time integration using Runge-Kutta methods [1].

The model was first implemented in the conventional, sequential way using FORTRAN77 and then parallelized. Here we made use of PVM-FORTRAN [3] that provides means of communication. The program structures usually contain a master/slave scheme where one processor has a control function and the other ones calculate their jobs. Figures 1&2 show the parallel and sequential structures with the necessary data exchange.

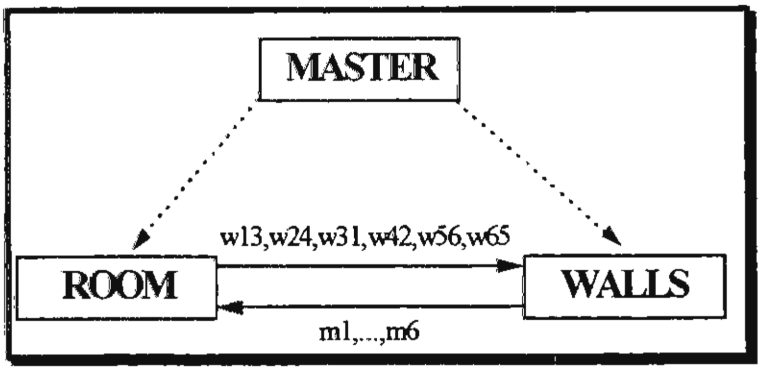


Fig. 1. Geometric-parallel structures and data exchange.

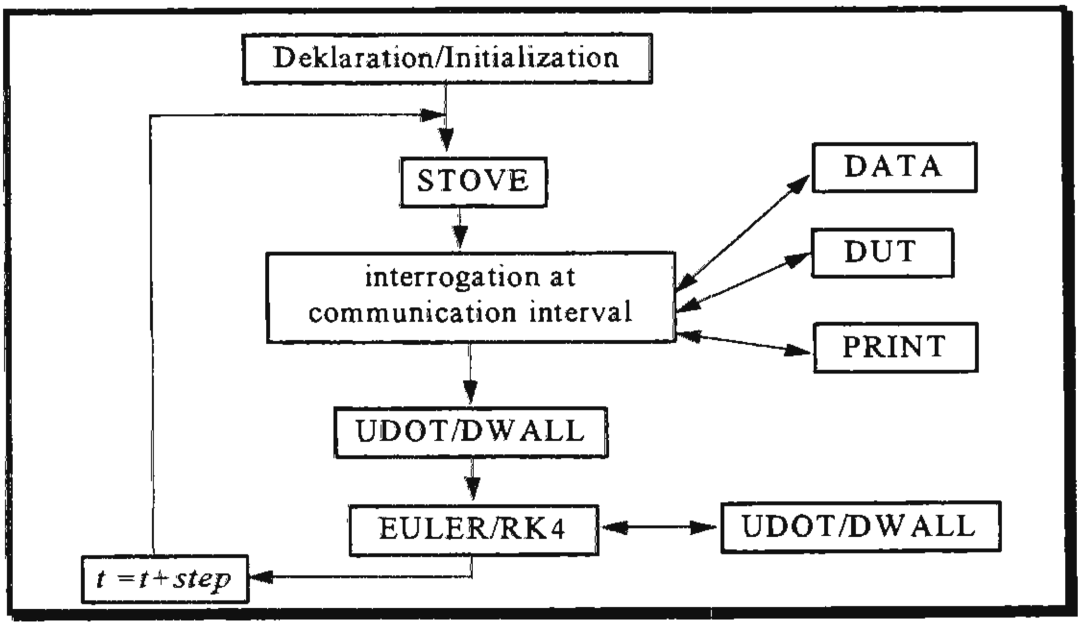

Deklaration/Initialization
STOVE
DATA
DUT
interrogation at communication interval
PRINT
UDOT/DWALL
EULER/RK4
UDOT/DWALL
t =t+step

Fig. 2. The FORTRAN77-structures

4 Results

The first task is to provide convergence and numerical conformity of both the sequential and parallel solutions. This was done by using about 150.000 state variables for discretization and comparing the results. The simulation runs took place on two DEC 5000-workstations (33 MHz and 20 MHz) connected by TCP/IP–Ethernet where the faster machine calculated the room. The communication amount was found not to be too extensive. Table 1 shows calculation time, speedup and efficiency.

system	Time (min)	S_n	E_n
DEC 5000/133, sequential	2020	1	1
DEC 5000/20, sequential	3240	1	1
PVM, parallel	1590	1,27	0,64

Table 1. Parallel and sequential simulation times.

Finally some case studies concerning the position of the stove were carried out. The figures 3 and 4 show vertical distributions above the stove and that best temperature distributions are obtained by placing the stove in the center of the room.

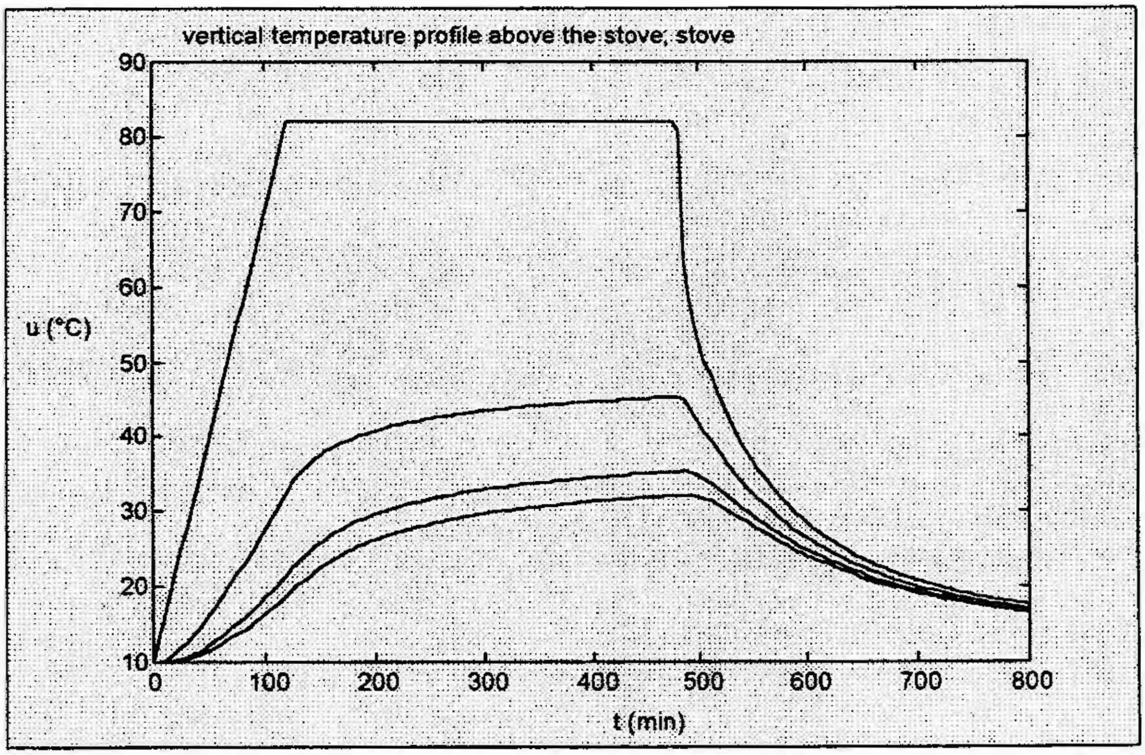


Fig. 3. Obtained temperature distribution.

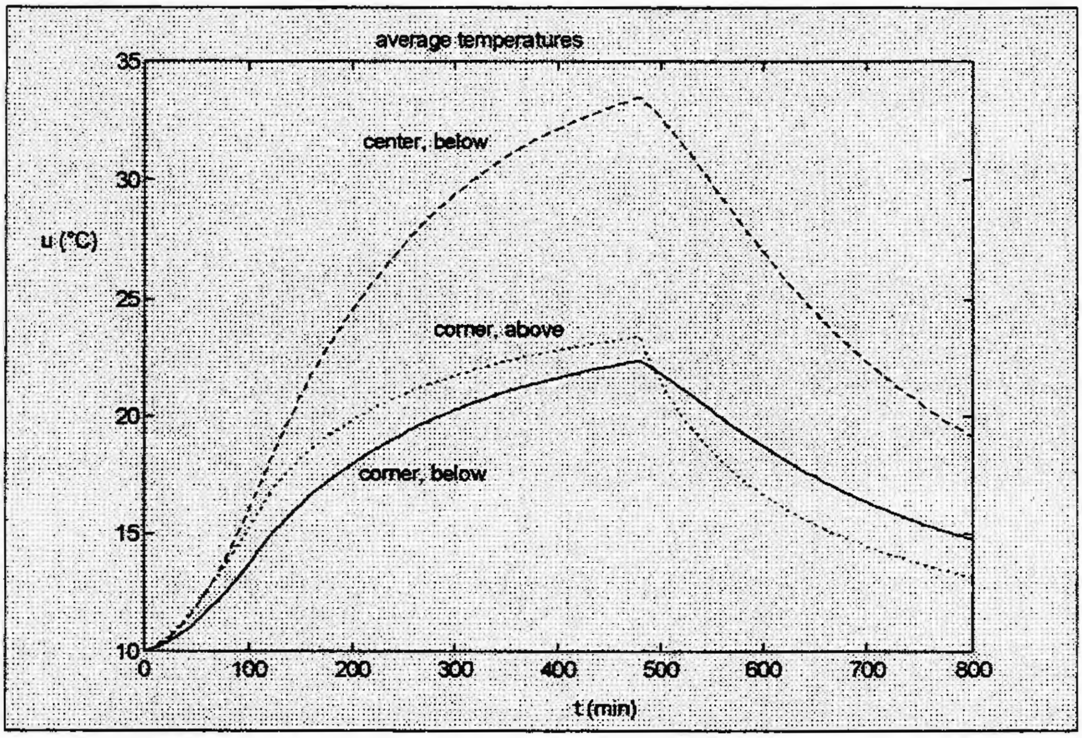


Fig. 4. Case study stove-position.

5 Summary

To achieve higher efficiency, it seems to be necessary to develop further partitions of the room. Each of these geometric ranges can then be calculated with different machines. This attempt seems not to be as trivial: the existence of a source of heat causes a non-homogeneity in respect of the temperature gradient. Hence one has to split the room in symmetric sectors around the stove where equal conditions are being guaranteed.

We developed and examined a rather simple model. Nevertheless, the amount of detecting and implementing the parallel sub-models was very high, same the probability of program incorrectness. As for larger models, it seems possible that parallel developing time takes longer than one or even more processor generations.

6 References

1. Breitenecker F., Ecker H., Bausch-Gall I.: Fortschritte in der Simulationstechnik: Simulieren mit ACSL. Eine Einführung in die Modellbildung, numerischen Methoden und Simulation. Vieweg-Verlag, Braunschweig/Wiesbaden 1993, 140-152.

2. Breitenecker F., Schuster G., Husinsky I., Fritscher J.: Parallelization of simulation tasks: Methodology - Implementation - Application, Proc. First International Conference of the Austrian Center of Parallel Computation. Springer-Verlag, Salzburg 1991.

3. Geist A., et alteri: PVM 3 User's Guide and Reference Manual. Oak Ridge National Laboratory, Oak Ridge 1993.

4. Holzinger, M.: Modellorientierte Parallelisierung von Modellen mit verteilten Parametern (master thesis). See also: http://ws3.atv.tuwien.ac.at/~mholz

CONTINUUM MODELLING OF LARGE NETWORKS IN DOMAINS

J. Gwinner
Institute of Mathematics, Department of Aerospace Engineering
University of the Federal Army Munich, D - 85577 Neubiberg

Abstract. This note presents a continuum model of large discrete networks in planar domains. For this model, the Kirchhoff law and capacity constraints lead in a system optimization approach to a infinite dimensional constrained optimization problem and to "mixed" variational inequalities. Mixed finite element methods are extended to these variational inequalities such that computable discretizations of the continuum problem are obtained. Finally the finite element approximations are interpreted in a suitably constructed discrete network.

Introduction

This note presents a continuum model of large discrete networks. Similar to [1, 2, 10, 17], we embed the network in a bounded planar domain and describe the unknown flow at each point by a vector field. As in classical network theory [5], the unknown flow has to satisfy capacity constraints and the Kirchhoff law (conservation of mass) modelled by a div constraint. Here in contrast to [1, 2, 10, 17], we adapt some concepts of functional analysis (up to now applied to the Stokes problem in fluid mechanics, see e.g. [6]) to make precise the understanding of a feasible flow, respectively of an optimal flow, where employing a system optimization approach, our continuum model leads to a infinite dimensional constrained optimization problem. Further by introduction of the Lagrangian associated to the div constraint we arrive at novel "mixed" variational inequalities. Then mixed finite element methods are extended to these variational inequalities such that computable discretizations of the continuum problem are obtained. Finally starting from a finite element triangulation we construct a connected discrete graph such that the approximate finite element flows can be interpreted as approximate edge flows in a discrete network.

The Continuum Model

In this section we develop our continuum model of large discrete networks. We concentrate on the system optimization approach. However as we point out at the end of this section, there is a formal similarity in the structure of the resulting variational problems arising from the system optimization approach and from the user optimization approach. Both lead to "mixed" variational inequalities in the unknown flow and an additional Lagrange multiplier which lives in a convex subset that is simply given by unilateral constraints only.

To begin with, we embed a given large dense (planar) network, e.g. a urban street network, in a large enough (simply connected) bounded domain Ω in $\mathbf{R}^2$. In our continuum (transportation) model as in [1, 2, 10, 17], the unknown flow at each point $\underline{x}$ of Ω is described by a vector field $\underline{u}(\underline{x})$, whose components u_1 and u_2, e.g. in the traffic network, represent the traffic density (number of vehicles per unit length and unit time) through a neighborhood of $\underline{x}$ in the direction x_1 and x_2, respectively. Thus the flow crossing some path $\mathcal{K}$ in Ω is given by the line integral

$$\int_{\mathcal{K}} \underline{u} \cdot d\underline{x} .$$

We have to satisfy the conservation law (Kirchhoff law), modelled by the constraint

$$\operatorname{div} \underline{u}(\underline{x}) = 0 \qquad (\underline{x} \in \Omega).$$

More generally one can prescribe a scalar field $\rho(\underline{x})$ (number of vehicles per unit area and unit time) and demand

$$\operatorname{div} \underline{u}(\underline{x}) = \rho(\underline{x}) \qquad (\underline{x} \in \Omega).$$

Then obviously, $\rho(\underline{x})$ will be positive if net flow is generated in a neighborhood of $\underline{x}$, respectively negative if net flow is absorbed there. However, this more general case can be reduced to the homogeneous case $\rho = 0$ by the solution of an appropriate auxiliary problem: Solve the Poisson problem $\Delta\phi = \rho$ with te same boundary conditions as the original problem (see later) and replace the flow $\underline{u}$ by $\underline{\tilde{u}} = \underline{u} - \nabla\phi$.

In contrast to [1, 2, 10, 17] we make more precise the function analytic setting. The most simple (in contrast to [14, 16] and also different from [11, 12]) and – in view of the div constraint – the natural function space is the Hilbert Space

$$\mathcal{E}(\Omega) = \{\underline{v} \in L^2(\Omega, \mathbf{R}^2) |\, \operatorname{div} \underline{v} \in L^2(\Omega)\}$$

on the Lipschitz domain Ω, equipped with the scalar product

$$\langle \underline{v}, \underline{w} \rangle = \langle v_1, w_1 \rangle_{L^2(\Omega)} + \langle v_2, w_2 \rangle_{L^2(\Omega)} + \langle \operatorname{div} \underline{v},\ \operatorname{div} \underline{w} \rangle_{L^2(\Omega)},$$

a well-known space in fluid mechanics [6]. This necessitates that all equations and inequalities on Ω are to be understood almost everywhere in what follows.

Further we have capacity constraints. Instead of implicit set constraints in [17] we introduce a subdomain ω of Ω, vector fields $\underline{s}, \underline{t} \in L^2(\omega, \mathbf{R}^2)$ such that $\underline{s} \leq \underline{t}$ componentwise and impose the unilateral constraints

$$\underline{s} \leq \underline{u} \leq \underline{t} \quad \text{in } \omega.$$

This includes the nonnegativity constraint $\underline{u} \geq \underline{0}$ in [2, 12]. On the other hand, one can imagine more general constraints, e.g. constraints on the components separately in different subdomains. However since these more general constraints do not lead to more insight, we do not elaborate on these further here.

Finally we have boundary conditions on the boundary $\Gamma = \partial\Omega$. We consider the most simple case of a closed system, with no traffic across Γ. This leads to the function space

$$\mathcal{E}_0 := \overline{\mathcal{D}(\Omega)}^{\|\cdot\|},$$

which is the closure of the space $\mathcal{D}(\Omega)$ with respect to the Hilbert norm $\|\underline{v}\| = \langle \underline{v}, \underline{v} \rangle^{1/2}$ and where $\mathcal{D}(\Omega)$ consists of all infinitely differentiable vector fields $\underline{v} : \Omega \to \mathbf{R}^2$ with compact support.

More generally one could also fix some scalar field τ on the boundary Γ and prescribe the boundary flux along the normal $\underline{\nu}$ by $\underline{u} \cdot \underline{\nu} | \Gamma = \tau$. In virtue of the Green-Stokes formula (see [3] Part A §1 Theorem 1), this boundary condition can formulated for flows $\underline{u} \in \mathcal{E}(\Omega)$ and boundary data $\tau \in H^{-\frac{1}{2}}(\Gamma)$. However in the context of urban traffic, such a distributed in/out flow across the boundary does not appear very realistic. If one considers traffic entering or leaving the city via district roads or highways, one is led to consider given point measures on the boundary, but this makes the mathematical model more complicated.

To sum up, all feasible flows $\underline{u}$ are contained in the set

$$K := \{\underline{u} \in \mathcal{E}_0(\Omega) |\, \underline{s} \leq \underline{u} \leq \underline{t} \text{ in } \omega,\ \operatorname{div} \underline{u} = 0 \text{ in } \Omega\}$$

which is convex and closed. In [7] the feasibility problem is studied and equivalent conditions on the data for the nonemptiness of K are given. Here we simply assume, $K \neq \emptyset$. The set K ist generally infinite dimensional, as the example in section 3 of [7] shows.

Therefore to single out a particular flow, we employ an optimization procedure. In the system optimization model one supposes that there is a planning authority (or society) that has a clearly defined target for the network considered, e.g. an "ideal" traffic pattern seeking for a best approximate "real", that is feasible flow, or the target of reduction of costs (e.g. transportation costs) seeking for a feasible flow that minimizes a given cost function. Thus one obtains an optimization problem of the form

$$(P) \begin{cases} \text{minimize } F(\underline{u}) := \int\limits_{\Omega} f(\underline{x}, \underline{u}(\underline{x}))\, d\underline{x} \\ \text{subject to } \underline{u} \in K. \end{cases}$$

Under some appropriate structural assumptions on the integrand f one can prove the existence of an unique minimizer ([7] Theorem 4.1) . Further let us introduce the Lagrangian

$$\mathcal{L}(\underline{u}, \eta) = F(\underline{u}) + \langle \operatorname{div} \underline{u}, \eta \rangle_{L^2(\Omega)}, \quad u \in U, \eta \in L^2(\Omega),$$

where U is simply given by box constraints,

$$U := \{\underline{u} \in \mathcal{E}_0(\Omega) |\, \underline{s} \leq \underline{u} \leq \underline{t} \text{ in } \omega\}\,.$$

Then appropriate structural assumptions permit to obtain the saddle point problem

$$(P^I) \text{ Find } (\hat{\underline{u}}, \hat{\eta}) \in U \times L^2(\Omega) \text{ such that } \begin{cases} F'(\hat{\underline{u}}, \underline{u} - \hat{\underline{u}}) + \langle \operatorname{div}(\underline{u} - \hat{\underline{u}}), \hat{\eta} \rangle_{L^2(\Omega)} & \geq \ 0 \quad \forall\, \underline{u} \in U\,; \\ \langle \operatorname{div} \hat{u}, \eta \rangle_{L^2(\Omega)} & = \ 0 \quad \forall\, \eta \in L^2(\Omega)\,. \end{cases}$$

There is another formulation, where at the cost of increasing the required regularity of the multiplier we can weaken the regularity of the primal variable and replace $\mathcal{E}_0(\Omega)$ by a simpler space. Namely by the Green-Stokes formula ([3] Part A §1 Theorem 1 ; [6] Theorem 2.2), we have

$$\langle \operatorname{div} \underline{u}, \varphi \rangle_{L^2(\Omega)} + \langle \underline{u}, \nabla\varphi \rangle_{L^2(\Omega, \mathbb{R}^n)} = 0 \quad \forall\, \underline{u} \in \mathcal{E}_0(\Omega), \varphi \in H^1(\Omega).$$

Put in another way, -div and ∇ are adjoint operators. Therefore let us introduce (here $n = 2$)

$$W := \{\underline{w} \in L^2(\Omega, \mathbb{R}^n) |\, \underline{s} \leq \underline{w} \leq \underline{t} \text{ in } \omega\}$$

and we obtain the following saddle point problem:

$$(P^{II}) \text{ Find } (\hat{\underline{w}}, \hat{\varphi}) \in W \times H^1(\Omega) \text{ such that } \begin{cases} F'(\hat{\underline{w}}, \underline{w} - \hat{\underline{w}}) + \langle \underline{w} - \hat{\underline{w}}, \nabla\hat{\varphi} \rangle & \geq \ 0 \quad \forall\, \underline{w} \in W\,; \\ \langle \hat{\underline{w}}, \nabla\varphi \rangle & = \ 0 \quad \forall\, \varphi \in H^1(\Omega)\,. \end{cases}$$

Both problems (P^I) and (P^{II}) can be formulated as a single variational inequality; in particular in the case of a quadratic function F we arrive at bilinear forms in the variational inequality. It turns out that also the user optimization model leads also to such variational inequalities on convex sets defined by simple constraints; see [2] for a heuristic discussion and [8] for a derivation of a finite dimensional variational inequality from a constrained market equilibrium model of supply and demand in a bipartite graph.

Discretization of the Continuum Model

In this section we describe the discretization of our continuum model by the finite element method.

Let us use the most elementary setting for finite element methods: Ω and ω are supposed to be (bounded, open) polyhedral subsets of $\mathbb{R}^2$; $\mathcal{T}_h\,(\supset \tilde{\mathcal{T}}_h)$ denotes a triangulation of $\overline{\Omega}\,(\supset \overline{\omega})$ by triangles T of diameter h_T less than h such that

$$\overline{\Omega} = \bigcup_{T \in \mathcal{T}_h} T, \quad \overline{\omega} = \bigcup_{T \in \tilde{\mathcal{T}}_h} T.$$

Moreover, the family of triangulations $\{\mathcal{T}_h\}_{h>0}$ is supposed to be regular, that is, with ρ_T the radius of the circle inscribed in T there holds

$$\inf_{h>0} \min_{T \in \mathcal{T}_h} \frac{\rho_T}{h} > 0.$$

The direct approach to the discretization of the problem (P) would be the construction of a finite dimensional subset K_h of K by polynomial interpolation on each triangle T (this is a successful procedure in the case of scalar unconstrained variational problems like the Dirichlet problem for the Poisson equation) and to seek a minimizer $\hat{\underline{u}}_h \in K_h$ of F (or of an appropriate approximation F_h by a quadrature method) on K_h. Leaving aside the unilateral constraints $\underline{s} \leq \underline{u}_h \leq \underline{t}$, it is difficult to construct a basis of a finite dimensional subspace V_h of the subspace

$$V = \{\underline{u} \in \mathcal{E}_0(\Omega) |\, \operatorname{div} \underline{u} = 0 \text{ in } \Omega\}\,.$$

Indeed, any $\underline{v}_h \in V_h$ should satisfy the two constraints:

(i) Within each triangle $T \in \mathcal{T}_h$, the divergence of $\underline{v}_h$ must vanish; i.e. div $\underline{v}_h | T \equiv 0$, thus in particular

$$\int_{\partial T} \underline{v}_h \cdot \underline{\nu}_T \, ds = 0.$$

(ii) The normal traces of $\underline{v}_h$ must be "continuous across the interface" between any two neighboring triangles T_1 and T_2 with common edge, i.e. $\underline{v}_h \cdot \underline{\nu}_{T_1}$ and $-\underline{v}_h \cdot \underline{\nu}_{T_2}$ must coincide on $T_1 \cap T_2$ (see [15] Theorem 1.3).

To circumvent this difficulty we use the Lagrangian formulations (P^I) and (P^{II}). Then we can adopt the construction of finite dimensional subspaces that is known from the numerical analysis of mixed finite element methods; for an exposition see section 6 of [15].

First for the "dual" mixed formulation (P^I), to obtain a finite dimensional subspace E_h of $\mathcal{E}(\Omega)$, we can choose the Raviart–Thomas elements $D_k(T)$ $(T \in \mathcal{T}_h)$ as the space of restrictions to T of the functions in $D_k := \mathcal{P}^2_{k-1} \bigoplus \underline{x}\mathcal{P}_{k-1}$ $(k \in \mathbb{N})$, where $\mathcal{P}_{k-1}$ denotes the space of polynomials (here in 2 variables) of degree $\leq k-1$. This leads to the nonconforming approximation

$$U_h = \{\underline{u}_h \in E_h |\, \underline{s}(\underline{x}) \leq \underline{u}_h(\underline{x}) \leq \underline{t}(\underline{x}),\ \forall\, \underline{x} \in \tilde{\mathcal{N}}_h\},$$

where $\tilde{\mathcal{N}}_h$ denotes the finite point set (depending upon the degree k) associated to the triangles $T \in \tilde{\mathcal{T}}_h$. The corresponding finite dimensional subspace of the multiplier space $L^2(\Omega)$ is then

$$M_h^I = \{\eta_h \in L^2(\Omega) |\, \eta_h|T \in \mathcal{P}_{k-1},\ \forall\, T \in \mathcal{T}_h\}\,.$$

Thus neglecting numerical integration, we obtain the following finite dimensional approximate problem (P_h^I) : Find $(\underline{\hat{u}}_h, \hat{\eta}_h) \in U_h \times M_h^I$ such that

$$\left\{ \begin{array}{lll} F'(\underline{\hat{u}}_h, \underline{u}_h - \underline{\hat{u}}_h) + \sum_{T\in\mathcal{T}_h} \langle \text{div } (\underline{u}_h - \underline{\hat{u}}_h), \hat{\eta}_h\rangle_{L^2(T)} & \geq 0 & \forall\, \underline{u}_h \in U_h\,, \\ \langle \text{div } \underline{\hat{u}}_h, \eta_h\rangle_{L^2(\Omega)} & = 0 & \forall\, \eta_h \in M_h^I\,. \end{array} \right.$$

Further for the "primal" mixed formulation (P^{II}), we can simply choose as ansatz for $k \in \mathbb{N}$ the sets

$$\begin{array}{lll} W_h & = & \{\underline{w}_h \in L^2(\Omega, \mathbb{R}^2) |\, \underline{w}_h|T \in \mathcal{P}^2_k,\ \forall\, T \in \mathcal{T}_h\,;\, \underline{s}(\underline{x}) \leq \underline{w}_h(\underline{x}) \leq \underline{t}(\underline{x}),\ \forall\, \underline{x} \in \tilde{\mathcal{N}}_h\}\,, \\ M_h^{II} & = & \{\varphi_h \in H^1(\Omega) |\, \varphi_h|T \in \mathcal{P}_{k+1},\ \forall\, T \in \mathcal{T}_h\}\,. \end{array}$$

Thus neglecting numerical integration, we arrive at the following finite dimensional approximate problem (P_h^{II}) : Find $(\underline{\hat{w}}_h, \hat{\varphi}_h) \in W_h \times M_h^{II}$ such that

$$\left\{ \begin{array}{lll} F'(\underline{\hat{w}}_h, \underline{w}_h - \underline{\hat{w}}_h) + \langle \underline{w}_h - \underline{\hat{w}}_h, \nabla\hat{\varphi}_h\rangle & \geq 0 & \forall\, \underline{w}_h \in W_h\,; \\ \langle \underline{\hat{w}}_h, \nabla\varphi_h\rangle & = 0 & \forall\, \varphi_h \in M_h^{II}\,. \end{array} \right.$$

Interpretation of the Finite Element Discretization

Let by finite element discretization approximate flows $(v_h, w_h) := \underline{u}_h(\underline{x})$ at the vertices $\underline{x}$ of the triangulation $\mathcal{T}_h$ of Ω be given. In this section we show how these data can be interpreted as edge flows in a special graph $\mathcal{G} = (N, E)$, where the set N of vertices and the set E of edges are appropriately defined.

The triangles T of $\mathcal{T}_h$ can be distinguished as follows:

- interior triangles, i.e. triangles lying in (the interrior of) Ω;
- boundary triangles, i.e. triangles with one side lying at $\partial\Omega$;
- corner triangles, i.e. triangles with only one side Γ_T in Ω.

Because of the boundary condition and

$$0 \approx \int\limits_T \text{div } \underline{u}_h\, d\underline{x} = \int\limits_{\partial T} \underline{u}_h \cdot \underline{\nu}\, ds = \int\limits_{\Gamma_T} \underline{u}_h \cdot \underline{\nu}\, ds$$

we can drop all corner triangles. With the remaining boundary and corner triangles T of $\mathcal{T}_h$, we perform the following subdivision procedure (as for the construction of the Powell–Sabin splines in [4] p.34):

1. Choose an interior point $\underline{z}_j$ in each triangle T_j (e.g. the centre of the inscribed circle of T_j), so that if two triangles T_j and T_l have a common edge, then the line joining these interior points $\underline{z}_j$ and $\underline{z}_l$ intersects the common edge at a point $\underline{q}_{j,l}$ between its vertices.
2. Join each point $\underline{z}_j$ to the vertices of T_j.
3. For each edge of the triangle T_j

- which belongs to $\partial\Omega$, join $\underline{z}_j$ to an arbitrary point of the edges between the vertices (e.g. the midpoint)
- which is common to a triangle T_l, join $\underline{z}_j$ to $\underline{q}_{j,l}$.

Thus we obtain a refinement $\mathcal{T}_h^*$ of $\mathcal{T}_h$ where each triangle of $\mathcal{T}_h$ is subdivided into six smaller subtriangles. By a regular transformation the six computed values v_h^l, w_h^l $(l = 1, 2, 3)$ at the three vertices of any triangle $T \in \mathcal{T}_h$ correspond to the six flows f_ν along the edges Γ^ν $(\nu = 1, \dots, 6)$ of the subtriangles of T that coincide with ∂T via

$$\int_T \operatorname{div} \underline{u}_h \, d\underline{x} = \sum_{\nu=1}^{6} \int_{\Gamma^\nu} \underline{u}_h \cdot \underline{\nu} \, ds =: \sum_{\nu=1}^{6} f_\nu .$$

Similar as in the subdivision procedure above, we find interior points z_j^* in each triangle T_j^* of $\mathcal{T}_h^*$ and connect these points z_j^* across common edges. Thus we obtain a connected graph $\mathcal{G} = (N, E)$, where N consists of these points z_j^*. Obviously,

$$\text{degree } (z_j^*) = \begin{cases} 2 & \text{if } z_j^* \in T_j^* \subset T_l,\ T_l \text{ boundary triangle} \\ 3 & \text{if } z_j^* \in T_j^* \subset T_l,\ T_l \text{ interior triangle.} \end{cases}$$

and the flow on the edge $e_\nu \in E$ connecting z_j^* and z_l^* is given by the flow f_ν along the joint boundary Γ^ν of T_j^* and T_l^*. Clearly by the integration along the edges Γ^ν, the edge flows f_ν lie in some intervals. However due to the nonconform approximation, the constraints

$$\int_{\Gamma^\nu} \underline{s} \cdot \underline{\nu} \, ds \le f_\nu \le \int_{\Gamma^\nu} \underline{t} \cdot \underline{\nu} \, ds$$

and the Kirchhoff law are only approximately satisfied.

A Concluding Remark

We emphasize that our continuum model merely intends to cover the static network problem under constraints, e.g. urban traffic flow at some fixed time instant or at the stationary state, or the transport problem of supply and demand. For more realistic modelling of time dependent behaviour – however in the setting of finite networks on contrast – we refer e.g. to [13], where congested urban traffic flow is modelled by difference equations with delays.

References

1. Beckmann, M. J. and Puu, T., Spatial Economics: Density, Potential, and Flow. North-Holland, Amsterdam, 1985.

2. Dafermos, S., Continuum modelling of transportation networks. Transportation Research, 14B (1980), 295-301.

3. Dautray, R. and Lions, J.-L., Mathematical Analysis and Numerical Methods for Science and Technology, Volume 3. Springer, Berlin–New York, 1993.

4. Dierckx, P., Curve and Surface Fitting with Splines. Clarendon Press, Oxford, 1995.

5. Ford, L. R. Jr. and Fulkerson, D. R., Flows in Networks, Princeton University Press, Princeton, New Yersey, 1962.

6. Girault, V. and Raviart, P.-A., Finite Element Approximation of the Navier–Stokes Equations. Lecture Notes in Mathematics, 749. Springer, Berlin–New York, 1981.

7. Gwinner, J., A Hilbert space approach to some flow problems. In: Recent Developments in Optimization, Seventh French German Conference on Optimization. Lect. Notes Econ. Math. Syst., 429 (Eds.: Durier, R. and Michelot, C.) Springer, 1995, 170 - 182.

8. Gwinner, J., Stability of monotone variational inequalities with various applications. In: Variational Inequalities And Network Equlibrium Problems, International School of Mathematics "G. Stampacchia", (Eds.: Giannessi, F. and Maugeri, A.) Plenum Press, New York, 1995, 123 – 142.

9. Iri, M., Network Flow, Transportation and Scheduling. Academic Press, New York, 1969.

10. Iri, M., Theory of flows in continua as approximation to flows in networks. In: Survey of Mathematical Programming, Vol. 2, (Ed.: A. Prekopa) North-Holland, Amsterdam, 1980, 263 - 278.

11. Klötzler, R., Optimal transportation flows. Journal for Analysis and its Applications, 14 (1995), 391–401.

12. Maugeri, A., New classes of variational inequalities and applications to equilibrium problems. Rendiconti Accademia Nazionale delle Scienze detta dei XL, Memorie di Matematica, 11 (1987), 277–284.

13. Mikhailov, L. and Hanus,R., Hierarchical control of congested urban traffic – mathematical modelling and simulation. Mathematics and Computers in Simulation, 37 (1994), 183 – 188.

14. Nozawa, R., Max–flow min–cut theorem in an anisotropic network. Osaka Journal of Mathematics, 27 (1990), 805–842.

15. Roberts, J. E. and Thomas, J.-M., Mixed and Hybrid Methods. In: Handbook of Numerical Analysis: II 1, (Eds.: Ciarlet, P. G. and Lions, J.-L.), North–Holland, Amsterdam, 1991, 523 – 639.

16. Strang, G., Maximal flow through a domain. Mathematical Programming, 26 (1983), 123 - 143.

17. Taguchi, A. and Iri, M., Continuum approximation to dense networks and its application to the analysis of urban road networks. Mathematical Programming Study, 20 (1982), 178 - 217.

SEMI-PHYSICAL MODELLING OF A FDI BENCHMARK PROCESS

Đani Juričić, Gregor Dolanc, Andrej Rakar
Jožef Stefan Institute
Jamova 39, Ljubljana, Slovenia
e-mail: dani.juricic@ijs.si

Abstract. Many of the faults in chemical processes occur on the level of transport of fluids and raw materials. Leaks, clogs, valve blockages and sensor faults are only few of them. To study the corresponding diagnostic problems, a benchmark is built around the laboratory desktop plant composed of tanks interconnected with various hydrodynamic paths. The paper describes an approach to the mathematical modelling of the plant with aim to achieve a model which could simulate the system in fault-free and various faulty states. The approach consists of two main steps. First a nominal model of the plant is derived. Then, in the second step, the model is extended in order to capture the fault entries. In the first step, we start with causal ordering of the process variables with aim to remove algebraic loops from the simulation model. Then, a stage-wise identification approach, supported by additional physical assumptions, is applied in order to establish the model structure and parameters. To extend the model validity also to faulty regimes, additional fault entries are added to the nominal model. The final result is a benchmark which consists of the SIMULINK file with non-linear simulation model of the plant. It covers the whole operating region and allows for simulation of over 20 different faults in sensors, actuators and components.

1. Motivation

In spite of intensive research in the area of model based fault detection and isolation (FDI) the methods developed are still not so widely used in industrial practice [1]. One of the reasons is certainly lack of thorough evaluation of the FDI algorithms on realistic plants. A way to alleviate this problem is to provide the FDI research community with diverse test cases equipped with sound mathematical models of the plant along with data records obtained at different faulty states of the plant (c.f. [2]). However, the development of a comprehensive analytical model for simulating nominal and various faulty regimes of a plant operation is often not a trivial task.

The intention of this paper is to present a stage-wise modelling strategy which facilitates the model building by gradual extension of the model's scope either from particular pieces of a priori knowledge or from dedicated experimental data. The plant considered is described in the second section. Causal ordering - a step towards model structure design - is illustrated in the third section. The stage-wise semi-physical modelling procedure is outlined in the forth section. Finally, an idea concerning extension of the nominal model with faulty modes is presented in the fifth, section.

2. Plant description

This benchmark is built around a desktop test facility available at the Department of Computer Automation and Control at Jožef Stefan Institute. The plant "mimics" some of the processes which are common in the transport of fluids in chemical plants. A look at the plant is given in Figure 1. Its main features are the following:

- the plant is well equipped with sensors, thus allowing for synthesis of detailed mathematical models,
- injection of real faults is relatively easy and
- the input/output signals of the plant are reachable by Simulink running on a PC-486 interfaced with system Burr-Brown PCI 20000.

Process flowsheet is depicted in Figure 2. It consists of three tanks R1, R2 and R3 connected with flow paths which serve to supply water from the main reservoir R0. Two of the paths have built-in pumps, that is pump P1 and P2, driven with DC motors with permanent magnet. The angular speed of the motor is controlled by the analogue controller. The time constants of the angular speed is very short so that (practically) there is no lag between the reference speed and the real speed of the pump.

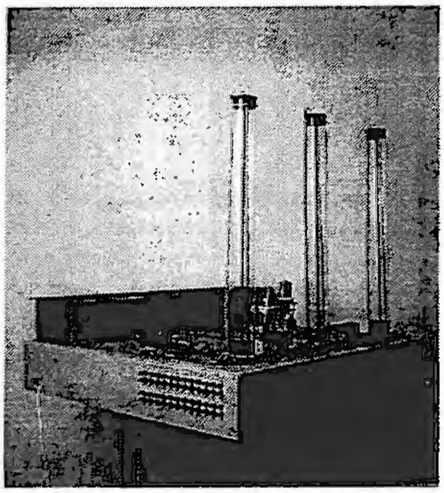


Figure 1: View of the benchmark test plant

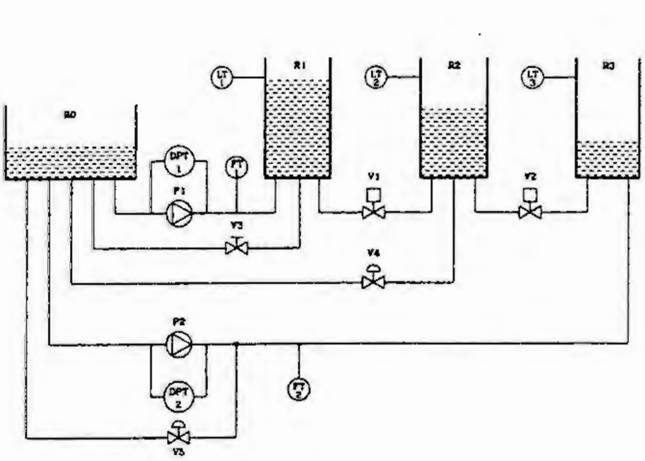


Figure 2: Process flowsheet

There are two configurations of active flow paths available. In the first one, flow is generated by varying the angular speed of the pump P1. In the second case, pump P2 works at constant speed. Flow is then varied by manipulating the valve V5.

There are two servo-valves in the plant, i.e. V4 and V5 driven by DC motors. Valves V1 and V2 are on-off while V3 is manual valve. The purpose of the latter is mainly to emulate "real" faults, i.e. leakage of the tank R1. Capacity of the reservoir R0 is much greater than the capacity of the tanks so that its level is practically constant during the operation.

In our study tanks R1 and R3 take over the role of buffers for supplying R2. Contents from R1 and R3 are "mixed" in R2 and then fed back to the reservoir R0. The level in R2, and hence the flow from R2 to R0, is controlled by the valve V4. This could be viewed as production rate. Level in the tank R1 is controlled by manipulating the reference speed ω_{1r} of the pump P1 while level in R3 is controlled by manipulating the command signal u_5 of the valve V5. The working point is defined by the constant speed ω_{2r}.

3. Towards the model structure: causal ordering

The modelling process starts with decomposing the plant into the independent functional modules (e.g. tanks and flow branches). Then for each module we try do derive a physical model whenever possible. If this is not the case, the model is built by gradually combining different pieces of evidence about the module behaviour. The first step is ordering of causal relationships. The method of Iwasaki and Simon [3] is used. To illustrate it let us take the branch with pump P1 and Venturi pipe VE1 acting as FT1 (c.f. Figures 2 and 3).

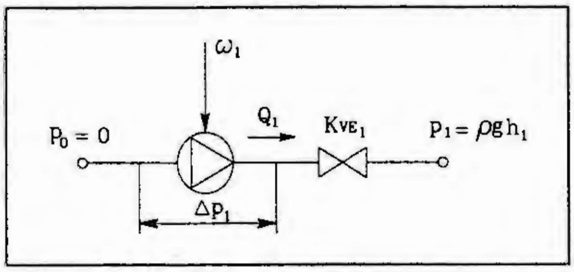


Figure 3.: Branch with pump P1

Let us now stress the equations which describe particular parts of the branch starting with the hydrodynamic part.

Observe first that pressure difference Δp_1 on the pump P1 depends on flow produced by the pump Q_1 and speed of rotation ω_1:

$$\Delta p_1 = f_{\Delta p_1}(Q_1, \omega_1) \qquad (1)$$

Pressure drop on the Venturi pipe is

$$p_{VE1} = f_{pVE1}(Q_1) \tag{2}$$

For both pressures it holds

$$\Delta p_1 + \Delta p_{VE1} = \rho g h_1 \tag{3}$$

where h_1 is level in the tank R1. Putting (1) and (2) into (3) results in

$$f_{\Delta p_1}(Q_1, \omega_1) + f_{\Delta p_{VE1}}(Q_1) = \rho g h_1 \tag{4}$$

Model of the DC motor running P1 consists of two parts: electrical and mechanical. The electrical part can be satisfactorily represented by the following relation connecting voltage U_1, current I_1 and speed ω_1 (c.f. [4]):

$$U_1 = R_1 I_1 + \Psi_1 \omega_1 \tag{5}$$

Mechanical power produced by the DC motor equals $\Psi_1 I_1 \omega_1$. Part of this power is consumed on compensating friction, the other part serves for generating flow. The latter depends on the flow Q_1 and required pressure difference Δp_1. Hence the whole relationship could be written as follows:

$$\Psi_1 I_1 \omega_1 = M_{fr1}(\omega_1) \cdot \omega_1 + \Delta p_1 \cdot Q_1 \cdot (sign(Q_1) + 1) \cdot 0.5 \tag{6}$$

The *sign* part is required to allow for the second term on the right side only when the pump acts as flow generator. By dividing both sides of expression (6) with $\Psi_1 \omega_1$ we get the explicit expression for I_1.

The model (1), (4),(5) and (6) is not a self-contained structure (Iwasaki and Simon, 1986) since it contains 4 equations with 6 variables. Therefore two external variables are defined, in this case pump rotational speed ω_1 and level h_1. The remaining variables form a system represented in Table 1.

Table 1

	Q_1	Δp_1	U_1	I_1
Eq1	1	1		
Eq4	1			
Eq5			1	1
Eq6	1	1		1

Table 2

	Δp_1	U_1	I_1
Eq1	1		
Eq5		1	1
Eq6	1		1

Table 3

	U_1	I_1
Eq5	1	1
Eq6		1

Eq4 is minimal complete subset of first order with $\{Q_1\}$ belonging to it. By substituting Q_1 into Eq1, Eq.5 and Eq6 the system shown in Table 2 follows.

The minimal complete subset of second order is now Eq1 with $\{\Delta p_1\}$. By inserting Δp_1 to Eq5 and Eq6 the system in Table 3 follows resulting in Eq5 as minimal complete subset of third order with $\{I_1\}$ The final minimal complete subset of fourth order is then Eq4 with $\{U_1\}$.

The causal structure is depicted in Figure 4.

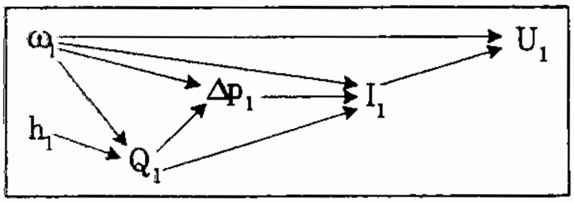


Figure 4: Causal structure of the branch with pump P1

Using similar arguments, the causal structure for the branch with pump P2 (Figure 5) is obtained (Figure 6).

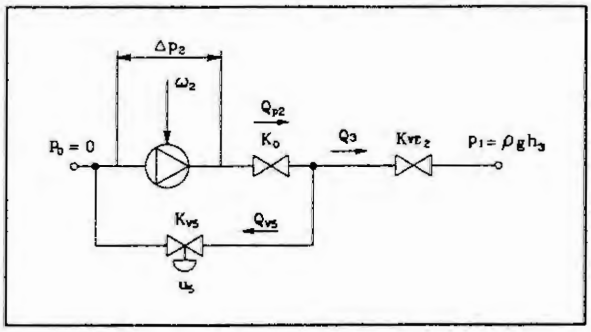


Figure 5: Branch with pump P2

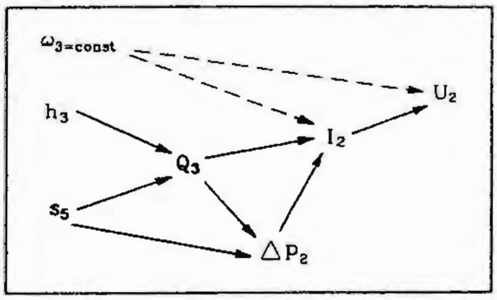


Figure 6: Causal structure of the branch with P2

It is important to note that some of the causal relationships indicated in Figure 4 can be obtained by physical modelling (c.f. Eqs.5 and 6). Hydrodynamic behaviour of the pump P1 can not be properly derived by physical modelling. Instead, a stage-wise identification is applied to the measured static characteristic. The same holds also for the branch with P2.

4. Semi-physical modelling approach

4.1 Stage-wise identification procedure

Pure physical modelling of the active flow paths turned to be impossible due to complex characteristics of the corresponding active elements. Instead, the semi-physical modelling is applied so that heuristic arguments and insight about the system behaviour is taken into account.

To illustrate the idea, let us take the pump P1 and the relationship $Q_1=f(\omega_1,h_1)$. A straightforward way to get the model structure would be Taylor expansion of the expression $f(\omega_1,h_1)$ and then to pick up several most significant terms. However, this approach might be computationally demanding and also the number of unknown parameters to be estimated could be unreasonably high. Instead of polynomial dependencies we try with other types of dependencies based on experiments tailored to reveal particular aspects of the plant. In our case three steps have to be done.

Step 1: Let us start with the observation that condition for zero flow through the branch is fulfilled at given speed ω_1 if the level h_1 in the tank R1 takes the value $h_1=h_{1stat}(\omega_1)$. Then

$$Q_1(\omega_1, h_{1stat}(\omega_1))=0. \qquad (7)$$

It is easy to find (from the measured data) the regression for $h_{1stat}(\omega_1)$ which is a second order polynomial of ω_1:

$$h_{1stat} = h_{11}\omega_1 + h_{12}\omega_1^2$$

Step 2: Try to describe the relation between the flow through the pump and the level h_1 at constant speed ω_1. The data obtained suggest the following type of the regression consistent with (7):

$$Q(h_1|\omega_1 = const) =$$

$$= k_1 \cdot \sqrt{h_{1stat}(\omega_1)-h_1} + k_2 \cdot (h_{1stat}(\omega_1)-h_1) + k_3(h_{1stat}(\omega_1)-h_1)^{\frac{3}{2}} + \ldots.\textit{higher terms} \qquad (8)$$

By means of the stage-wise regression analysis it is found that the first two terms suffice to describe the curves $Q(h_1|\omega_1)$ good enough at any ω_1.

Step 3: From the set of estimated k_1 and k_2 at various ω_1 the regressions for $k_1(\omega_1)$ and $k_2(\omega_1)$ are constructed relatively easily in terms of the rotational speed ω_1. Regression over the values estimated in the second step results in a second order polynomial for k_1 and first order one for k_2, i.e.

$$k_1(\omega_1) = k_{1F0} + k_{1F1}\omega_1 + k_{1F2}\omega_1^2$$
$$k_2(\omega_1) = k_{2F0} + k_{2F1}\omega_1 \tag{9}$$

The final flow model is thus

$$Q(h_1,\omega_1) = k_1(\omega_1)\cdot\sqrt{h_{1stat}(\omega_1) - h_1} + k_2(\omega_1)\cdot(h_{1stat}(\omega_1) - h_1) \tag{10}$$

Comparison between modelled and measured flow Q_1 is shown in Figure 7.

4.2. Summary of the nominal non-linear model

Similar stage-wise procedure is applied to the branch with pump P2 resulting in:

$$Q_3(s_5,h_3) = f_{1F}(s_5)(h_{3stat}(s_5) - h_3) + f_{2F}(s_5)\sqrt{h_{3stat}(s_5) - h_3} \tag{11}$$

and

$$\Delta p_2(s_5,Q_3) = \beta_{00} + \beta_{01}s_5 + (\beta_{10} + \beta_{11}s_5)Q_3 + (\beta_{20} + \beta_{21}s_5)Q_3^2 \tag{12}$$

where

$$f_{1F}(s_5) = f_{1F0} + f_{1F1}s_5$$
$$f_{2F}(s_5) = f_{2F0} + f_{2F1}s_5$$
$$h_{3stat}(s_5) = h_{30} + h_{31}s_5$$

and s_5 is measured position of the valve V5.

Modelling and identification of other modules of the plant is relatively straightforward. Basic assumption used is that quadratic relationship between pressure and flow holds. To sum up, the nominal model of the plant, is defined by the following variables:

state variables	h_1, h_2, h_3
command inputs	ω_1, s_4, s_5
measured outputs	h_1, h_2, h_3, Q_1, Δp_1, I_1, U_1, Q_3, Δp_2, I_2, U_2

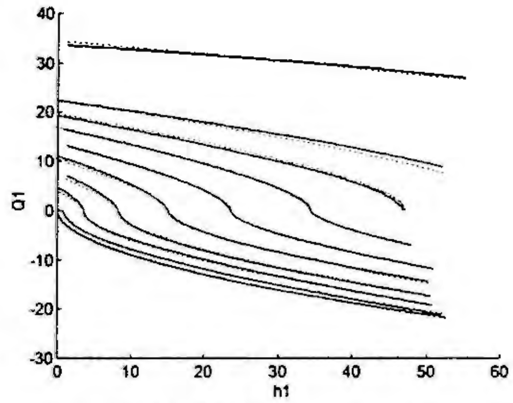


Figure 7: Modelled and measured static characteristic of the branch with pump P1

where h_1, h_2, h_3 are levels in R1, R2 and R3, s_4 and s_5 are positions of valves V4 and V5, Q_3 is flow produced by the branch, Δp_2 is pressure on the pump P2, I_2 and U_2 are current and voltage in the DC motor of the second pump. For more details concerning modelling and model validation c.f. [5].

5. Extending the nominal model with fault entries

The three-tank system allows realisation of various faults which can be either *realistic* or *virtual*. Some of the real faults are:

a) leak from the tank R1 (by opening the manual valve V3)
b) clogs in branch with P1 and P2 and branch with V4
c) increased friction in the pumps
d) offsets in sensors.

Faults a), b) and c) can be realised on the equipment. Faults c) might occur during long term runs. Virtual faults include:

a) sensor faults, i.e. biases, change in gain and
b) actuator faults, e.g. hysteresis and blockages in valves.

They can be easily realised by properly "contaminating" the realistic measured values.

With slight modifications of the nominal model of the plant it is possible to include sensor, actuators and some component faults in the simulation model. For example, clog in the branch with V4 can be simulated by defining a fault entry $f_{V4} \in [0,1]$ which multiplies the valve flow coefficient k_{V4}, i.e. $k_{V4} \rightarrow (1-f_{V4}) \cdot k_{V4}$. In case of no fault $f_{V4}=0$. In case of total clog $f_{V4}=1$resulting in zero flow coefficient of the branch with V4.

However, simulation of certain faults is not that obvious. As example, let us take clog in branch with pump P1. To solve the problem let us represent clog as additional resistance connected in series with VE1 having the flow coefficient

$$k_{c\log} = k_{VE1} \frac{1-\xi}{\xi} \quad , \xi \in [0,1] \tag{13}$$

The problem is that flow coefficient k_{VE1} does not explicitly occur in expression for Q_1 (10). For $\xi=0$, Q_1 obeys (10), otherwise Q_1 is solution of the algebraic equation

$$Q_1 = f\left(\omega_1, h_1 + \frac{Q_1^2}{k^2_{c\log}(\xi)}\right) \quad , \text{ f}(\cdot) = \text{expresssion (10)} \tag{14}$$

These solutions should be found for all combinations of ω_1, h_1 and ξ, so that the extended model $Q_1(\omega_1,h_1,\xi)$ can be identified. This is certainly quite demanding task. Instead, we make use of the following approximation:

$$Q_1(\omega_1,h_1,\xi) = f(\omega_1,h_1) + (\frac{\partial Q_1}{\partial \xi}|_{\xi=0})\cdot\xi \;\; + (\frac{\partial^2 Q_1}{\partial \xi^2}|_{\xi=0})\cdot\xi^2 + ... \tag{15}$$

where the partial derivative $\partial^i Q_1/\partial \xi^i$ is obtained from (14). The resulting expressions are somewhat lengthy, but, what is most important, they can be calculated from the nominal model of the plant .

6. Conclusions

Main ideas used in the realisation of a FDI benchmark case are presented. The procedure consists of two steps: first, derivation of the nominal model of the plant and then, extension of the nominal model with faulty entries. Here we address the problem of derivation of the FDI simulation model of the plant in case that parts of the model are obtained by identification ("grey" models). A stage-wise identification strategy is introduced which relies on gradual extension of the "space" of model validity by adding new pieces of model structure consistent with partial experimental observations. This offers clear advantage compared to the case when the entire model structure has to be guessed from scratch. The procedure might demand substantial experimental data as well as designer's skill. If measurements are not feasible, a priori knowledge from other sources has to be used (e.g. data provided by producers of the equipment).

The final result is benchmark which consists of the non-linear model in Simulink format and a set of data files containing measurements on the real plant at different working conditions and in the presence of real faults.

Acknowledgement

This research has been supported by the EC through COPERNICUS grant CT94-02337 and by the Slovene Ministry of Science and Technology.

7. References

1. Frank,P.M., Analytical and qualitative model-based fault diagnosis - a survey and some new results. European Journal of Control, Vol. 2, No. 1, 1996, 6-28.
2. Blanke,M., S. Bogh, R.B. Jorgensen, R.J. Patton, Fault detection for diesel engine actuator - a benchmark for FDI, Prep. SAFEPROCESS'94, Espoo, Vol.2, 1994, 498-506.
3. Iwasaki, Y., H.A. Simon, Causality in device behavior. *Artificial Intelligence*, Vol. 29, 1986, pp.3-32.
4. Vrančić, Đ. Juričić and T. Höfling (1994). Measurements and mathematical modelling of a DC motor for the purpose of fault diagnosis, IJS Report 7091.
5. Dolanc,G., Đ. Juričić, A. Rakar, J. Petrovčič, D. Vrančič, Three-tank benchmark test, Report, Copernicus Project CT94-02337.

Sequential and Parallel Performance of EMEP's Transboundary Air Pollution Model

S. Unger
GMD FIRST
Rudower Chaussee 5, D-12489 Berlin, Germany

Abstract. The parallelisation procedure is described for the implementation of EMEP's Transboundary Air Pollution Model on the MANNA - a distributed memory platform, developed at GMD FIRST. In addition to the standard PVM implementation, which can run on almost all parallel platforms, the special feature of the MANNA is used (two processors on each board, which share a common memory), and it is shown, that this gives a further speed-up. The resulting super-linear speed up is discussed. The performance of the serial program or of the parallel program, running on few processors, can be increased using the same partitioning techniques. The corrected speed-up characteristics are given.

Keywords: transport simulation; parallelisation; air pollution modeling; Lagrangian modeling; parallel processing.

Introduction

This paper is a continuation of [7], where the parallelisation procedure and some speed-up characteristics were given. It was mentioned there, that the resulting super-linear speed-up is a result of better cache use in case of arrays with lower dimensions. Consequently, applying the same techniques to cases, where the array dimensions are greater than the optimal, an increase in perfomance should follow. We will give here a summary of the contents of [7], and than present the 'corrected' speed-up values, defined in comparison with some 'optimal' serial performance.

The work was carried out in collaboration with the International Institute of Applied Systems Analysis (IIASA), where an Integrated Assessment Model for ozone is now under development [4]. In particular, this model requires a description of the source-receptor relationships between precursor emissions and the concentrations of ozone in the atmospheric boundary layer. These relationships need to be valid over a range of different spatial patterns of emission sources, and should not be restricted to the present-day situation only. For this reason, attempts to define these relationships solely on the basis of recent ozone measurement data are likely to prove inadequate.

Instead, the simplified ozone formation description was built on the basis of EMEP's ozone model [5,6], which has gained widespread international acceptance. Unfortunately, the operation of the EMEP model on the mainframe computer of the Norwegian Meteorological Institute is rather time- and resource-intensive. Therefore, carrying out the large number of scenario runs (some hundreds, each requiring about 6 hours of CPU-time) necessary for constructing the regression model is an expensive undertaking. In order to accomplish and simplify this task, the EMEP model has been parallelised, which results in a significant decrease of computing time.

In Section 1, some characteristics of the EMEP model are given and the used parallel computer and its run time system is shortly described. Section 2 deals with the methods of parallelisation. Section 3 contains the results, e.g. CPU-time and speed-up's, and some discussion of these facts. The speed-up is defined as ratio of the CPU-time for a single node run and the CPU-time of a run on n nodes for the same problem. The parallel efficiency is the ratio of speed-up and the number of nodes. At the end of the paper, conclusions and some short remarks are given.

1. The EMEP Model and the used Hardware

The EMEP ozone model is a single-layer Lagrangian trajectory model which calculates concentrations of photochemical oxidants every six hours for a set of up to 740 arrival points (on a 150 km x 150 km grid) covering the whole of Europe. Columns of air in the atmospheric boundary layer are followed along specified 96-hour trajectories, picking up emissions of NOx, VOC, CO and SO2 from the underlying grid. The height of the air column, the mixing height, containing the bulk of the polluted air is updated from radiosonde data at 1200 GMT each day.

Along each trajectory, the mass conservation equations are integrated, taking into account emission inputs, photolysis and chemical reactions, dry and wet removal, and the influence of meteorological parameters. These equations are solved numerically, using the quasi-steady state approximation method with a fixed timestep of 15 minutes.

As in every Lagrangian model, one has to deal with huge sets of data, e.g. the meteorological arrays, the concentration array for assimilation (initialisation of concentrations), emission data, which are given in a finer 50 km x 50 km grid. In addition, there is a difference compared to common Lagrangian models:

- There is only a relatively small (about 740) set of relatively short (maximum 49 points) trajectories, which are independent of each other considering one timestep. There is a coupling between trajectories of different timesteps due to the initialisation of the concentrations.

The program was designed to run on the MANNA (*M*assively parallel *A*rchitecture for *N*umerical and *N*onnumerical *A*pplications) computer, developed and built at GMD FIRST [3]. MANNA is a dual-processor-node distributed-memory machine using crossbar technology for node interconnection. A dual-processor-node consists of two software-programmable i860XP-processors. The basic idea behind this architecture is to have one processor in charge of application program processing and utilize the second processor for global communication (i.e., inter-node message passing). PEACE - a special object-oriented parallel operating system for MANNA was also developed at GMD FIRST [2]. It allows to operate the machine not only in this AP/CP (Arithmetic Processor/ Communication Processor) mode, but also in single processor mode and AP/AP^{CP} (Arithmetic Processor/Arithmetic Co-Processor) mode, where one processor is used as an arithmetic co-processors to the other one, which is responsible for communication and part of calculations.

Our application is distinguished by very coarse grain communication with few, but long messages. Consequently, the MANNA was used in single processor mode, which is easier to program, and in the AP/AP^{CP} mode of operation. Latency hiding was reached by an overlay of communication with calculation (send as early as possible, receive if necessary). The AP/AP^{CP} mode may lead to some performance degradation due to insufficient memory access rate, if both processors run memory intensive applications. The corresponding speed-up factor turned out to be 1.8 instead of the theoretically possible 2.0. The calculations were done on a base block of the MANNA with 20 nodes (40 processors).

2. Parallelisation and Optimisation

The parallelisation was done by means of the PVM (Parallel Virtual Machine) message passing software [1]. Essentially, it consists of programs allowing to start, synchronise and stop processes on (possibly) remote processors and to exchange data between them.

The extensive input and output operations require something like a host node which has to face this task. On the other side, only the calculations of a particular timestep can be parallelised, because of the inter-timestep-dependency. So, the calculation of the trajectories will be done on this host node, besides the input/output work. After defining the trajectories, they have to be distributed among the other processors in such a way, that good load balance is achieved. The simplest way is, to order at first the trajectories according to their length, beginning with the longest. After that, one can control the amount of work given to each processor. The distribution is then done in the following way:

Assume, there are *nproz* processors, the *n*-th processor (*n=0* is the host, so *n=1,nproz-1*) will get trajectories with numbers *n+i*(nproz-1),i=0,ntraj/(nproz-1)-1*, where *ntraj* is the number of trajectories.

This procedure has proved to be very efficient. First, it is very fast, because it requires only few calculations. Second, it gives a very good load balancing and third, it makes the program much simpler itself. In the following, it is explained shortly, why.

In the program, an index array is defined which shows, if a given trajectory is 'active' at a given time. This means, some trajectories will start later with respect to a loop over all possible points of a trajectory, because the trajectory has left the computational domain, before the maximal length of a trajectory of four days has been reached. Conseqently, the calculations for these trajectories begin later. So, nearly all do-loops over the trajectories contain a statement '*if the trajectory is active at given time, then do the following operations*'. Now, all these do-loops can be transformed to do-loops with variable upper bounds, not containing such an if-construct.

At each timestep, the new meteorological data have to be sent to all processors. But, this is not necessary for the assimilated values. Only a little amount of trajectories are so very short, that they need assimilated values of the preceeding timestep. This little amount will be computed on the host node itself. The other trajectories can be initialised before the last timestep is finished and then these initial concentrations can be sent to the nodes. This allows an overlay of communication and calculation. The time for communication will be hidden behind the calculation time, because data are sent much more earlier when they are necessary for the receiver. Consequently, the receiver has not to wait for new data, it will find the data in its mail box.

Besides the elimination of the if-constructs in the do-loops, which is possible having an ordered array of trajectories, the FORTRAN-code was widely restructured, to get performance gain by compiler optimisations. The used PORTLAND-compiler allows a software pipelining, increasing the performance of the processors at least by a factor of two. This pipelining is only possible for so called '*simple do-loops*', which do not contain any function calls or if-constructs. Besides this, the corresponding loops must not contain to many statements to allow the compiler to do its work. Fortunately, a great part of the code may be restructured in such a way and the resulting good perfomance is mostly due to this fact.

Since [7] was published, some further changes have been made, mostly concerning the organisation of the calculations in the chemistry solver. In particular, in the case, when all photolytic rates disappear due to night conditions for all trajectories, a 'night-time' solver is introduced, which is about 10-15 % faster than the 'day-time' solver. This gave a further improvement of the performance (cf. the values in Table 1 with the values in Table 1 of [7]).

3. Results and Discussion

Program's CPU-time, in particular for low numbers of processors, is quite long. Consequently, all time measurements have been done for 20 timesteps of the program. This covers a real time of 5 days.

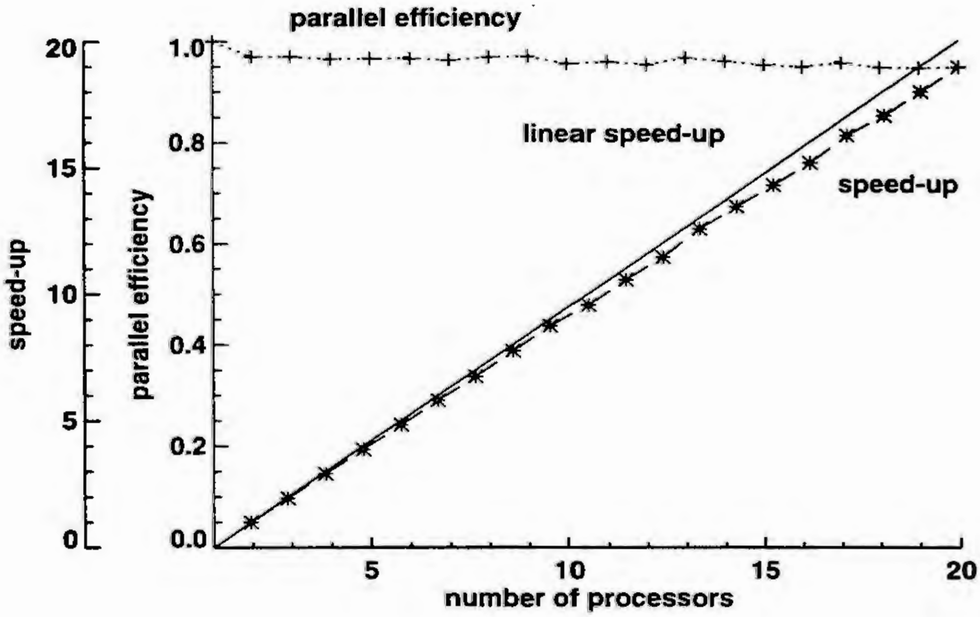


Figure 1. Speed-up and parallel efficiency in single processor mode

The original program requires a CPU-time of 2 hours 10 minutes on one processor of the MANNA for these 20 steps. Besides the fact, that in the original program some more output is done than in the parallel version, this is a result of the above mentioned structure of the program, which does not allow to use the pipelining capabilities of the i860XP-processor. Application of the optimisation procedure, in particular ordering of trajectories and rearrangement of the do-loops, to get simple loops, gives a CPU-time of somewhat more than 36 minutes for the optimised version. This time can be further reduced by splitting of the arrays into 27 parts to less than 32 minutes.

In [7] different speed-up characteristics were given, which did not take into account, that the splitting of the arrays would give better performance of the serial program. Consequently, a super-linear speed-up of 21.6 for the one processor case and of 22.3 for the two processor case was acchieved. Here we give the same values in comparison to an optimised version of the single node case. Consequently, the speed-up values are no longer super-linear.

Table 1: CPU-time, optimum CPU-time and optimal number of parts for 20 steps of the EMEP model on MANNA, n: number of used nodes, a: single processor mode, b: AP/AP^{CP} mode

n	a: CPU	a: best	a: parts	b: CPU	b: best	b: parts
1	2178.8	1908.2	27	1228.5	1053.0	16
2	1081.3	985.9	13	599.1	540.0	8
3	698.2	655.1	8	383.0	355.8	5
4	525.7	494.0	6	281.7	265.9	4
5	413.5	395.4	5	220.3	210.9	3
6	338.9	329.0	5	182.6	176.0	3
7	291.0	283.1	4	154.2	150.3	2
8	252.6	246.2	3	135.2	132.1	2
9	222.9	218.7	3	119.6	117.4	2
10	203.4	199.9	2	107.9	107.9	1
11	182.8	180.8	2	97.8	97.8	1
12	168.4	166.8	2	90.0	90.0	1
13	154.2	151.8	2	83.6	83.6	1
14	142.5	142.0	2	77.8	77.8	1
15	134.5	133.5	2	72.1	72.1	1
16	126.3	125.8	2	67.4	67.4	1
17	117.4	117.4	1	63.9	63.9	1
18	112.0	112.0	1	60.8	60.8	1
19	106.2	106.2	1	57.5	57.5	1
20	100.7	100.7	1	55.0	55.0	1

In Table 1, some CPU-times (in seconds) and speed-up values are given for up to 20 nodes of the MANNA. The CPU-times again correspond to 20 timesteps of the program. A full run of 6 months requires 734 timesteps. So, the CPU-time on 20 nodes, using both processors, will be about 36 minutes. Values with letter *'a'* correspond to the single processor mode, letter *'b'* to the two-processor version. In each case, the column marked *'CPU'* contains the CPU-times of the program without the initialisation time, which is nearly equal for all runs. The column marked *'best'* contains the CPU-time necessary to compute 20 timesteps for the 'optimally splitted' version. Column marked *'parts'* gives the number of parts in which the arrays are divided for this best version. Note, that the array of trajectories is split here in order of occurance of the trajectories in the ordered array, whereas the subdivision between processors is done according to rule, mentioned in section 2, to ensure load balance.

In Figures 1 and 2, the speed-up and parallel efficiency are plotted for the single- and two-processor versions relative to the 'best' version of the single node program, respectively. There is a good speed-up for both modes, reaching a value of about 19 for 20 nodes.

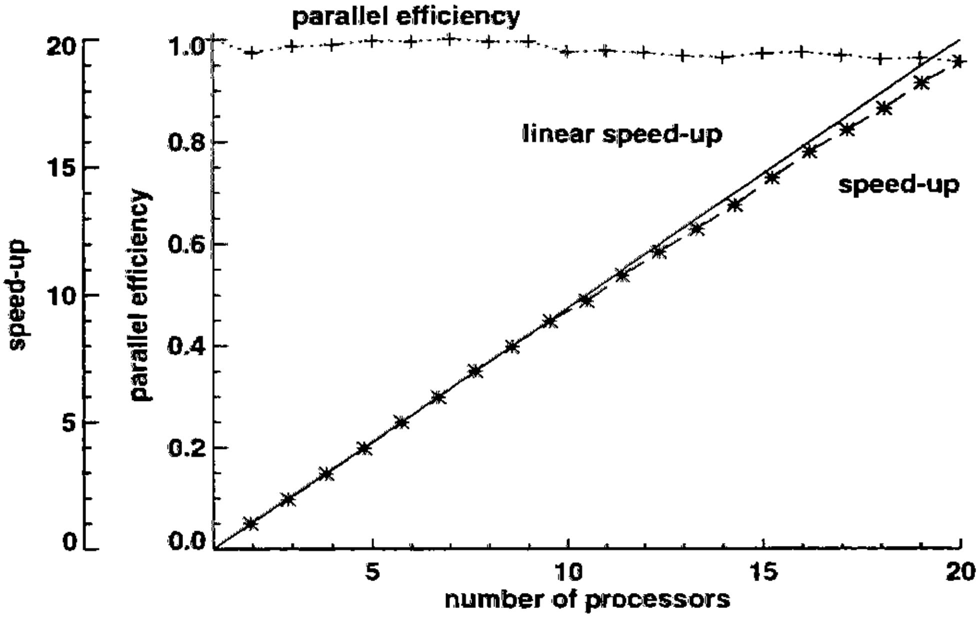


Figure 2. Speed-up and parallel efficiency in AP/AP^{CP}-mode, relative to the one-node/two-processor mode

The CPU-time to compute one trajectory is less in the single processor case. It is nearly proportional to the two processor case with a factor of about 0.9. This is a result of the fact, that both processors use the same data bus, if they exchange data with memory. Of course, this affects the performance.

Conclusions and Remarks

The parallelisation of the EMEP model provides the possibility of running a lot of scenarios in a short time. The time necessary to do a full run is reduced from several hours on a supercomputer to about 36 minutes on a 20 node MANNA. There is good speed-up even in the case of running a problem of fixed size in parallel.

Applying the above mentioned methods to the serial program, in particular, ordering trajectories and splitting calculations similar to the parallel version, improves the performance considerably. The best performance was reached with a split of the 740 arriving sites into 27 parts of length 28 in the single processor case and into 16 parts of length 24 in the two-processor case (note, that in case of two processors each of the processors has to do only half of the work). The corresponding speed-up values are no longer super-linear, but remain very good, reaching a value of about 19 for 20 nodes in both cases even for a problem of fixed size.

Using the 20 node AP/AP^{CP} version of the program, we were able to do 250 scenario runs in about a week

Acknowledgements. The work reported here has been carried out in collaboration with IIASA Laxenburg (Markus Amann, Chris Heyes and Wolfgang Schöpp, who work on the integral assessment model) and I readily acknowledge their helpful cooperation. In particular, I'm grateful to Dave Simpsom from EMEP's Meteorological Synthesizing Centre - West in Oslo, who made the EMEP ozone model available to me. In addition, I want to acknowledge the help of my colleagues at GMD FIRST, who built the MANNA computer and provided the necessary software support for the PVM message passing system and full exploitation of the hardware, in particular

of the second processor.

References

1. Beguelin, A., Dongarra, J., Geist, A., Manchek, R. and Sunderam, V., A User's Guide to PVM - Parallel Virtual Machine. Oak Ridge National Laboratory, Report No. ORNL/TM-11826, Oak Ridge, Tennessee, 1991.
2. Garnatz, Th., Haack, U., Sander, M. and Schröder-Preikschat, W., Experiences made with the Design and Development of a Message-Passing Kernel for a Dual-Processor-Node Parallel Computer. In: Proceedings of the Twenty-Ninth Annual Hawaii International Conference on System Sciences, Maui, Hawaii, January 3-6, 1996. IEEE Computer Society Press.
3. Giloi, W. K. and Brüning, U.. Architectural Trends in Parallel Supercomputers. In: Proceedings of the Second NEC International Symposium on Systems and Computer Architectures, Tokyo, August 1991. Nippon Electric Corp.
4. Heyes, C., Schöpp, W., Amann, M. and Unger, S. , A Simplified Model to Predict Long-Term Ozone Concentrations in Europe. WP-96-12. International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria.
5. Simpson, D., Long period modelling of photochemical oxidants in Europe. Calculations for July 1985. Atmos. Environ., 26A, 1609-1634.
6. Simpson, D., Photochemical model calculations over Europe for two extended summer periods: 1985 and 1989. Model results and comparisons with observations. Atmos. Environ., 27A, 921-943.
7. Unger, S., Parallelisation and Optimisation of EMEP's Transboundary Air Pollution Model. In: Proc. of the 1996 EUROSIM International Conference on HPCN Challenges in Telecomp and Telecom: Parallel Simulation of Complex Systems and Large-Scale Applications, (Eds.: Zuidervaart, J. C., Dekker, L. and Smit, W.) Elsevier, Amsterdam, 1996.

SIMULATION OF SUSPENDED PARTICULATE MATTER TRANSPORT IN A NAVIGATION CANAL

C. Engelhardt, D. Prochnow and H. Bungartz
Institut für Gewässerökologie und Binnenfischerei im Forschungsverbund Berlin e.V.
Rudower Chaussee 6a, D - 12489 Berlin, Germany

Abstract. In this paper, the suspended particulate matter transport along a canal reach of some kilometres upstream from the mouth of the Oder-Spree-Kanal is simulated by a vertically integrated two-dimensional model which devides the suspended solids into different settling velocity classes. In the model, effects of commercial boat traffic in the shipping segment of the canal are considered. Computed spatial distributions of suspended particulate matter (SPM) concentration and of settling velocity are found to be in agreement with experimental data from a field campaign in this area in 1995.

1. Introduction

During dry summer months, water from the Oder River is used to maintain the water level in the upper part of the Oder-Spree-Kanal a measure entailing reversed flow direction in the lower canal reach between the Oder river and the Eisenhüttenstadt pumping station (Fig. 1). This temporary change in flow direction results in a transport of dissolved and particulate matter from the Oder river into the Spree river system and is connected with a number of ecological problems. Particularly because of the particle-bound pollutants the sediment load is a criterion for the assessment of water quality.

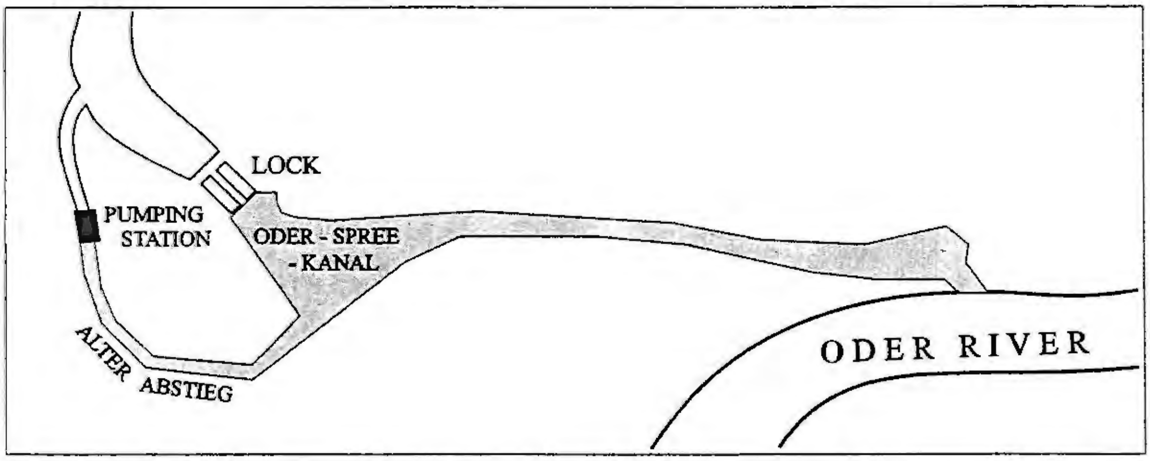


Figure 1. Map of the considered canal reach between the Oder river and the pumping station

Often these pollutants are not homogeneously distributed to the total suspended particulate matter but tend to associate with fractions of it. So it is well known that some heavy metals and some organochlorines are transported predominately by the slow settling velocity fractions of seston [3,5]. For a better assessment of particle-bound pollutant input from the Oder river into the Spree river system it is advantageous to use a transport model which devides the suspended particulate matter (SPM) into settling velocity classes.

2. Transport problem and input data

To predict turbulent flow and suspended particles transport in a navigation canal [1] like the Oder-Spree Kanal (Fig. 1), an initial/boundary value problem have to be solved. Beside a number of parameters the hereby employed 2D, depth-averaged, fractionated model SEDIFLOW [2] requires the knowledge of the initial and boundary values of flow and transport variables. In that case when water was charged from the lower part to the upper part of the canal the outflow rate at the pumping station and the water lost at the lock were known for August 1995 as 5 m³/s and 1,7 m³/s, respectively. Assuming no evaporation, no infiltration at the bed, no withdrawal and no additional inflow, a water input from the Oder river of 3,3 m³/s results.

For the suspended particle transport problem, however, the initial and boundary values have to be predicted by data of field experiments. In addition to the SPM concentration, (determined by filtering, drying, and weighing) the SPM distribution to settling velocity classes as well as values of particle sizes and particle densities are necessary to operate the numerical model . Such a set of experimental data for the water input from the Oder river in August 1995 is given in the following example: SPM concentration 33 mg dry weight/l; SPM dry density 1,9 g/cm³; SPM water content 78,8 %; mean particle size 15,6 µm.

The rapidly decreasing flow velocities from about 1m/s in the Oder river to about 1 cm/s in the Oder-Spree-Kanal lead to a particle settling on their way through the canal. This sedimentation process has to be simulated by the model and, particularly, it must be shown how the measured [4] settling velocity distribution of the particles changes downstream from the Oder river inflow (Fig. 2) to the pumping station.

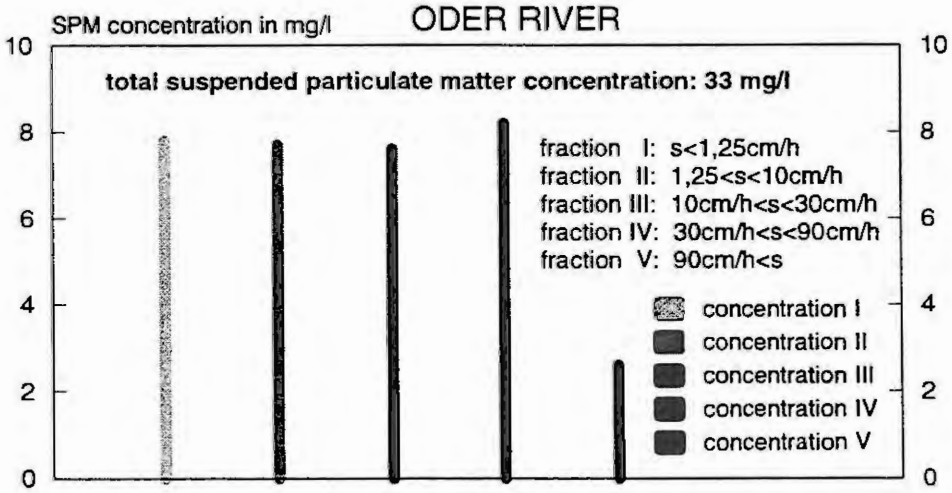


Figure 2. Distribution of SPM concentration of the Oder river water

3. The 2D model simulation

The vertically integrated flow velocities in the discrete flow domain computed by the SEDIFLOW model are shown in figure 3. The highest flow velocities occur in the old part of the canal (called Alter Abstieg, see Fig. 1). Based on this steady flow the transport of five particle fractions, each described by a mean settling velocity was computed. Intervals of the five settling velocity classes are given in figure 2. Figure 4 shows the computed spatial concentration distribution of fraction I, which contains the most slowly sinking particles. One can recognize particles belonging to this class have computed flow velocities in the Alter Abstieg strong enough to erode sediment from the bottom of the canal. That results in an increasing SPM concentration of fraction I downstream the Alter Abstieg. In the remaining stretch, the predicted SPM concentration of fraction I decreases downstream.

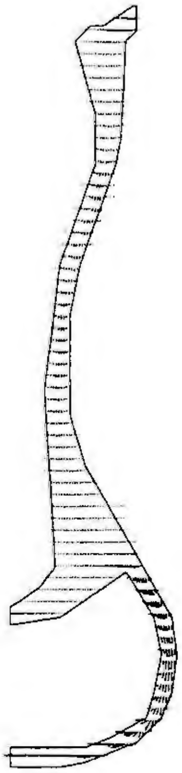


Figure 3. Computed flow velocities

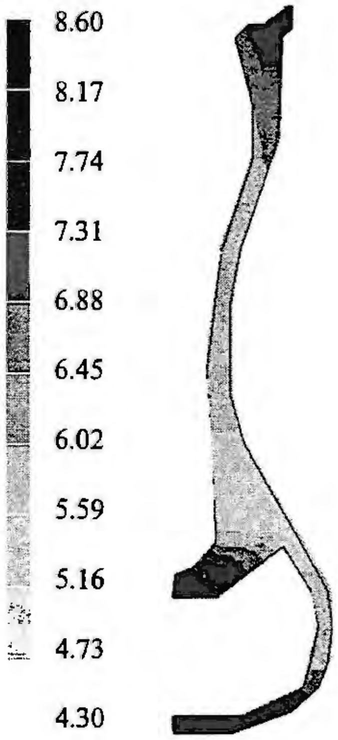


Figure 4. Spatial distribution of the low settling velocity fraction concentration in mg/l

The model simulation for the faster-sinking particles of fractions II-IV presents a sedimentation process, which, after adding the result of fraction I, yields a total SPM concentration between 33 mg/l at the input and 11 mg/l at the pumping station. This model simulation was inconsistent with the measured SPM concentration at the outflow (pumping station) in August 1995, being about twice as big (20 mg/l).

To overcome this contradiction an attempt was made to consider the effects of navigation in the model, as the field data (even in times without navigation) were influenced by shipping too. All the more water samples taken immediately after the passing of ship show an increase of 30 % of SPM concentration and a high part of slowly sinking particles in the total concentration. To describe the navigation effect this measured distribution of the SPM concentration to settling velocity classes was used as a changed initial condition in another model simulation. The non-steady spatial distribution of SPM in the flow domain given by the model after the disturbance of a ship moves with time to the steady-state spatial SPM distribution without any shipping disturbances. If repeated disturbances are assumed (here the assumption is a sixfold, as described above, disturbance over the day), the increased SPM concentration doesn't decompose completely and a new time-independent SPM concentration for every fraction and therefore the total SPM concentration can be calculated. At the pumping station, the agreement of this computed solution (concentration over time diagram for the fractions I-V) with the measured data of SPM concentrations is satisfactory (Fig. 5) for the total seston concentration and fraction I and II. However, figure 5 also shows that according to the model simulation, pumped water contains only slowly-sinking particles of fractions I-III while a small amount of the fast-sinking particles can be found in water samples taken at the pumping station.

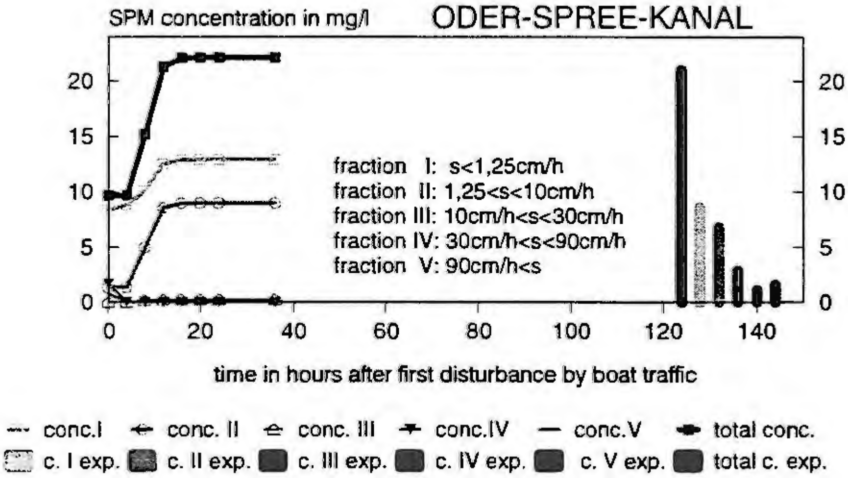


Figure 5. Computed SPM concentrations (simulation with disturbances of six ships per day) in comparision with measured data near the pumping station at the outflow of the flow domain

References

1. Celik, I. and Rodi, W., Mathematical modelling of suspended sediment transport in open channels. In: 21st Congress of the IAHR, Melbourne, Australia, August 19-23, (1985), 533-538
2. Engelhardt, C., Prochnow, D. and Bungartz, H., Modelling and simulation of sedimentation processes in a lowland river. Mathematics and Computers in Simulation, 39 (1995), 627-634.
3. Engelhardt, Ch.; Bungartz, H.; Krüger, A.; Prochnow, D.; Sauer,W.; Schild, R.; Thiele, M.; Krawczyk, H. and Rennert, R., Parameter des Schwebstofftransports in Oder, Neiße und im Oder-Spree-Kanal bei Eisenhüttenstadt. Vom Wasser, 88 (1997), (in press).
4. Prochnow, D., Bungartz, H. and Engelhardt, C.. On the settling velocity distribution of suspended sediments in the Spree River. Arch. Hydrobiol. Spec. Issues Advances in Limnology, 47 (1996), 469-473.
5. Schrap, S. and Opperhuizen, A. Quantifying the sorption of organic chemicals on sediments. Chemosphere 18 (1989), 1883-1893.

DETERMINATION OF SOURCES OF PARTICLES IN AIR BY TRAJECTORY MODELS

I. Gerharz, A. Sydow, S. Unger
GMD FIRST
Rudower Chaussee 5, D-12489 Berlin

Abstract. The importance of the determination of sources of particles or substances in the air has grown in recent years. In particular, interest has been focusing on medical and environmental research. In this paper we will study two applications of trajectory models relevant to these research fields in more detail. These applications are based on the "inversion" of the theory of Lagrangian models. In order to provide basic information for an understanding of the background of the trajectory models, the general idea of this theory is briefly reviewed. The following discussion presents a comparison, taking into account the distinct preconditions of the two models. Respective applicability and restrictions are discussed. Potential consequences will be mentioned. The concluding summary does not intend to present an appraisal.

Introduction

The dispersion of airborne substances is subject to a variety of simulation models. In the group of deterministic dispersion models the trajectory models form a separate class. As the title "determination of sources of particles in air by trajectory models" indicates we are especially interested in this model type. They offer the possibility of being applied in the opposite direction.

Considering the expenditure involved in the inversion of dispersion processes, the computation time and data requirements form the main criteria for the choice of an appropriate model class.

Based on these considerations the Lagrangian models which represent a class of dispersion models have been chosen. Their basic principles will be described in the following section, introducing two applications of the tracing-back trajectory models with different preconditions and objectives. General and specific assumptions and arising problems will be discussed.

Lagrangian dispersion models – a brief review

The basic concept of Lagrangian models is the observation of air volumes or airborne particles. Therefore these models are also referred to as Lagrangian particle simulation models. In this paper the term "particle" denotes any air pollutant or any buoyant substance in the air. However Lagrangian dispersion models are generally also applied to fluid particles.

In contrast to Gaussian models, the Lagrangian trajectory models are appropriate for the description of dispersion in complex meteorological situations or heavily structured topography. The explicit form of a Lagrangian model is mainly determined by the chosen scale influencing, for example, the type of turbulence simulated. The trajectory models use winds and fluctuation caused by turbulence to predict the pathways of particles or air volumes individually, and register modifications in their characteristics in each time step. Particles or air volumes respectively may be released from any number of locations. The simulation of turbulence is based on the statistical theory of Taylor for diffusion effects [7], and an extension made by Obukhov [4] and Smith [6]. The fluctuation is simulated by a Markov process of first order: the velocity fluctuation is composed of a correlated component representing the power of recollection and an independent random component. The recollection is determined by the Lagrangian time-scale, and the random component describes the coincidental effects in diffusion. Since the theory of Taylor expects homogeneous stationary turbulent fields, but most turbulent conditions are inhomogeneous, Legg and Raupach [2] extended the vertical velocity term. Hence the position of a particle or air volume at a certain time will be described by its position at the previous time step and its instantaneous velocity. The latter is composed of the advective wind and the fluctuation component at each position and point of time. For more details see [1] or [3].

One aspect to be mentioned in connection with trajectory models is their need for computer power. It may become tremendous in case of specific requirements, such as a large number of particles or sources, or the request of high accuracy. Although simplifications have been made within the models, HPCs are necessary to achieve a reasonable answering time. Furthermore the implementation on parallel computers is cost-efficient.

Back tracing – two trajectory models

The general idea of back tracing is the determination of the used trajectory of a particle or air box and its initial location at a previous time step. This calculation is done by inverting the previously described theory of Lagrangian trajectory models. The following two examples differ in their preconditions and their conception towards their objective(s). Before entering a deeper discussion we will give some basic information about them.

The trajectory model we regard first is a constituent part of the Transboundary Air pollution model (TAP) developed at EMEP, the European Monitoring and Evaluation Programme at the Norwegian Institute for Meteorology, Oslo (s. also [5]). As the name EMEP implies, it is a Europe-model or more precise, a complex, single-layer macroscale Lagrange-model. Its objective is the determination of the concentrations in each arrival point of a trajectory. The aim of the model component we are interested in is the computation of the origin of each trajectory. At this initial point the concentration of each substance within the traced back air volume is given. With these initial values, the emission, entrainment, deposition, and chemistry are calculated along the determined trajectory of each air box.

The resolution of the model is based on the latitude-longitude-grid with a grid spacing of approximately 150 km respectively. Emission is calculated on a smaller resolution of 50 km×50 km. The vertical coordinate is given by the atmospheric boundary layer. In this model the objects of interest are air boxes or air volumes for which the pathways will be determined. The number of air boxes is limited through the resolution of the modeled terrain. Since the horizontal resolution is quite coarse, a time step less than two hours is not required for the calculation of the trajectories. As the tracing back is carried out for four days, the considered time period is divided into 48 time intervals. Consequently the number of trajectory points in the back tracing is limited to 49, unless the air box is leaving the simulation area early.

The second trajectory model has been realized within the project ANTIGEN at GMD FIRST. It is supposed to locate the original source(s) or at least the region(s) of origin of particles (experimentally) registered. In order to assess the result of computation, a dispersion simulation based on the Lagrange particle theory starting at each of the potential sources is performed to compare the given and calculated values.

The model is intended for the use in urban scale. The horizontal resolution is controlled by the resolution of the given meteorological data and the required accuracy for the dispersion simulation. Therefore the (horizontal) grid spacing is placed in the range of 200 m to 1 km. The vertical component is variable. But since the region of most interest is the ground or surface level with a fixed height of 50 m, a vertical resolution of 50 m has been chosen. The objects considered in this model are particles according to the definition given above combined with the assumption that they are constant in mass and inert. The arbitrary but fixed time step is set to ten seconds. The number of trajectory points is given by a certain tracing-back time period or distance.

Discussion

For simplification the trajectory model used in TAP will be referred to as "model A" and the second one as "model B". The following discussion may not be complete.

The realization of the idea to locate the source of particles or air boxes rests on the reversal of the basic thoughts of the Lagrange model theory. A critical point involved is the fact that effects of diffusion are not reversible. This is especially a crucial aspect in the microscale model B. Within the considered resolution, the fluctuation has a certain influence on the trajectory simulation in the direction of the dispersion. Taking this into consideration, the determination of the trajectories is only based to the velocity of the main advective wind. Furthermore the terrain is assumed to be weakly structured. In order to minimize the errors due to these two restrictions, a dispersion simulation with the full Lagrange model starting at the determined potential sources has been added. The correspondence between the calculated and given data will be evaluated. Whenever it is necessary, variations at the origin points will be performed. In addition, also the time step and computation limit may be varied.

The aspect of the reversed diffusion is of minor relevance in the macroscale model A. Due to the larger scale in time and space the coincidental fluctuation can be disregarded. Currently its eventual effects are included in the provided meteorological data. Topographical effects are only considered through the flow.

In both models the vertical transport is ignored. As the vertical coordinate of model A extends over the whole region of interest, that is the complete atmospheric boundary layer, vertical effects are assumed to be of no influence in a well mixed layer.

In model B the vertical resolution is an arbitrary but fixed parameter like mentioned before. The disregarding of the vertical transport is founded on the assumption that the transport in vertical direction is dominated by turbulent effects which are not considered during the tracing-back in model B. In this case strong vertical wind components are excluded as we demand a weakly structured terrain.

Further aspects are concerning the time step and setting of limits.

In the considered version the model A is oriented by the update of the meteorological data which is supplied every six hours (s. above). This is influencing the size of the time step. Compared with that the time increment in model B is only limited by the requirements of the Lagrange model theory (Lagrangian time-scale). The meteorological information is updated whenever new data is available for the simulation.

In case of model A the concentrations at the arrival point is of interest. It is calculated using the computed trajectory and the initial concentrations four days ago. In contrast to this, the crucial point in model B is a quite good estimation of potential emission sources with the number of steps as independent as possible on time and space. "Natural" limits are given by the range of available meteorological data and the expansion of the simulation terrain.

One basic difference between the two models is the type of object for which the trajectory is determined. Individual particles are traced in model B. They may be equipped with special characteristics and are limited in number of sorts. In model A, i. e. in TAP, seventy different substances are generally of interest. Since the substances are not necessary inert, chemical reactions would have to be considered during the stepping back on each of the particle trajectories. In addition their number would become very large, which could leads to a problem concerning the computer power. However, creating a chemical solver describing the whole reversed chemical processes is not practicable. Therefore air boxes defined by the given resolution are chosen for individual tracing. Within each of them, chemistry, emission and deposition effects are considered in respect to changes in the concentrations.

Summary

The discussion above points out some basic topics in modeling and simulation. A model is required to meet the reality as close as possible. In general simplifications have to be imposed on the model in order to get results in a reasonable time or even to make the model realizable. In this case it is essential to check whether the simplified version still fulfills the requirements of the original model. Concerning the resolution, we have to study for example what kind of physical effects, e. g. turbulence, molecular effects, etc., have to be considered.

In the discussed trajectory models these aspects are taken into account to a high degree. They present a proper application corresponding to the respective objective. Model B represents a specialized microscale trajectory model showing a high flexibility in its choice of the space and time dimensions. On the other hand it has to solve the problem of diffusion effects (s. above). Being a component of a global program the main dimensions of model A like the maximal length of a trajectory, their number, the time step, etc. are fixed. Compared to model B it is coarser concerning the time and spatial resolution. Consequently it is easier to handle and fits to its common use within statistical analysis.

The discussed model versions have not reached their final state yet. In model B the quality of the registered particle data influences the accuracy and quality of the result. Therefore specific demands have to be imposed on the measuring method. In order to promote the computation of the potential sources and the assessment of the correspondence, it is worth including in the algorithm a cadastral map of emission sources.

Acknowledgments. We would like to thank W. Rufeger for helpful comments on drafts of this paper, and P. Mieth for fruitful discussions and helpful suggestions.

References

1. Gerharz, I., Mieth, P. and Sydow, A., A Model to Identify Sources of Particles in Air. In: Proc. Symposium on Modelling, Analysis and Simulation, CESA '96 IMACS Multiconference, Lille, F., 2 (1996), 1218 - 1220.

2. Legg, B.J. and Raupach, M.R., Markov-Chain Simulation of Particle Dispersion in Inhomogeneous Flows: The Mean Drift Velocity Induced by a Gradient in Eulerian Velocity Variance. Boundary-Layer Meteorology, 24 (1982), 3 - 13.

3. Martens, R., Mameyer, K., Pfeffer, W., Haider, G., Morlock, G., Bestandsaufnahme und Bewertung der derzeit genutzten atmosphärischen Ausbreitungsmodelle. Gesellschaft für Reaktorsicherbeit (GRS) mbH, 1987, C140 - C148.

4. Obukhov, A.M., Description of Turbulence in Terms of Lagrangian Variables. Advances in Geophysics, 6 (1959), 113 - 116.

5. Simpson, D., Photochemical model calculations over Europe for two extended summer periods: 1985 and 1989. Model results and comparisons with observations. Atmospheric Environment, 27A (1993), 921 - 943.

6. Smith, F.B., Conditioned Particle Motion in a Homogeneous Turbulent Field. Atmospheric Environment, 2 (1968), 491 - 508.

7. Taylor, G.I. Diffusion by Continuous Movements, In: Proceedings London Mathematical Society, 20 (1921), 196 - 211.

TRAFFIC SIMULATION WITH THE AI CONTROLLED CASSANDRA SIMULATION SYSTEM*

A. Jávor, M. Benkő, Gy. Buzásy, A. Farkas, G. Szûcs
KFKI Research Institute for Measurement and Computing Techniques
H-1525 Budapest, P.O.Box 49, Hungary
E-mail: javor@sunserv.kfki.hu

Abstract. A simulation model based on Knowledge Attributed Petri Nets and a procedure using intelligent demons for optimizing urban traffic of large cities is presented. The model and the simulation experiment has been implemented in the CASSANDRA (Cognizant Adaptive Simulation System for Applications in Numerous Different Relevant Areas) simulation system. Finally some results obtained from the simulation run are provided.

Introduction

The investigation and optimization of the traffic of large cities is a highly complex problem. The representation of traffic dynamics where the fine structure of events influences the macrostructure of the dynamic trajectory of model behaviour has to be undertaken in such a way that the resolution of the model should provide sufficient information to get realistic results. On the other hand the model has to be constructed in such a way that computing time and storage space requirements should be within feasible limits. The aim of our investigations has been to decrease the air pollution caused by urban traffic [9] [10].

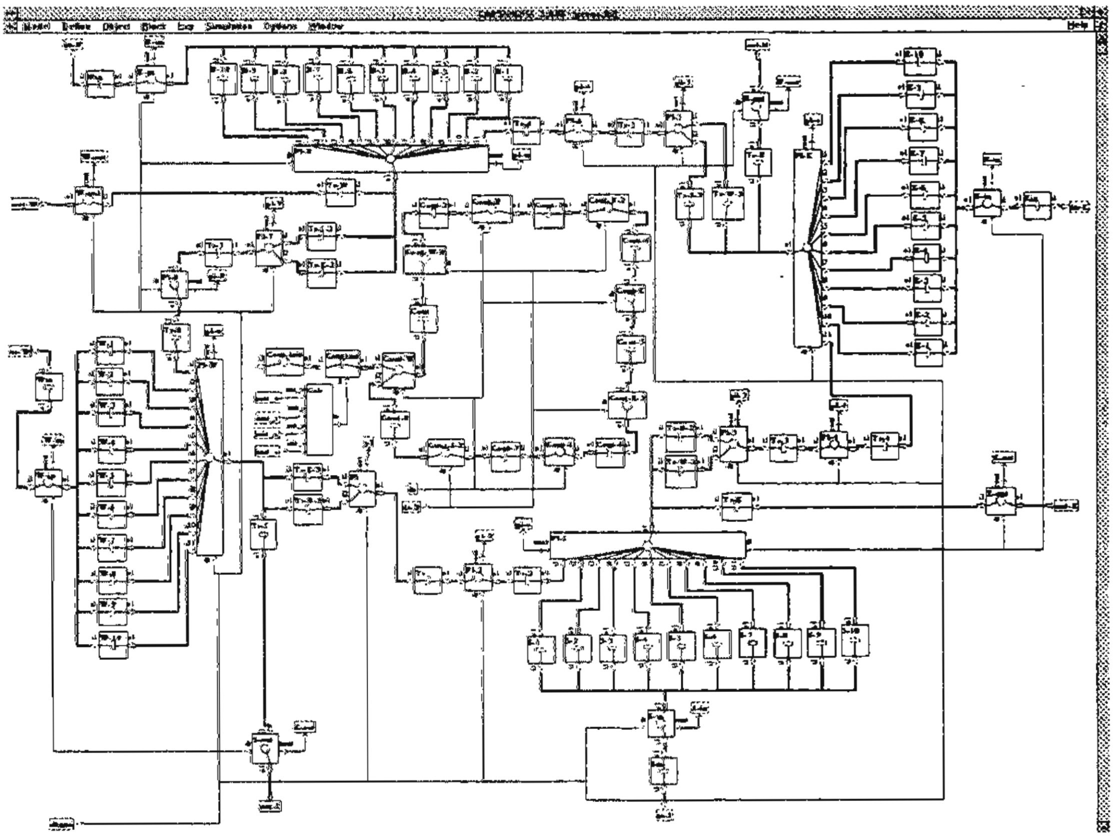
The search for optimal solutions would require a large number of simulation runs as an iterative process since the various measures that can be undertaken in order to improve the results (i.e. decrease the level of air pollution) as e.g. changing the relative phases of traffic lights, the determination of speed limits on various road sections, changing one way roads to two way or the other way round, altering the routes and time tables of buses, etc. In order to improve the efficiency of problem solving artificial intelligence has been used to control this mechanism [4].

The structure of the traffic model

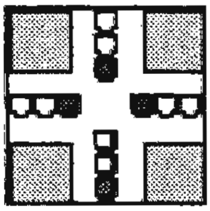
The traffic model has been constructed using the object oriented CASSANDRA simulation system, where the models were represented internally by high level Petri Nets [6] [7] in particular *Knowledge Attributed Petri Nets (KAPN)* [5]. As an example such an element from which the whole network has been constructed is shown in Figure 1, where in Figure 1/a the internal KAPN structure is shown, while in Figure 1/b the external representation [1].

As the methodology has been applied to describe and investigate the traffic of Budapest, the model became extremely large - even so that the representation encountered only the most important streets of the city - and the model structure (together with its animation) could only be represented on a large number of monitor screens. Therefore a paging system for displaying parts of the model during dynamic simulation has been implemented. The paging system showing the various sectors of the city can be seen in Figure 2 while in Figure 3 the chosen model segment is displayed.

* This work was partly sponsored by the COPERNICUS project "PATRIC" and partly by the National Scientific Research Fund, contract number: T 014017



a)



b)

Figure 1. Traffic light controlled crossing

The vehicles moving in the model have been represented by Knowledge Attributed Tokens. This enabled the possibility of representing various types of vehicles (4 and 2 stroke as well as diesel engined personal cars of various categories with catalyzer and without, lorries, buses, motorcycles).

This approach enabled the attachment of knowledge bases as attributes of the mobile vehicles in form of token attributes. That way the intended destinations of the vehicles (also enabling various routes in case of traffic jams or closed roads) could be attached to them. This provided a more refined description of the traffic conditions.

The representation of each individual vehicle by a mobile entity has obviously not been feasible, so various numbers of the different vehicle types have been represented in an aggregate form by macro entities made equivalent internally to tokens in a similar way as the macromolecular models have been introduced by Donald Greenspan [3].

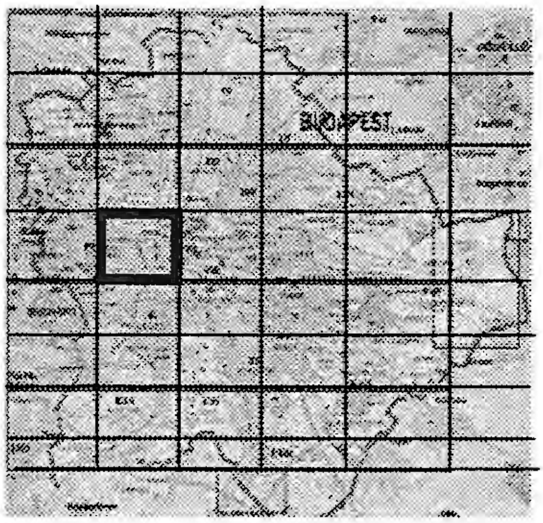


Figure 2. Paging structure for the model of Budapest

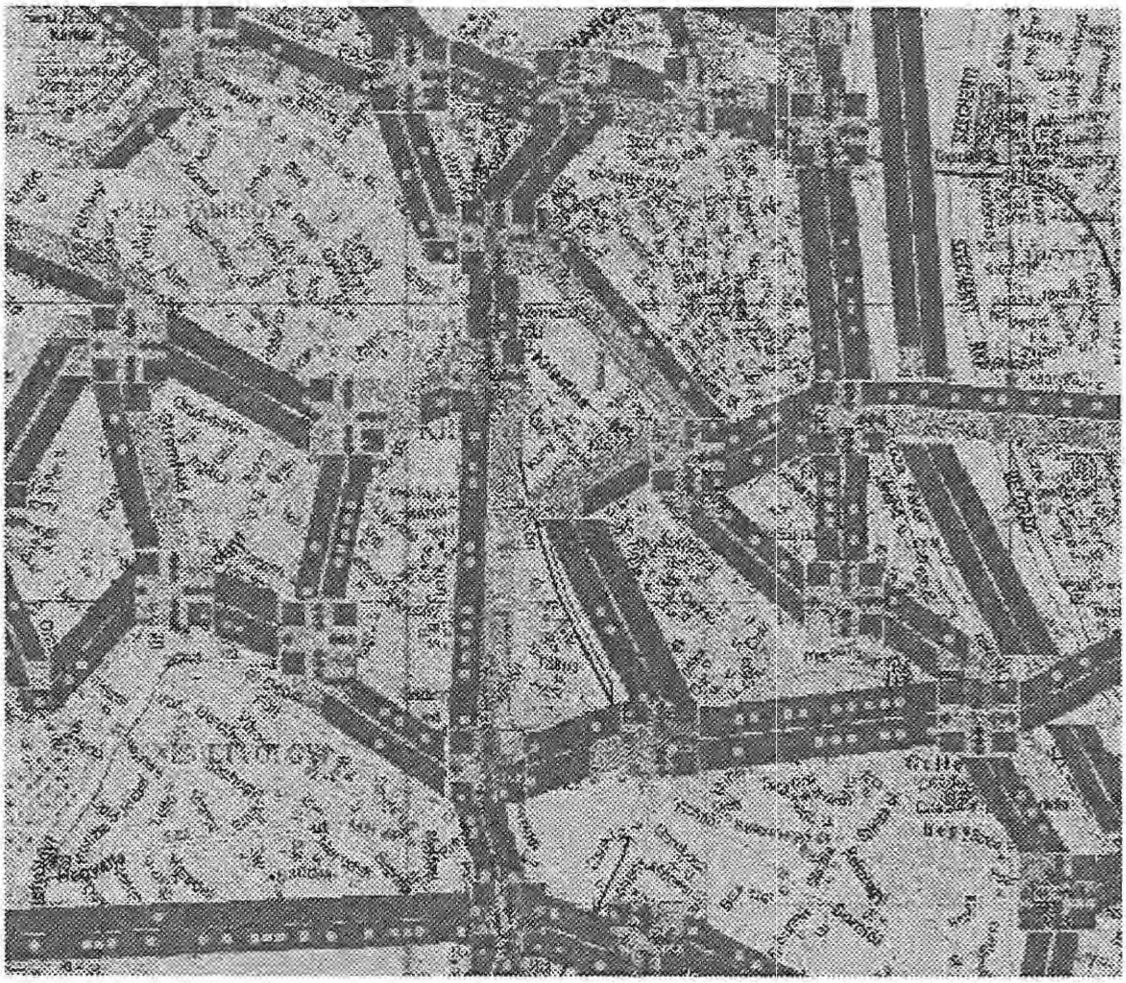


Figure 3. The chosen segment of the whole model indicated in Figure 2.

Optimization by intelligent demons

The emitted air pollution (NO_X, CO, HC) caused by the vehicles depends on the number of the various vehicle types as well as their speed on a given road section. In the model line sources corresponding to the road sections have been constructed [10] [2].

The optimization by intelligent demons during simulation run is undertaken in such a way that the trajectory of the behaviour of the simulated model parts (e.g. districts) are monitored by various intelligent demons and according to possible strategies and requirements stored in their knowledge bases the model segments are modified to decrease the emitted air pollution.

One of these optimization procedures where the relative phases of the traffic lights in a part of the city has been shifted by the controlling demons in order to decrease the emitted pollutants has been applied to the model segment shown in Figure 4, while the values obtained during simulation as well as the final optimal result is shown in Table 1. The reason for decreasing the emitted pollution is the more continuous and smooth flow of traffic. In general the decrease of the average speed increases the caused air pollution and in particular vehicles waiting at a red light emit the most pollutants.

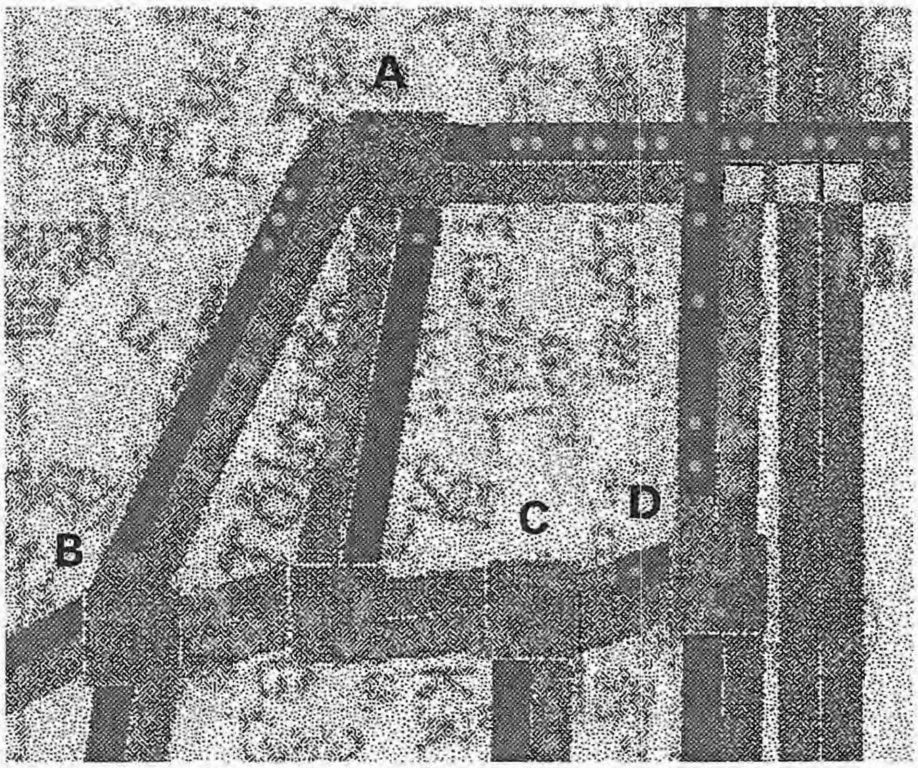


Figure 4. Traffic light optimized model segment

In Table 1 the first column shows the simulation time (where 1000 units are equivalent to one simulated hour). The second column shows the emission obtained from the model segment consisting of the line sources shown. The value Fi2 means the phase shift between the green lights of crossings A and B, Fi3 means the phase shift between A and C, while Fi4 corresponds to the phase shift between A and D. The T values mean the whole period time of the traffic light (in this particular case 1 min.). The minimal emission obtained during the search can be seen in the last column. In that case an approximate decrease of 12% emitted pollution could be achieved.

T	Emission	Fi2	Fi3	Fi4	Min. Emission
7500	5722	0	0	0	5722
10500	5575	T/2	0	0	5575
13500	5106	T/2	T/2	0	5106
16500	5218	T/2	T/2	T/2	5106
19500	5975	3T/4	T/2	0	5106
22500	5635	T/2	3T/4	0	5106
25500	6415	T/2	T/2	T/4	5106
28500	5964	5T/8	T/2	0	5106
32000	5936	T/2	5T/8	0	5106
35500	7140	T/2	T/2	T/8	5106
38500	5381	7T/16	T/2	0	5106
42000	5094	T/2	7T/16	0	5094

Table 1. Results of the optimization procedure

Conclusions

The methodology and its implementation has proved its value in practice. An important experience has been that in dealing with such extremely large models beyond efficient simulation algorithms powerful hardware tools are required. Therefore we had to implement our system on a parallel processing environment to increase efficiency. In this direction further investigations are conducted to attain further improvements. It seems that our methodology can be applied successfully to solve problems in other large cities. This can be done easily due to the object oriented modular simulation system and model architecture. The emission values are provided via computer network to simulate the imission data for the simulator developed by our partner in the PATRIC COPERNICUS project [8] [9].

References

1. Benkõ, M., Szûcs, G., An Object Set for Traffic Simulation. In: Proc. IMACS European Simulation Meeting on Simulation Tools and Applications, Gyõr, Hungary, 1985, 82-87.

2. Berlin Senate department for Environmental Protection. 1991. "KFZ-Belastbarkeitsanalyse der Berliner Innenstadt Luft- und Lärmbelastung"

3. Greenspan, D., Discrete Mathematical Physics and Particle Modeling. In: Simulation in Research and Development, (Ed.: Jávor, A.) North-Holland, 1985, 39-46.

4. Jávor, A., Demon Controlled Simulation. Mathematics and Computers in Simulation, 34 (1992) 283-296.

5. Jávor, A., Knowledge Attributed Petri Nets. Systems, Analysis, Modelling, Simulation, 13 (1993) 1/2, 5-12.

6. Jensen, G., Rozenberg, G., High-Level Petri Nets. Springer, 1991.

7. Peterson, J.L, Petri Net Theory and Modeling of Systems. Prentice Hall, 1981.

8. Sydow, A., Schmidt, Lux, Th., Schäfer, R.-P., Mieth, P.: Simulation of Air Pollutant Dispersion on Parallel Hardware. Simulation Practice and Theory 1(1993), 57-64.

9. Sydow, A., Lindemann, J., Lux, Th., Schäfer, R.-P.: A Concept for the Parallel Simulation of Traffic Flow, Traffic Emissions and Air Pollutants Dispersion in Urban Areas, IMACS European Simulation Meeting on Simulation Tools and Applications, 28-30 August, 1995, Györ, Hungary, 69-75.

10. Szûcs, G., Vigh, Á., Farkas, A., AI Controlled Traffic-Emission Line Source. In: Proc. of European Simulation Multiconference, Budapest, Hungary, 1996, 130-134.

EXAMPLES OF THE KNOWLEDGE BASES USED IN THE COMPUTER EXPERT SYSTEMS FOR ENVIRONMENTAL PROTECTION

Cezary Orłowski Ph. D. Andrzej Tubielewicz Prof.
Technical University of Gdańsk Narutowicza 11/12
Gdańsk E-mail: cor@sunrise.pg.gda.pl

Abstract. The paper presents expert system with examples of knowledge bases containing, formalised by syntactic rules, expert knowledge useful for making an analysis of the pollution caused by vehicular traffic in Gdansk. The bases are intended for application by taking advantage of NEXPERT OBJECT program in combination with which they create the problem oriented expert system.

Introduction.

The close proximity of the industrial and harbour areas of Gdańsk adjacent to the old town, the housing estates and recreational centres causes the condition of the natural environment of these areas as well as their main sources of pollution to have a significant effect upon the entire natural surroundings of the city and the life of its population. For this reason the data relevant to the degree of air pollution, ground pollution and water pollution of the areas analysed and the adjacent harbour sites are of interest in view of some general data relating to pollution of the city and an evaluation of the trends. The paper does not include pollution connected with the municipal vehicular traffic.

Knowledge base systems.

Computer systems with knowledge bases find greater and greater application in management aided programs. This is mainly due to a capability of transferring bases of knowledge witch is an essential element of every system and can be installed in a system shell base. It is also connected with a tendency towards decision decentralisation where the universality of knowledge and accessibility to it favour to make decision at any level of the management structure.

An area if great significance where such systems should be applied is ecology, and in particular, level pollution control. Factors that affect the level are: traffic intensity, road traffic signalling, urban infrastructure, terrain and climatic conditions. For their assessment it is necessary for people to have a through expert knowledge in various disciplines, e.q., traffic engineers, designers, architects, environmental protection specialists, and others.

Hence also the expert systems created on the base of the knowledge are conducive to decision making.

Construction of the computer knowledge base system requires an involvement of groups of people working in the system of knowledge base engineer - expert for preparing the basic element of the decision making system, which is the knowledge base. In general the knowledge bases are in form of a rule-objects structure. For this reason the knowledge base engineer must have much experience in the range of knowledge acquisition from an expert or experts. Creation of the data bases, elements of the knowledge bases, can also be possible at real time level, or it can result from simulation processes.

Other elements of the system, like the concluding mechanism and the explaining modules, are only a matter of an appropriate selection of the system shell. If the user is not determined to use a given system shell, the alternative is to create it using the program tools for the construction of the expert systems (e.q. Prolog).

Mutual penetration of decisions and undertaking the right ones requires the engagement of large systems with extensively developed knowledge bases. An example of setting up knowledge bases for such a system, and its utilisation is presented in the paper.

Characteristic of the problem.

Within the scope of the COPERNICUS program the universities of Germany, Hungary, and Poland build a computer decision system to control the pollution level caused by road traffic. The system is established on the basis of standard expert system shell, known as NEXPERT OBJECT 3.0.

A demand of the decisions - makers has created grounds for the system which would enable to estimate and give decisions related to an analysis of the pollution level caused by the vehicular traffic in Gdansk.

It has been assumed that the decisions will be undertaken at two stages:

- operational and strategic decisions;

- operational decisions:
changes in vehicular traffic how (prognosis of pollution level, analysis of traffic light signalling and directing road traffic to others areas);
- strategic decisions:
construction of new traffic solutions, effect on decisions in the range of urban development and with drawl of some groups of vehicles from road traffic.

Data for the needs of the system.

In course of accumulating information to build the knowledge base, advantage was taken of several independent sources. One of them was professional literature related to the problems of environmental protection, with a particular attention paid to the effect of pollution level resulting from vehicular traffic. Other sources of the data come from experts of institutions responsible for the protection of the environment. As a result of joint undertakings it was possible to collect data, which made contribution to set up the knowledge base. While elaborating the data attention was concentrated on the following problems:
- definition and characteristic of the obtained sources;
- identification of data and their analysis;
- logical organisation and structure of data;
- creation of sets based on knowledge.

In Gdansk the major sources of pollution are such industrial companies as, the oil refinery, the sulphur terminal, the shipyards, and private households, and the vehicular traffic. In order to obtain an indispensable knowledge about a given area it is necessary to collect data relating to an area of 50 x 50 km subdivided into smaller squares of 2 x 2 km. The meteorological data were measured in Gdansk at 3 points (the airport, the sea port, and the recreational areas). So far there have not been carried out measurements of the ozone layer.

During the first stage of creating the base, the data acquisition referred to a 4 km ^{2} central part of the town consisting of cross-roads of busy artery of the Gdansk agglomeration. In the future the area under investigation will gradually be expanded by longer distances.

In view of a great variety of the data sources, their various peculiarity, insufficient number of measuring check-points in Gdansk, and an incomplete range of measurements to obtain indispensable knowledge, it was difficult to carry out the task. However, the problems were partly overcome owing to the application of simulation and mathematical methods.

Type of data acquired.

In order to make the elaborated decision-making system based on the knowledge base function it should include three types of information:
- emission data;
- topographical data;
- meteorological data.

In course of our investigations it was possible to accumulate the following knowledge:
- communication systems (structure of cross-roads, traffic intensity at various times of the day and days of the well);
- pollution levels at 9 testing points (daily distribution of NO_2, SO_2 pollutants, dust and lead pollution in the months of February and July with respect to days and hours);
- vehicles according to group and age;
- meteorological data (temperature, pressure and windforce) appropriate for the data related to pollution;
- applicable functional relations between the traffic intensity and pollution;
- software for simulation and optimisation of traffic with variable lighting frequency (acquired).

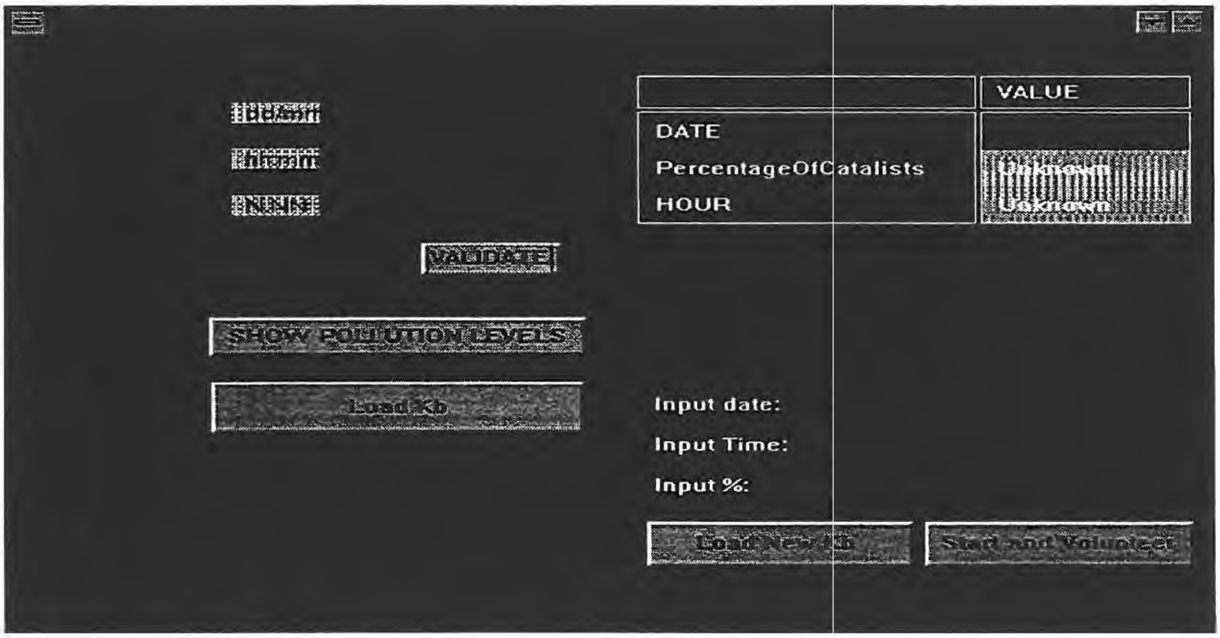
Out of Topographical data it was possible to distinguish the following ways of land use:
- areas of dense, medium and dispersed development,
- areas of forests,
- water reservoirs,
- agricultural fields.

Knowledge bases.

Knowledge dealing with road traffic and pollution has been structured making use of syntactic rules of the type:
IF (conditions, premises) THEN HYPOTHESIS and action in the case of fulfilling or not fulfilling the hypothesis.
Both the condition and actions appear in form of threes, objects, attribute, and value.
For the purpose of clarity of the knowledge base and easy access to it advantage has mainly been taken of text attributes in values described in written form. The way of making rules, as well as their editors has been carried out in compliance with the directors for NEXPERT OBJECTS.
The hypothesis are primarily put forward with regard to traffic levels and pollution magnitude.
The inferences illustrate the share of the respective rules in the inference process, or include suggestions to undertake adequate actions in the decision procedure.

Ruled knowledge bases.

The main part of the knowledge base is procedural knowledge extracted by experts by means of rules.
The rule bases have been divided, in the same way as in the case of original material, into modules including: traffic intensity, pollution levels, meteorological and topographical data.
The inference process, which concludes with a verification of the hypotheses concerning some probable pollution levels, puts forward adequate permissible vehicular traffic levels. The control process takes place at SMART ELEMENTS level. An example of software is presented in Fig.1.



	VALUE
DATE	
PercentageOfCatalists	Unknown
HOUR	Unknown

Input date:
Input Time:
Input %:

Fig. 1 An example of software

Knowledge base engaged in co-operation with data bases of client-server system.

Another type of knowledge bases are ones containing rules to cooperate with data bases connected to them. An example of such a base are bases of the dbf type where the knowledge accumulated refers both to the pollution level and the road traffic intensity at particular cross-roads.
The cooperation between the system and the knowledge bases occurs at the Client-server system level, where the system once plays the part of the server and another time the part of the client depending on whether the data are obtained from the system or sent away by it. The rules creating the elements of this base are connected with the records of the data bases. An example of such a connection is illustrated in. Fig.2.

Example of session utilising knowledge bases.

The process is controlled by activating the reference mechanism. Step one is responsible for the determination of the consultation data with the system (can be an immediate one). Next the system checks the data availability in the data bases and inquires about modification of vehicular groups with respect to the type of engine used. Finally some anticipated pollution levels are provided and violated pollution at respective cross-roads are given. The interaction proceeds with a possibility of an assessment of the particular hypotheses.
An example of session is given in Fig.3.

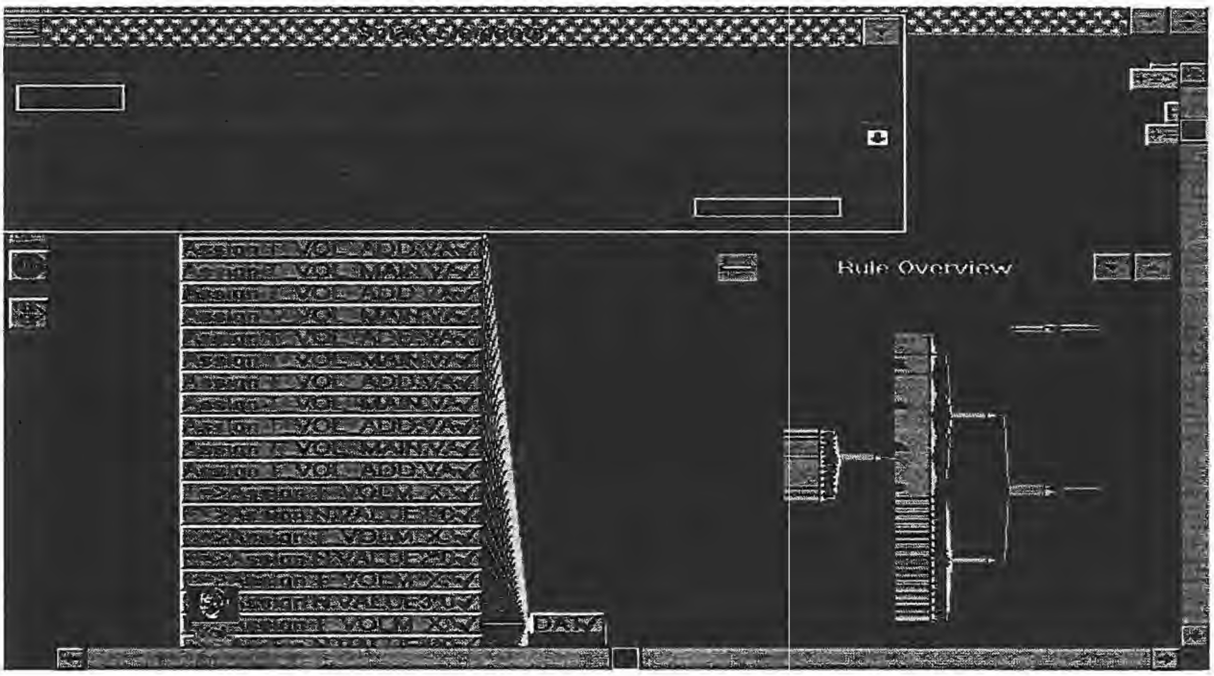


Fig.2 An example of such a connection.

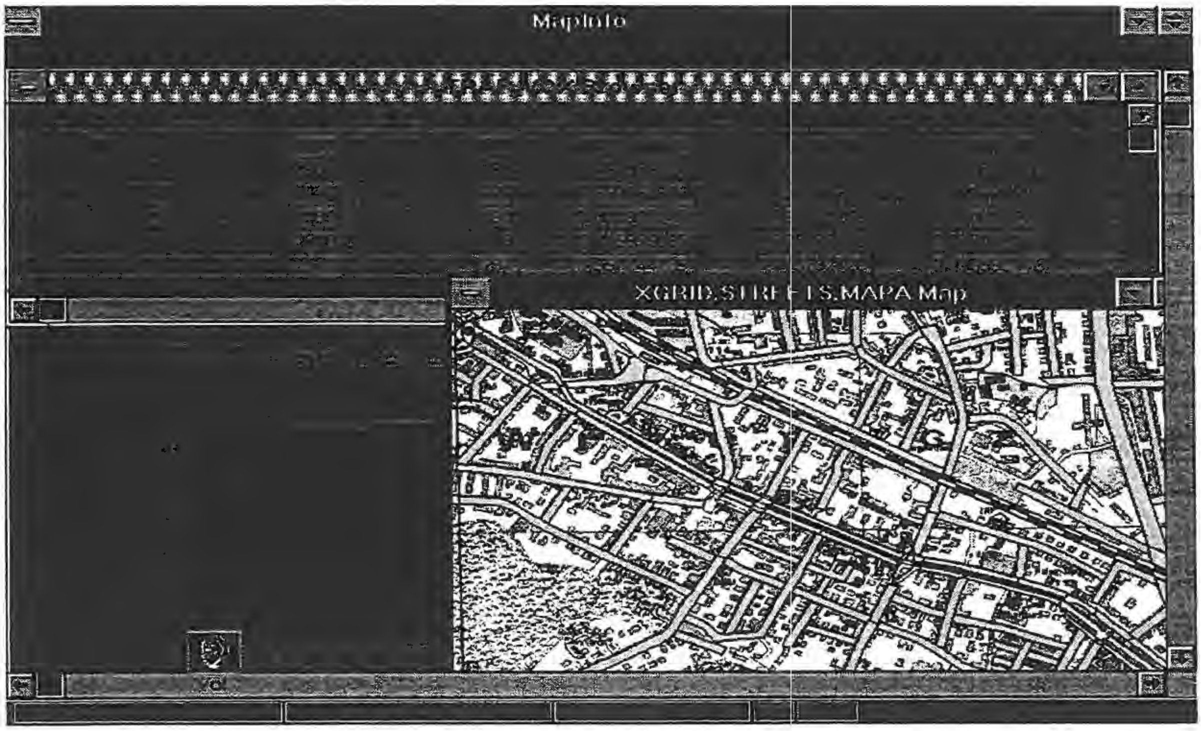


Fig.3. An example of session

Summing up.

1. The paper presents expert system useful for making an analysis of the pollution caused by vehicular traffic in Gdansk.
2. A decision making system co-operating with knowledge base being a combination of two different information technologies: standard and client-server system.
3. Advantage has been taken of Nexpert Objects system shell for the construction of the system and check-up of the process.

References.

1. McGovern and Mitchell. 1995 "Open Client/server Systems for Decision Support." *Computer Technology Review* (Spring/Summer): 9
2. Chiang and Roger H.L. 1995. "A knowledge-based system for performing reverse engineering of relational databases." *Decision Support Systems* Vol: 13 Iss: 3,4 (Mar.) : 295-312
3. Chatterjee, Abhirup; Segev, Arie. 1995." Rule based joins in heterogeneous databases." *Decision Support Systems* Vol: 13 Iss: 3,4 , (Mar.):313-333
4. Chang, Chen-Yuan; Chung, Chyan-Goe.1994 "A knowledge-based operation support system for network traffic management." *Decision Support Systems* Vol: 11 (Jan.): 25-36
5. Orłowski C. and Tubielewicz A.1996 " The Use of Decison Support System in the Control of Pollution caused by Traffic, *Proceedings, International Conference on Environment and Climate*, Roma :35

ANALYSIS OF TRAFFIC INDUCED AIR POLLUTION IN THE BUDAPEST REGION

A. Sydow, Th. Lux, R.-P. Schäfer and M. Schmidt
GMD Research Institute for Computer Architecture and Software Technology (GMD FIRST)
Rudower Chaussee 5, D-12489 Berlin, Germany

Abstract. Recent investigations have shown that vehicle traffic is the main source for emissions leading to summer smog. A study of the impact of traffic emission on urban air quality requires a complex simulation system for traffic induced air pollution. This paper presents results of an European cooperation project dealing with the development and application of such a simulation system which aims to support users in governmental administration and industry with forecasting and operative decision-making as well as short to long-term regional planning. The components of the simulation system are parallelly implemented simulation models for traffic flow, traffic emissions, meteorology, air pollutant transport and air chemistry, data bases for model input and simulation results, as well as a decision support tool. Results will be presented from the application of the demonstrator system in the region of Budapest.

1 Introduction

During hot mid-summer periods health-critical concentrations of surface-near ozone can be found in many urban and industrial areas around the world. Urban traffic emissions are actually the main cause for this so-called summer smog. Up to now, the only means for a local authority decision-maker to obtain information on traffic induced air pollution has been via measurements made from different points of the urban area. In order to obtain an overall picture of pollutant concentrations in the urban area and its surroundings and to simulate the consequences of different traffic control measures, a complex simulation system for traffic flow, traffic emissions and air pollution dispersion must be available. The system would provide a scenario analysis, on the basis of which, it will be possible to select the most appropriate measures for a specific situation.

Over recent years, increasing computing capacity has made it possible to develop numerical models which describe the transport and chemical transformation of air pollutants, taking into account complex flow and dispersion characteristics. On this basis, simulation systems for air pollution dispersion have been developed and used for scientific investigations at different application sites. On the other hand, various models for the dynamic flow of vehicle traffic have also been developed which enabled investigations to be carried out with the aim of avoiding traffic jams in urban areas by applying traffic control measures.

Section 2 presents a simulation system for traffic induced air pollution currently under development which consists of a traffic flow model basing on a special kind of high-level Petri nets [4], the air pollution simulation system DYMOS [10], [11] and an expert system for decision support [7]. In Section 3 results of the application of the simulation system for traffic induced air pollution in the region of Budapest are presented.

2 Simulation system for traffic induced air pollution

2.1 Traffic flow and emission model

A new approach to traffic flow modelling is the use of a special kind of high-level Petri nets. The Petri-net-based traffic flow model [4] has a discrete event description where the vehicles are to be represented by mobile entities. As the number of vehicles moving in a large city is extremely large it is envisaged to use a macro representation where a single mobile entity in the model may correspond to a given number of individual vehicles in the real system investigated. The simulation model will have two layers of representation. The external layer will have a graphic representation that corresponds to the field of traffic providing a natural view of the system. Internally for the efficient execution of the process of simulation the so-called Knowledge Attributed Petri Nets (KAPN) will be used that have already proved their value in various fields [3]. In this case a highly valuable feature is that the Knowledge Attributed Tokens (KAT) of KAPN - representing the mobile vehicles - may have knowledge bases attached to them providing for the convenient description of the destination and possible routings.

After the sequential implementation of the Petri-net-based traffic flow model test simulations has been carried out for the main traffic network of the internal part of Budapest. Because of the large number of vehicles the simulations require extensive computation time. Therefore the traffic flow model has been implemented on a dual-processor workstation and has been parallelized using the multithread approach.

Input data for the traffic flow model is the urban traffic demand (normally in form of origin-destination-matrices) and the traffic conditions in the model area (street network with junctions, position and state of traffic lights and existing traffic regulations such as one or two way traffic, number of lanes per direction, existence of bus lanes, etc.). Output data of the traffic flow model is the vehicle location within the road network as well as speed and acceleration values of the simulated vehicles.

Traffic emissions are influenced by a number of parameters, including internal parameters like engine size, engine type, different kinds of catalytic converters or exhaust gas emission reduction devices, the engine temperature as well as the determination of the traffic volume and the driving parameters of a single vehicle. To model traffic emissions the following input data have to be collected:

- vehicle types
- exhaust gas emission factors
- vehicle location within the road network
- vehicle speed and acceleration.

The percentages of the various vehicle types within the whole vehicle fleet of a model domain strongly depends on the country and continent considered for application. The exhaust gas emission factors are based on measurements of the exhaust gas emission of vehicles driving on standardized driving modes, which were, for instance, standardized for the European Union. Giving these inputs, the traffic emissions are calculated basing on the vehicle location within the road network as well as speed and acceleration values of the simulated vehicles received from the traffic flow model. The output of the traffic emission model are the summarized emissions of all vehicles driving on a street section to one line source for all line sources (street sections) within the model domain.

2.2 Air pollution models

For the simulation of the dispersion and chemical transformation of air pollutants emitted by traffic, industry and private households the DYMOS system [11] is used. The DYMOS system has been developed at GMD FIRST and is intended for mesoscale applications. The mesoscale deals with a range of 20 to 500 km including phenomena such as tropical cyclones, sea breeze flows, urban heat islands, mountain lee waves, local storms, etc. Typical applications are urban and industrial regions, metropolitan areas or coastal regions.

In close cooperation with the model developers, a first version of the DYMOS system was implemented, coupling the Eulerian atmospheric model REWIMET [2] with the air-chemistry model CBM-IV [1]. Unlike Eulerian models, which use a fixed grid structure in all three dimensions, REWIMET uses it only for the horizontal direction. Vertically, the model is divided into three layers derived from physical approximations. The air-chemistry model CBM-IV contains 34 species in 82 reaction equations for simulating the photochemical processes in the lower atmosphere that lead to the formation of ozone.

A second version of the DYMOS system incorporates an additional Eulerian atmospheric model (GESIMA [5]) with high vertical resolution. This system enables a better vertical resolution of the model domain which may be required for special applications. It also includes cloud physics. GESIMA is a truly three-dimensional, non-hydrostatic mesoscale model. The current version of DYMOS also contains a mesoscale Lagrangian model.

Due to their properties, Lagrangian models are especially suited for calculating air pollutant trajectories (e.g. transboundary analysis) and for localizing emission sources. One major difference between Eulerian and Lagrangian models is that Eulerian models calculate meteorology and transport variables for the model grid, whereas Lagrangian models only calculate particle or air column transport. The meteorological fields are input variables for Lagrangian models.

Various case studies of summer smog conditions in urban areas have been carried out using the DYMOS system. The Department of the Environment of the Berlin state government and the Ministry for Environment of the state Brandenburg commissioned summer smog analyses for the results of a measuring campaign carried out in July 1994 [6]. Greenpeace commissioned an analysis of the influence of emissions caused by traffic in Munich on the ozone concentration in the Munich area [9]. The analysis was for a typical mid-summer day in 1994.

A special problem for these types of application is determining the emission input which is caused mainly by traffic. Emission quantities, viewed as a dynamic function for a 24-hour period, have to be calculated for line sources representing a certain part of the street network in the model domain. These line-source emissions are computed from the fuel consumption of the vehicles on that line [8]. To date, fuel consumption has been calculated from statistical data (e.g. vehicle-counting measurements). In order to obtain more realistic dynamic traffic emission data, the traffic flow model described above and a traffic emission model are being coupled with the DYMOS system (see Figure 1).

Simulation runs with these complex atmospheric and traffic models require extensive computation time. In order to supply users with results of case studies within an acceptable time period or to enable a smog prediction to be made at all (computing time less than simulation period), the DYMOS system is already parallelized and implemented as message-passing version on several HPC platforms.

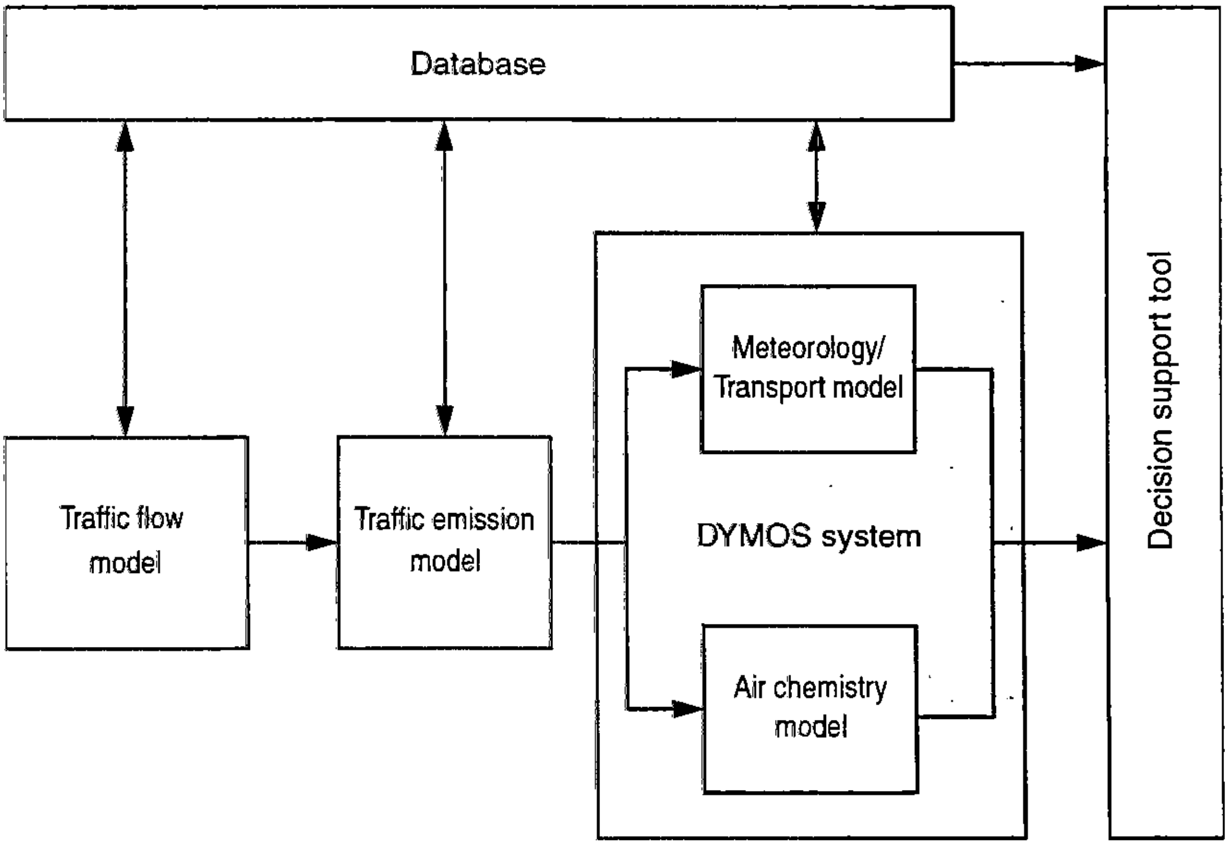


Fig. 1 Structure of the simulation system for traffic induced air pollution

2.3 Decision support tool

Decision-making problems of traffic authorities in urban areas can be divided in two categories:

- operational decision-making (control of traffic lights, control of traffic regulations, e.g. changes in the number of lanes per direction, building by-passes, etc.
- strategic decision-making (construction of new roads, investments in public transport systems, general urban planning, e.g. new industrial areas, etc.

In order to support the user of the integrated simulation system for traffic induced air pollution in these tasks, a decision support system has been developed and implemented. The decision support is given on the base of three kinds of data: input data for simulations, simulation results, and measurement data. The decision support system will be able to:

- visualize data (traffic flow, air pollutant dispersion, etc.)
- analyze the sensitivity of the model system (study the reaction of the model domain in terms of traffic flow, traffic emissions and air pollutants concentrations to changes in the model inputs and parameters)
- building a knowledge base of an expert system
- derive decision proposals by means of an expert system.

3 Application

3.1 Acquisition of input data

To run simulations using the integrated simulation system for traffic induced air pollution for the application area Budapest a variety of input data have to be collected. The Municipality of the City of Budapest greatly supports this work by supplying parts of the necessary input data for simulations. Other input data has been collected from topographical maps of the City of Budapest. The input data for the simulations can be divided into three groups.

- input data for the Petri-Net traffic flow model:
  This includes the traffic demand within the model domain and the data of the road network of Budapest. The traffic demand is determined from vehicle-counting measurements supplied by the Municipality of the City of Budapest. The vehicles are permanently counted at 29 measuring points in different directions at hourly intervals. The geometric structure of the road network has been manually determined from a topographic map of Budapest. The Environmental Department of the Municipality of the City of Budapest supplies with the coordinates of traffic lights on these streets. For the current simulations only the major streets are given.
- input data for the traffic emission model:
  As stated above this includes vehicle types, exhaust gas emission factors, vehicle location within the road network and vehicle speed and acceleration. The percentages of vehicle types are based on statistical data from the Municipality of the City of Budapest. The exhaust gas emission factors are supplied by TÜV Rheinland. Vehicle location, speed and acceleration are calculated by the traffic flow model.
- input data for the air pollution models:
  This includes meteorological data, topographic data (land use and surface elevation) and emission data (from traffic, industry and private households). The Meteorological Service of the Municipality of the City of Budapest supplies the meteorological data for the model domain. For any given period the daily variation of the surface temperature, wind speed and wind direction can be received on two measuring points. The topographic data have been collected and pre-processed for a model grid with a horizontal grid size of 1 km x 1 km. The data have been manually determined from a topographic map of Budapest. The Environmental Department of the Municipality of the City of Budapest supplies the emission data of industry sources and private households as mean yearly amounts per square kilometer for every relevant substance (VOC, NO_X, CO, SO_2). Using some factors for the daily variation and the weekly variation of these emissions, the emission amounts per year are processed into emission amounts per hour. The traffic emission data in form of line sources is calculated by the traffic flow and traffic emission models.

3.2 Simulation results

The simulation runs were carried out for a typical summer day in the Budapest region with temperatures between 18 - 30 °C as well as a northern geostrophic wind with a mean speed of 3 m/s. The meteorological situation of the considered day (July 5, 1995) is characterized by a long-term, high-pressure weather condition with strong insolation. The meteorological data as well as the initial and boundary values of the precursor substance concentrations are based on measurements of the Municipality of the City of Budapest on different measurements points in the city.

In addition to the measured meteorological data the traffic emissions were pre-processed for model input according to the spatial and temporal resolution of the DYMOS system. With respect to the physical background of the model system the update interval for the meteorological data and emission data was chosen to one hour. The horizontal grid resolution is 1 km x 1 km. The horizontal extent of the model domain is approx. 50 km x 50 km. The surface elevation is characterized by several hills, with an altitude between 450 - 550 m in western direction from the valley of the river Duna as well as smoother hills in the eastern part of the Budapest region. In Figure 3 the simulated surface-near ozone concentration in $\mu g/m^3$ is presented. The visualization of the simulation results shows a significant wide-area ozone trail on the lee-side of the urban region resulting from the anthropogenic emissions of ozone precursor substances in Budapest. The highest ozone concentrations simulated in the ozone plume reached a surface-near ozone level of 180 $\mu g/m^3$. In the city itself as well as near the motorways in the western parts of Budapest, where large amounts of exhaust gases were emitted, a decrease of the ozone concentrations can be observed. The reason for that phenomena are higher NO emissions (see Fig. 2), which reduce the ozone to oxygen by a chemical reaction. Using the integrated simulation system, scenario analysis can be carried out to study the consequences of possible measures for emission reduction on ozone production. The success of a specific measure under defined meteorological conditions could be evaluated. One scenario, that is expected to considerably reduce the ozone concentration, involves the substitution of the old eastern european 2-stroke and 4-stroke gasoline engines by engines with controlled catalytic converters as well as the limitation of the maximum speed on highways and motorways (e.g. 80 km/h). The simulation results for this scenario will be presented during the conference.

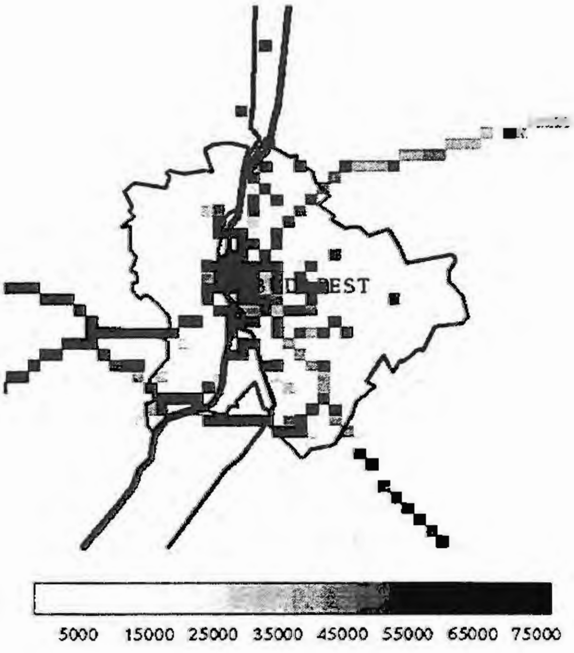


Fig. 2 NO_x grid emission [kg/year]

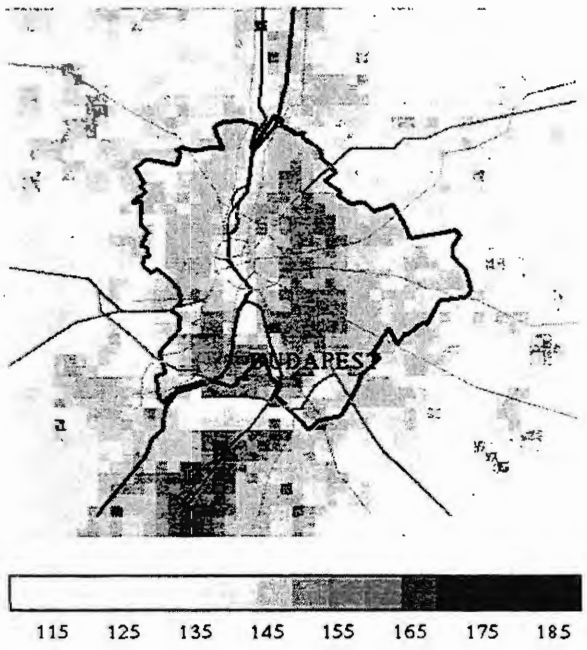


Fig. 3 Simulated surface-near ozone concentrations on 2 p.m. [$\mu g/m^3$]

4 Conclusions

An integrated simulation system for traffic induced air pollution has been developed consisting of parallelly implemented simulation models for traffic flow, traffic emissions, meteorology, air pollutant transport and air chemistry, data bases for model input and simulation results, as well as a decision support tool. Using the integrated simulation system, scenario analyses will be enabled to study the consequences of possible measures for traffic control (e.g. control of traffic lights and other controllable signs, closing parts of the city) and emission reduction (e.g. traffic ban for vehicles without catalytic converters) on the ozone concentration within the model domain. A first application of the simulation system has been carried out for the Budapest region. Future work will include the validation of the simulation results using measurements and comparisons with other simulation systems, the extension of the road network as traffic flow model domain and further improvements in determining the traffic demand.

Acknowledgment. Parts of this research has been carried out within the European cooperation project PATRIC. PATRIC has been funded by the European Commission, DG III, within the COPERNICUS Programme, under the reference number COP94 675. The Municipality of the City of Budapest greatly supports the research work by supplying necessary input data for simulations.

5 References

1 Gery, M.W., Whitten, G.Z. and Killus, J.P., Development and Testing of the CBM-IV for Urban and Regional Modeling. US Environmental Protection Agency, EPA-600/3-88-012, USA, 1988.

2 Heimann, D., Ein Dreischichten-Modell zur Berechnung mesoskaliger Wind- und Immissionsfelder über komplexem Gelände. Dissertation, Universität München, Germany, 1985.

3 Javor, A., Knowledge Attributed Petri Nets. Systems Analysis, Modelling, Simulation, 13 (1993) 1/2, 5-12.

4 Javor, A., Szücs, G., Traffic Simulation using AI. In: Proc. European Simulation Meeting on Simulation Tools and Applications, Györ, 1995.

5 Kapitza, H. and Eppel, D.P., The Non-Hydrostatic Mesoscale Model GESIMA. Part I: Dynamic Equations and Tests. Beitr. Phys. Atmosph., 65 (1992), 129-146.

6 Mieth, P., Unger, S., Estimation of the Influence of Anthropogenic Emissions of the City of Berlin on the Ozone Production. In: Proc. 5th International Conference on Atmospheric Sciences and Applications to Air Quality, Seattle, WA, 1996.

7 Orlowski, C., Tubielewicz, A., Selection of the Environmental Protection Data in the Building of Decision Systems. In: Proc. European Simulation Meeting on Simulation Tools and Applications, Györ, 1995.

8 Schäfer, R.-P. and Schmidt, M., Simulation of Traffic Emissions. In: Proc. Computational Engineering in Systems Applications - CESA'96 IMACS Multiconference, Part: Modelling Analysis and Simulation, Lille, 1996.

9 Smid, K., Cities Cause Ozone Smog in Rural Areas. GMD-Spiegel, Special: Simulation Models, Sankt Augustin, 1996.

10 Sydow, A., Lindemann, J., Lux, Th. and Schäfer, R.-P., A Concept for the Parallel Simulation of Traffic Flow, Traffic Emissions and Air Pollutants Dispersion in Urban Areas. In: Proc. European Simulation Meeting on Simulation Tools and Applications, Györ, 1995.

11 Sydow, A., Lux, Th., Schmidt, M. and Unger, S., Air Pollution Simulation Models for Air Quality Management and Risk Analysis. In Proc. Computational Engineering in Systems Applications - CESA'96 IMACS Multiconference, Part: Modelling Analysis and Simulation, Lille, 1996.

The choosen operating point is at pH = 7, and a feeding of 10g/l of glucose. Parameters values and determination of the equilibrium point can be found in [2].

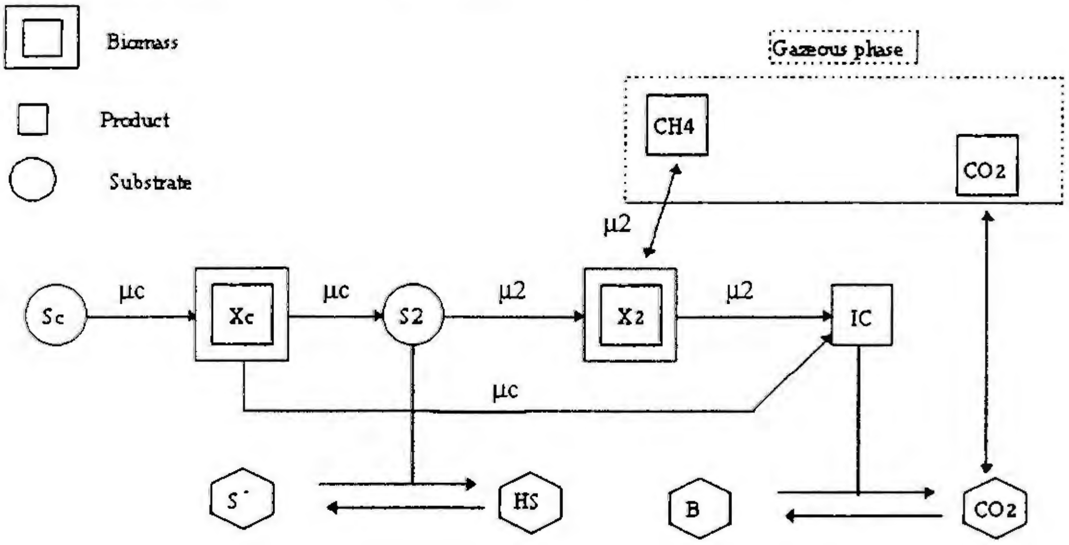


Figure 1 : Functional diagram of the anaerobic digestion

$$\begin{cases} (1) & 0 = H^{+}.S^{-} - K_{a}.HS \\ (2) & 0 = HS + S^{-} - S_{2} \\ (3) & 0 = H^{+}.B - K_{b}.CO_{2d} \\ (4) & 0 = B + CO_{2d} - IC \\ (5) & 0 = B + S^{-} - Z \end{cases}$$

$$\text{avec: } \mu_{c} = \frac{\mu_{c\,max}.S_{c}}{K_{sc} + S_{c} + \frac{S_{c}.HS}{K_{IC}}}$$

$$\mu_{2} = \frac{\mu_{2\,max}.HS}{K_{s2} + HS + \frac{HS^{2}}{K_{I2}}}$$

$$P_{CO_2} = \frac{CO_{2d}}{V.K_{H}}$$

$$\begin{cases} (6) & \frac{dX_{c}}{dt} = (\mu_{c} - D).X_{c} \\ (7) & \frac{dS_{c}}{dt} = -Y_{Sc/Xc}.\mu_{c}.X_{c} + D.(S_{cin} - S_{c}) \\ (8) & \frac{dX_{2}}{dt} = (\mu_{2} - D).X_{2} \\ (9) & \frac{dS_{2}}{dt} = -Y_{S2/X2}.\mu_{2}.X_{2} + Y_{S2/Xc}.\mu_{c}.X_{c} + D.(s_{2in} - S_{2}) \\ (10) & \frac{dZ}{dt} = D.(z_{in} + b_{inc} - Z) \\ (11) & \frac{dIC}{dt} = Y_{IC/X2}.\mu_{2}.X_{2} + Y_{IC/Xc}.\mu_{c}.X_{c} \\ & \quad - \frac{P_{CO2}}{P_{t} - P_{CO2}}.Y_{CH4/X2}.\mu_{2}.X_{2} + D.(IC_{in} + b_{inc} - IC) \\ (12) & Q_{CH4} = Y_{CH4/X2}.V_{s}.V.\mu_{2}.X_{2} \\ (13) & Q_{CO2} = \frac{P_{CO2}}{P_{t} - P_{CO2}}.Q_{CH4} \end{cases}$$

V : reactor liquid volume

$K_{a,b}$: acid-base equilibria constants

"$Y_{\alpha/\beta}$" : yield between α and β

$\mu_{c,2max}$: maximum growth rate for $X_{c,2}$

$K_{Sc,2}$: saturation constants of $X_{c,2}$ growth rates

$K_{Ic,2}$: inhibition constants of $X_{c,2}$ growth rates

Figure 2 : Anaerobic digestion model

The control problem

It is very well known, that anaerobic digestion is sensitive to pertubations on the organic load (input pollution) [11],[7]. Under normal operating conditions, the biomass degrades pollution into methan and carbon dioxide. As shown on figure 3, when a small amplitude overload occurs, the bioprocess is able to find a new equilibrium point for which there is still an organic pollution treatment. But when the amplitude is greater, the bioprocess goes toward a new operating point where no biomass is left in reactor and pollution is no more treated : it is the washout.

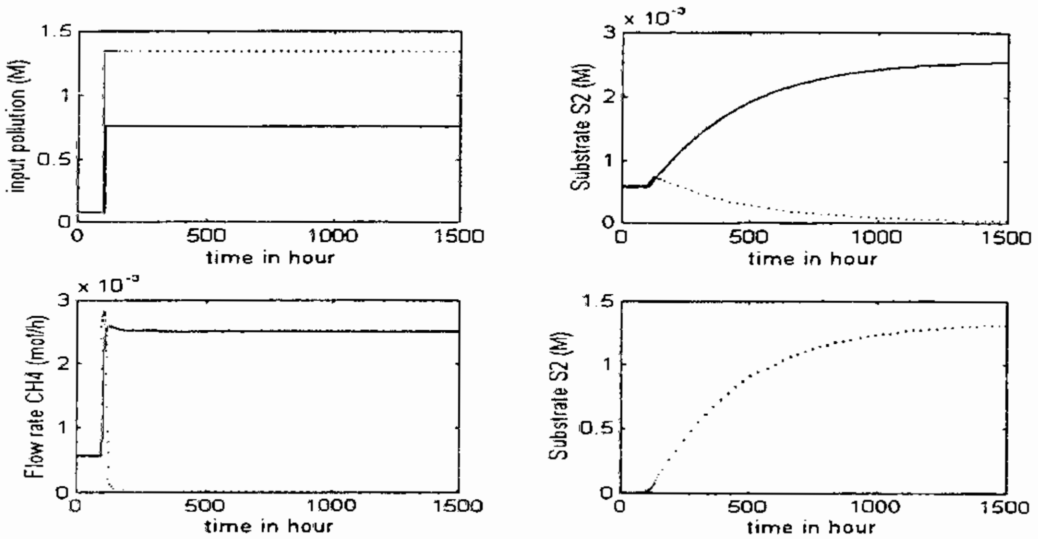


Figure 3 : Comparison of the system response when 2 different overloads occur

So from an automatic control point of view, the system is locally stable (small pertubations don't change process behaviour), but can be attracted toward an operating point which has to be avoided because there is no more organic load treatment ; see figure 4. A complete study of anaerobic digestion stability can be found in [2] based on the work of [4], [5], [6] and [12].

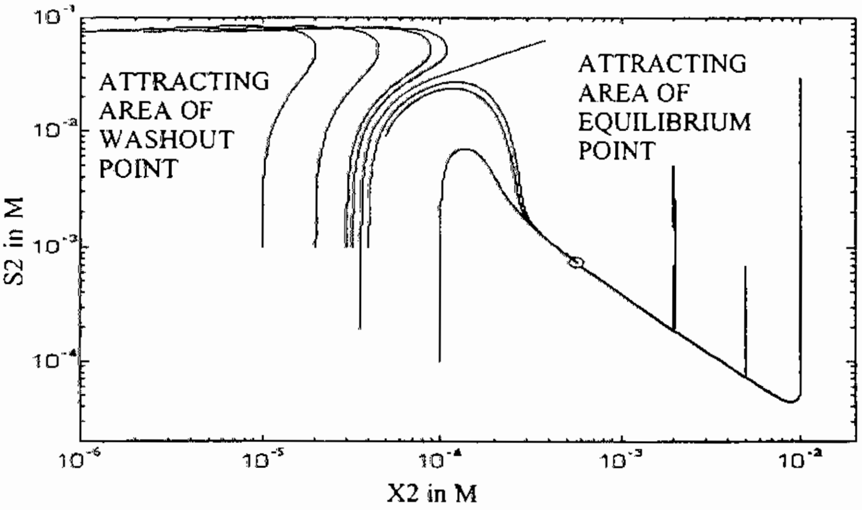


Figure 4 : Bioprocess trajectories in the state space

To ensure good operating conditions, it is essential to carry out an automatic control of the bioprocess. Its main goal is to stabilize the system around the desired equilibrium point, in order to guaranty organic load degradation. The different control strategies would be evaluated through the maximum overload the controlled system can accept before attraction towards washout. The second priority criterion is related to economical considerations : the output pollution has to stay under the maximum admissible level, and the CH4 production, which is sold, shouldn't decrease when overloads occur. The third criterion, with smallest priority is closed loop performances, especially the acceleration of the response time.

As shown in litterature [1], [8], [9], [10], different control strategies can fit the regulation problem. There are at least 4 possible measurements (bicarbonate alkalinity B, total acetic acid S_2, non ionized acetic acid HS, pH), 3 control actions (dilution rate D, addition of bicarbonate in the input b_{inc}, addition of a base), and many control laws (on/off control, PID, L/A control, adaptive linearizing control,...) have been proposed. So this represents 48 control strategies to be tested. In the results section, we will study and classify 7 among them with respect to previous criteria.

The first criterion, based on process stability, has to be quantified independently of the operating conditions (different type of perturbations e.g.) and the choosen control strategy. It means that to carry out this approach, we need a model describing bioprocess stability.

Stability modelling

The most common approach is to study process behaviour when a step load occurs and to find the maximum admissible amplitude A_{2max} before attraction towards washout appears. It seems important to us also to see what happens under square wave loads. In such a case, for a given load amplitude A_2, we search through realistic simulations the maximum time application of square wave, T, before bioprocess washout. Results analysis shows clearly a strong correlation between T and the input overload ($A_2.S_{2inm}$).

We conduct the same study for different equilibrium points corresponding to different dilution rates, D. Each time, the same types of correlation were found. A further analysis shows that, if we take into account the daily organic load ($D.A_2.S_{2inm}$), then the correlation f(1/T, $D.A_2.S_{2inm}$) is the same whatever D value is. The figure 5 shows the obtained limit between the production area and the attraction area to the washout.

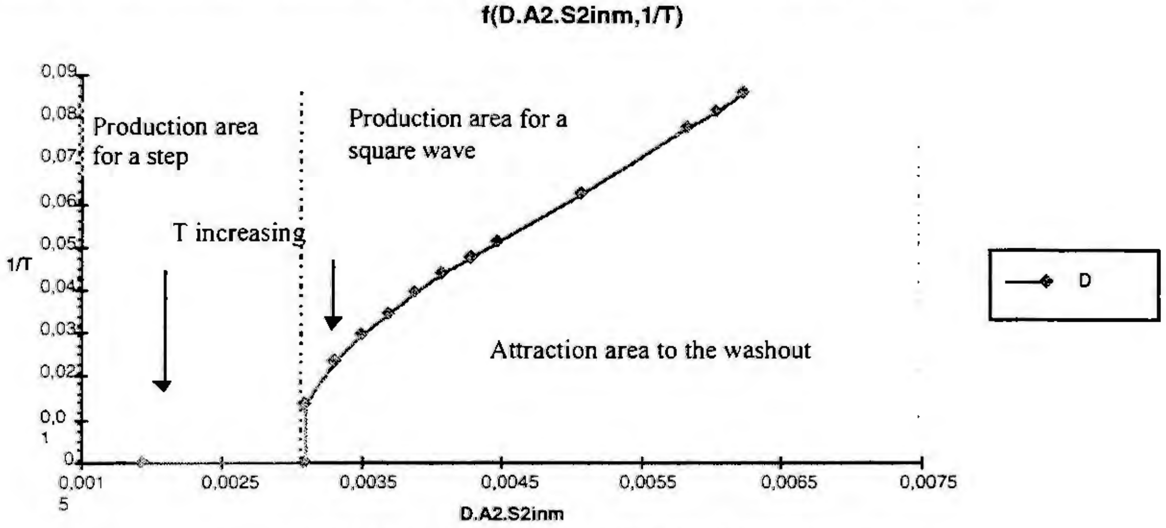


Figure 5 : Stability model for anaerobic digestion process

This experimental curve can be considered as a bioprocess stability model because it introduces an idea of distance from the attraction area to the washout limit. By knowing the dilution rate, the concentration variation and the date when the perturbation appeared, we can on line know where the bioprocess is located relatively to the limit. We can use this model either for step perturbations (infinite duration), either for square waves.

The part of the curve corresponding to square wave perturbations can be modelled by a linear equation :

$$\frac{1}{T} = 22, 797*\left(D.A_2.S_{2inm}\right) - 0, 0519$$

Then we can define stability margin as the distance from this equation, expressed as the inverse of a time. Thanks to its linearity, this model is easier to use than the complex non linear model of the anaerobic digestion. To quantify the first criterion, we will use the remaining time before reaching stability equation and for the choice of the most appropriate control strategy a pre-defined margin will help us to decide when the first criterion must have priority.

Results

We choose to test 2 control actions, the dilution rate D (modification of the input flow rate) and the addition of bicarbonate in the influent b_{inc}. This last action is limited because bicarbonate can't be removed from the process and there is a saturation due to precipitation problems. There exists today sensors for the 3 measurements we study : acetic acid substrate S_2, non ionized acetic acid HS and the producted bicarbonate B.

3 control laws were carried out : a classical PID controller, an L/A controller which take into account positivity constrainsts on variables, and an adaptive linearizing control which deals with process non linearities.

Each control strategy has been studied through realistic simulations on the model described in the first part of the paper [2]. The figure 6 summarizes the results obtained for each of the 3 criteria defined in the control problem section (maximum amplitude A_2 before attraction to washout, biogas production, response time on S_2). There are clearly 2 groups of control strategies depending on the action variable : with b_{inc} the biogas production and the volume of treated effluent is constant and stabilization is limited ; by using D as action variable the production decreases, but the bioprocess is always stable.

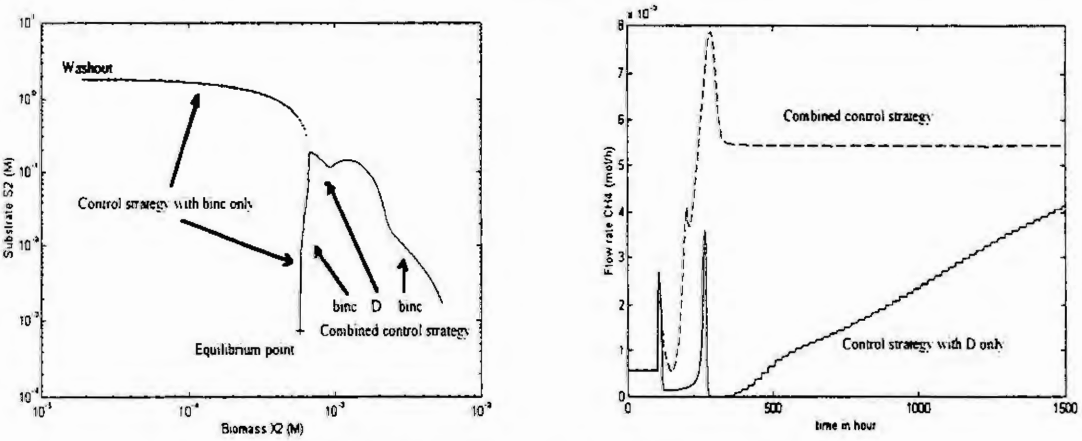
Action variable	Binc	Binc	Binc	D	D
Regulated variable	B : bicarbonate	B : bicarbonate	HS : acetate	S_2 : substrate	B:bicarbonate
Control law	L/A PID	NLL	L/A PID	L/A	L/A
Measurement	1	3	1	1	1
Measured variables	B	B CH_4 CO_2	HS	S_2	B
Response time S_2 (h)	450	450	400	200	50
Max. amplitude on A_2	21	22	24	Infinite	Infinite
Biogas production	Constant	Constant	Constant	Bad	Bad
Vol. of treated effluents (with respect to influent)	Totally (no storage)	Totally (no storage)	Totally (no storage)	Lower (storage tank)	Lower (storage tank)

Figure 6 : Comparison between different control strategies

For simulation tests, we consider a typical perturbation, a step on substrate S_2 with an amplitude $A_2=25$ which is destabilizing in open loop conditions. The figure 7 shows the results with three different control methods.

We used in the first case only the action variable b_{inc} and the bicarbonate measurement with an L/A control law : from the equilibrium point the bioprocess went to the washout. The perturbation exceeded the capacity of this action.

In the second case, we used the linear model of stability to select and apply the most appropriate control strategy at each time. We began like the first case but when the bioprocess is in danger of washing out (we reached the stability margin), we used a new control strategy : the regulation of bicarbonate with the dilution rate D. This is a stronger action variable. To ensure a good methan production and a better treated water volume, we recovered the first control strategy when the stability margin was away. In this second case we avoided the washout and the bioreactor went towards a new operating point.



Fugure 7 : Behaviour of a combined control strategy

To respect the first criterion (process stabilization), we could use only the regulation of bicarbonate with dilution rate D. But as shown in the figure 7, the second criterion (methan production and volume of treated water) is not optimized. By choosing the best control strategy on line, we can guarantee the stability and the maximum production, that is fulfil all the criteria simultaneously thanks to a combined control strategy.

Conclusions

In this work, we studied the control of anaerobic digestion processes. At first we established a 2 steps model describing complex organic matters degradation into CH_4 and CO_2. The second step was the methanogenesis, which dynamical behaviour is critical for operating conditions. This model was drawn from experimental work on a waste treatment process.

Our aim was to study how to improve the control efficiency of such a bioprocess. Indeed, many control strategies have been proposed in litterature, dedicated to peculiar problems. If their main goal was to stabilize the process, nevertheless it was tried actually to prove feasibility of a control law with available measurement and action. Our appraoch has been more general, and tried to compare all these control strategies, from a stabilization point of view but also from an economical point of view.

For this purpose, we introduced 3 criteria, related to closed loop performances, to biogas production and volume of treated effluent, and to bioprocess stability. In the last case, we proposed a new and original model describing stability limits of the anaerobic digestion, which was a function of time and pollution inputs. This model characterized the bioprocess itself and was independent from the choosen control strategy. We defined a distance from stability limits of the system, and stability margins. This model has been found through simulations, and in the future should be calculated directly from anaerobic digestion equations.

We applied criteria to select on line the most appropiate control strategy : when a perturbation occured and the bioprocess was far from stability limits, then a control strategy giving good economical results was chosen. If the stability was critical, then we selected another strategy to stay around the desired operating point. We proved that such a combined control strategy is much more efficient. It should be very interesting to study carefully the switching between the control strategies, and to take into account more criteria than in our work.

References

1. Béteau, J. F., Modélisation et commande d'un bioprocédé nidustriel de traitement des déchets urbains. PhD Thesis, Institut National Polytechnique de Grenoble, Grenoble, 1992.
2. Chaume, F., Application d'une stratégie de commande à un bioprocédé de traitement des effluents. DEA research report, Laboratoire d'Automatique de Grenoble, 1996.
3. Farza, M., Chéruy, A., CAMBIO : software for modelling and simulation of bioprocesses. CABIOS, 7 (1991), n°3, 327 - 336.
4. Fossard, A. J. and Normand-Cyrot, D., Systèmes non linéaires : II. Masson, Paris.
5. Gauthier, J.P., Structure des systèmes non linéaires. CNRS, Paris.
6. Lamnabhi-Lagarrigue, F., Analyse des systèmes non linéaires. Hermès, Paris
7. Moletta, R., Verrier, D. and Albagnac G., Dynamic modelling of anaerobic digestion. Water Research, 20 (1986), n°4, 427 - 434.
8. Renard, P., Dochain, D., Bastin, G., Naveau, H. and Nyns, E. J.,Adaptive Control of Anaerobic Digestion Processes : A Pilot Scale Application. Biotech. Bioeng., 31 (1988), 287 - 294.
9. Rozzi, A., Modelling and control of anaerobic digestion processes. Trans. Instr. on Measur. and Control, 6 (1984), n°3, 153 - 159.
10. Van Breusegem, V., Béteau, J. F., Tomei, M. C., Rozzi, A., Chéruy, A. and Bastin, G., Bicarbonate Control Strategies for Anaerobic Digestion Processes. In : Proc. 11th IFAC Worl Congress, Tallinn, Estonia, 11 (1990), 274 - 279.
11. Van den Heuvel, J. C. and Zoetemeyer, R. J., Stability of the Methane Reactor : A Simple model Including Substrate Inhibition and Cell Recycle. Process Biochemistry, May/June (1982), 14 - 19.
12. Vidyasagar, M., Nonlinear systems analysis. Prentice Hall, London.

THE PROCESS OF MODEL BUILDING AND SIMULATION OF ILL-DEFINED SYSTEMS: APPLICATION TO WASTEWATER TREATMENT

S. Kops, H. Vangheluwe, F. Claeys, F. Coen,
P. Vanrolleghem, Z. Yuan, G.C. Vansteenkiste
Biomath, University of Gent
Coupure Links 653
B-9000 Gent, Belgium

Abstract. In recent years, there has been a growing awareness of the ill-definedness of environmental processes. To provide a frame of reference for discussions regarding ill-defined systems, a taxonomy and terminology of the modelling and simulation of systems will be presented. Due to the complexity of ill-defined systems, it is not only necessary to describe the nature of models, but also to describe some procedures according to which the modelling will proceed. This will enable the modeller to obtain the model which best fits his goals (optimal model). For meaningful description of models, different model formalisms will be presented. Furthermore, modelling procedures will be described at a generic level and different model formalisms will be presented. Throughout this presentation, Waste-Water Treatment Plants and processes occurring within these plants will serve as illustrations of the definitions given.

Introduction

In recent years, mathematical models have gained importance in environmental studies. Environmental processes, such as those occurring in Waste-Water Treatment Plants (WWTP's), are often referred to as examples of ill-defined systems. Compared to the modelling of well-defined (*e.g.*, electrical, mechanical) systems, ill-defined systems modelling is more complex. In particular, the difficulty in choosing the "right" model is very apparent.

In the sequel a rigorous approach to modelling of ill-defined systems is presented. Illustrations are given for the case of WWTP's.

In order to develop a framework for the modelling of ill-defined systems, some definitions concerning modelling and simulation enterprise are given. Thereafter, a modelling procedure which may guide the modeller to find the "right" model, is presented. This modelling procedure consists of interactions between information sources and activities. These information sources and activities will be discussed. Models, the subset of the modelling enterprise, may be described in different formalisms. A common formalism classification will be presented.

Modelling and Simulation Concepts

One of the most important definitions in modelling and simulation is the definition of a *system*. A system is defined as a potential source of behaviour. It is *observable* when its behaviour can be transformed into data (information). Knowledge about given systems can be acquired through experiments. An *experiment* is defined as the process of causing (by known stimuli) and observing the behaviour of a system. In other words, given the inputs, the system outputs will be observed. In order to perform experiments on a system, its *experimental frame* has to be defined. The concept of experimental frame refers to a limited set of circumstances under which a system is to be observed or subjected to experimentation. As such, the experimental frame reflects the objectives of the experimenter who performs experiments on a system.

A way to organise collected knowledge about a system, *given its experimental frame*, is by means of *modelling* and *models*. In a very broad sense, a model is anything which is capable of generating behaviour resembling the behaviour of a system (given its experimental frame). In this paper only parametric models will be discussed. A parametric model, is a model consisting of *parameters*, where parameters are defined as constants or an experiment.

All systems may roughly be divided into two subclasses: *well-defined* systems and *ill-defined* systems. However, there is a fundamental problem in classifying systems. All information of a system can only be given by means of a model. Therefore, in order to classify, one has to define the properties of a model describing the system. Klir [1] solves this problem by defining epistemological levels at which the system may be observed and Zeigler [5] postulates a Base Model, a hypothetical model capable of describing

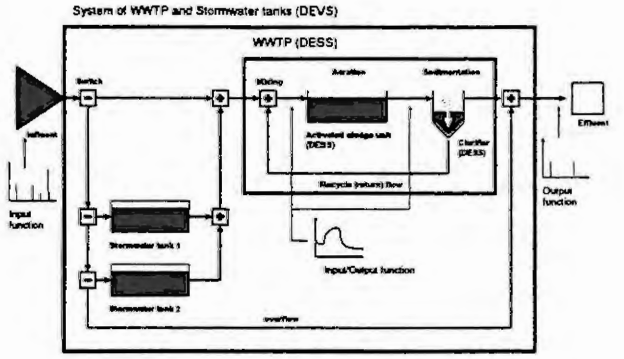


Figure 1: model formalisms in WWTP's

all possible behaviours of the system. Here, a well-defined system is a system of which it is possible to build, within an experimental frame and given the current *formalisms* and techniques, a structurally and behaviourally completely specified and, within a certain *accuracy*, *valid* model[1]. An ill-defined system can be defined as every system which is not a well-defined system.

The advantage of using a model to describe a system together with its experimental frame is that models are easier to experiment with. Experiments performed upon models are called *simulations*. Using simulations on a model instead of experiments on the system (and the experimental frame) it describes, has the advantage of *all* inputs and outputs being accessible. Hence, inputs or outputs can be applied to the model which lie outside the experimental frame of the system.

Despite the ease of use and general applicability of models and simulation, one has to be cautious in using these to describe ill-defined systems. Being ill-defined implies that there will always exist a chance that the behaviour (or structure) of the model describing the system will be different from the system itself, *i.e.*, that the model will not be valid.

Model Formalisms

Before describing the process of model building first model formalisms [4] will be discussed. During the whole process of model building model formalisms play an important role. In order to get an overview of the existing model formalisms, they are often being classified. However, the defined classes will never contain all formalisms. A well known classification is a classification given by [5]:

- *Differential Equation System Specification (DESS)*: Assumes continuous independent variables. The models are specified in differential equations which express the rate of change in the state variables.

- *Discrete Time System Specification (DTSS)*: Assumes discrete independent variables. The models are specified in difference equations which express the state transition from one time (and space) instant to the next.

- *Discrete Event System Specification (DEVS)*: Assumes a constant time base (the only independent variable) but the trajectories are piecewise constant, *i.e.*, the dependent variables remain constant for a variable period of time.

An example of the use of these three model formalisms in wastewater treatment is given in Figure 1. At the highest level, a system of WWTP's and storm-water tanks (buffer tanks) can be modelled using the DEVS formalism. Taking events (rain events, toxic discharges) into account, one must schedule the distribution of the wastewater loads between the WWTP and the tanks. In this case, a WWTP will be modeled as a "black box" with a given time delay and a given capacity. However, the WWTP can be seen as a system consisting of components such as aeration tanks and settling tanks. It may be modelled using the DESS formalism (PDE's or ODE's), or the DTSS formalism. Thus, within one system different formalisms may be used to describe its components and interactions.

[1] The concepts, formalisms and valid, will be explained in a later stage.

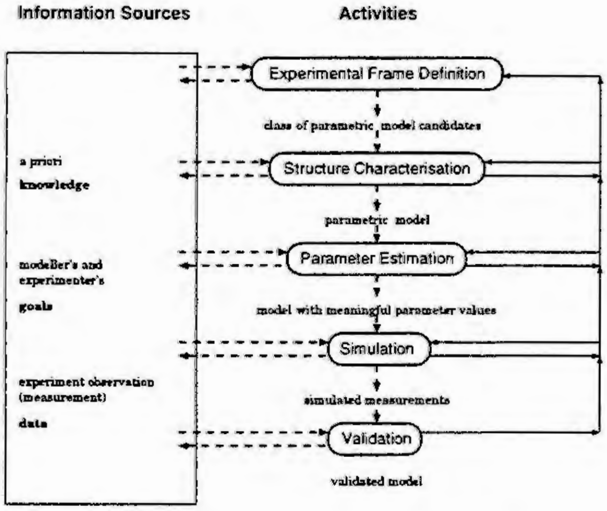


Figure 2: the process of model building

Another classification exists of (i) *deterministic*, (ii) *stochastic* and (iii) *probabilistic* classes. Whereas deterministic models originate from deductive modelling, probabilistic models originate from inductive modelling. Stochastic models can be seen as a combination of both. They assume one or more deterministic model attributes to have a statistical distribution.

The Process of Model Building

Roughly defined, the process of model building consists of constant interactions between *information sources* and *modelling activities*. A schematic representation of the process of model building is given in Figure 2.

From Figure 2 may be concluded that all activities have to be performed top down. However, a previously performed activity can be repeated depending on the outcome of the current activity. During the whole process of model building there exist constant interactions between activities and information sources. To ensure an equal importance of each information source, the modeller must justify each activity by using all information sources.

It has to be mentioned that since Figure 2 is a schematic representation (model) of a very complex and sometimes intuitive process (ill-defined system), it must not be taken for granted. Its only use lies in the rough guidelines it gives.

The next sections describe the informations sources and activities.

Information Sources

Three major information sources can be identified:

- *Goals and purposes*
- *A priori knowledge*
- *Experimental data*

The *goals and purposes* of the model user will orient the modelling process. The goals will, for example, determine the complexity of the model. The *a priori knowledge* available reflects the knowledge already gathered. This a priori knowledge often consists of (physical) "laws", such as the mass conservation law. A priori knowledge not always has to be developed within the (scientific) field in which the system to be described lies. Especially in environmental sciences, which is a rather new science, some of the "laws" used have been developed in other sciences and subsequently been adopted to model environmental systems. The *experimental data* are the observations of the systems behaviour. Experimental data may be collected to guide the modelling process or to validate the developed model.

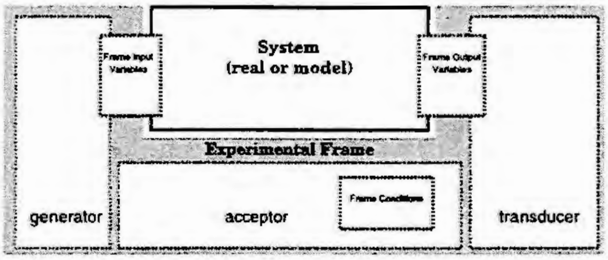


Figure 3: System versus Experimental Frame

Depending on the importance given to a priori knowledge and experimental data, two different modelling methodologies have been developed: *deductive* modelling and *inductive* modelling. Deductive modelling assumes a priori knowledge as the most important information source. Starting from the a priori knowledge, a deductive modeller will develop a model by using mathematical and logical deductions. Experimental data is only used to accept or reject the model or the hypotheses made during the modelling process. Inductive modelling assumes the observed behaviour to be the most important information source. Using the available data of a system, an inductive modeller will try to find a model describing the data. Often a part of the available data will be used to accept or reject the model or the hypotheses made during the modelling process.

Both deductive modelling and inductive modelling have a fundamental problem with the lack of a priori knowledge and data, respectively. Therefore, pure forms of both modelling approaches will seldom yield acceptable results in modelling ill-defined systems. This implies that a good mix between the two approaches is needed.

A good mix may only be obtained by (i) letting both the a priori knowledge and the experimental data influence the whole process of model building, and (ii) define a model formalism which can be used during both modelling approaches. A step towards a more general formalism is the concept of *uncertainty* [2]. Experience with the use of deductive modelling methodologies to model ill-defined systems has led to the conclusion that the systems behaviour (experimental data) could never be duplicated by the model output. Uncertainty was introduced as a measure for modelling errors such as errors in the model structure or in the parameter values. This implies that uncertainty can also be seen as a measure of the probability that a model output is a plausible system output. This probability is completely defined by the *probability density function (pdf)* of the model output. In order to obtain the model output pdf one must assume that the modelling errors in the model obtained by deductive modelling have an a priori statistical distribution. This distribution may be obtained using inductive techniques.

Modelling Activities

As mentioned before, five main modelling activities exist. All these activities will shortly be discussed below.

Experimental Frame Definition

As a model describes a system *together with its experimental frame*, the experimental frame definition must be the first modelling activity.

Referring to a limited set of circumstances under which a system is to be observed or subjected to experimentation, the experimental frame reflects the goals of the experimenter [5] (see Figure 3).

In its most basic form, an experimental frame consists of two sets of variables, the *frame input variables* and the *frame output variables*, matching the systems inputs and outputs, and a set of *frame conditions*, matching the conditions under which the systems behaviour is to be observed. On the input variable side, a *generator* describes the inputs or stimuli applied to the system or model during an experiment. On the output variable side, a *transducer* describes the transformations to be applied to the system outputs for meaningful interpretation. The *acceptor* will complete the experimental frame. It determines whether the system's output "fits" the conditions given.

For WWTP's inputs and outputs may, for example, respectively be defined as the incoming and outgoing wastewater flow, their substrate concentration and their dissolved oxygen concentration. However, the biomass within the aeration basin or the reaeration constant of the basin (in current models defined as a state variable and a parameter, respectively) may also be outputs. The conditions, which will be checked by the acceptor, may for example be defined as aerobic conditions. For example, consider the conditions being defined as aerobic and, during experimentation, it is measured that the oxygen concentration is zero. The acceptor will now conclude that the conditions do not hold at that particular time instant and the outputs measured in that time instant must not be taken into account during further analysis. In the observation of the behaviour of WWTP's often no generators are used; the inputs are not generated with known stimuli. An example in which transducers can be used is *risk analysis*. Here, the observer is, for example, only interested in all dissolved oxygen concentration below a certain threshold.

Structure Characterisation

Structure characterisation addresses the question of finding an adequate model structure.

Its aim is to reduce the class of models which are able to model the given system and experimental frame. The output class of structure characterisation may consist of more than one model structures, in which case each modelling activities (after structure characterisation) will be performed simultaneously for all model structures. In [3] some guiding principles for structure characterisation are given:

- *physicality*: A model must bare close resemblance to reality.
- *fit*: The experimental data available should be explained by the model as well as possible.
- *identifiability*: After structure characterisation, it must be possible to estimate the parameters.
- *parsimony*: The most simple explanation for phenomena must be found.
- *balanced accuracy*: The most useful model is often a balanced compromise of the previous principles.

Functions exist which, in order to reduce the class of models, will balance all or some of these principles. Such functions may, for example, be information criteria such as the AIC and BIC criteria. These criteria balance the fit and parsimony principles.

Furthermore, it has to be mentioned that structure characterisation issues cannot strictly be separated from other modelling activities, such as parameter estimation and validation.

Parameter Estimation

Parameter estimation will provide parameter values (and values for initial conditions) for a chosen model structure. Parameter estimation aims to reduce the class of parameters, using the fit principle defined previously. It is based on the optimisation of some criterion defining the goodness-of-fit such as Least Squares, Maximum Likelihood, *etc.*. Estimating parameters of ill-defined systems often results in a set of parameter values which have an (almost) equal goodness-of-fit criterion. A measure for the quantity of the set is parameter uncertainty. If the obtained set is very large one speaks of the parameters being *unidentifiable*. The identifiability may be *theoretical* or *practical*. Theoretical identifiability gives an answer whether, given the model structure, the parameters are identifiable, whereas practical identifiability gives an answer whether, given the available experimental data, the parameters are identifiable. Practical identifiability can be increased by increasing the information contained in experimental data using *optimal experimental design*.

Simulation

As defined earlier, simulation is an experiment performed on a model. In most sciences simulation consist of, given the inputs, determining the output trajectory of a model. However, it may also consist of obtaining information about the model. For example, the number of state variables. Simulation is performed by a *simulator*. A simulator consists of an *internal representation* and a *solver*. The internal representation is a representation of the model which can be understood by the solver. The solver "solves" the model, *i.e.*, generates data. Both the internal representation and solver depend on the model formalism [4]. Although these terms originate from computer science, they are generally applicable. For example, if one is able to solve a model analytically, the internal representation will be the model itself and the solver will be the person who solves the model.

Simulation is often said to be optimal if it can be done within a certain *accuracy* and *time instant*. Thus, within a given time instant, the simulator must provide output which resembles the "real" model output within a given accuracy. Both the accuracy and time instant depend on the goals and purposes of the modeller and user, the formalism and the current techniques.

Validation

Validation refers to the capability of the model to, up to a certain level and within a certain accuracy, replicate the system. Three different levels of model validity may be identified [3]

- *replicative:* the model is able to reproduce the input/output behaviour of the system (given an experimental frame).
- *predictive:* the model is able to be synchronised with the system into a state, from which unique prediction of future behaviour (thus outside the experimental frame) is possible.
- *structural:* the model can be shown to uniquely represent the internal workings of the system.

With each ascending level, the validity of the model becomes stronger causing a growth in the need for information and justification. This implies that, with each ascending level validation becomes harder.

As defined previously a model describing an ill-defined system will never be valid. One may only falsify the model. Therefore, from a practical point of view one should better use the term *falsification* when referring to the "validation" of ill-defined systems. A common error among scientists it that, when they could not falsify the model at the replicative level, resulting in a high confidence level, start to use it at the predictive level. However, at predictive level the confidence level may well be very low.

Conclusions

In recent years more and more different scientific fields have been involved in the modelling and simulation of systems. Moreover, the complexity of ill-defined systems has made it necessary to describe a procedure according to which the modelling will proceed.

By presenting both a taxonomy of modelling and simulation of systems and a modelling procedure, the above has provided a frame of reference for further discussions and research.

References

[1] KLIR, G. *Architecture of Systems Problem Solving.* Plenum Press, 1985.

[2] KREMER, J. Ecological implications of parameter uncertainty in stochastic simulation. *Ecological Modelling 18* (1983), 187–207.

[3] SPRIET, J. A., AND VANSTEENKISTE, G. C. *Computer-aided modelling and simulation.* Academic Press, London, 1982.

[4] VANGHELUWE, H. L., AND VANSTEENKISTE, G. A multi-paradigm modelling and simulation methodology: Formalisms and languages. In *Simulation in Industry, Proceedings 8th European Simulation Symposium* (Genoa Italy, October 24-26 1996), A. G. Bruzzone and K. E. J.H., Eds., vol. 2, Society for Computer Simulation International, pp. 168–172.

[5] ZEIGLER, B. *Theory of Modelling and Simulation.* John Wiley & Sons, New York, NJ, 1976.

MODEL BASED DESIGN OF A NOVEL PROCESS FOR AMMONIA REMOVAL FROM CONCENTRATED FLOWS

C. Hellinga, M.C.M. van Loosdrecht and J.J. Heijnen
Delft University of Technology, Dept. of Biochemical Engineering
Julianalaan 67, 2628 BC Delft, the Netherlands
+31 15 2785025 C. Hellinga@STM.TUDelft.NL

Abstract. A new full scale biological process for ammonia removal from flows containing hundreds to thousands grams NH_4^+ per liter has been developed at the Delft University of Technology. The SHARON process operates at a high temperature (30-40°C) and pH (7-8). Such conditions are favorable for high microbial specific growth rates, so that no sludge retention is required, and the process is carried out in a single, well mixed tank reactor. To reduce operating costs, ammonia oxidation is stopped at nitrite and denitrification is used for pH control by aerating intermittently. The number of laboratory experiments was kept minimal because unknown microbial kinetics and stoichiometry were identified as the bottleneck for process design in an early stage. Physico-chemical aspects were sufficiently covered in the literature. Moreover, the accurate formulation of microbial kinetics in terms of non-dissociated compounds helped to determine the pH dependency of this process with time varying concentrations efficiently. Process design was further enhanced by the construction of a dynamic model accounting for the simultaneous microbial conversion and chemical equilibrium reactions, and taking scale effects, especially for gas-liquid transfer, into account. In simulation, it was shown that with a simple, time based control strategy, process stability is very good despite large fluctuations in the expected influent flow rate and concentrations. A design was made for the full scale (1500 m^3), completely based on kinetic and stoichiometric parameters determined at 1.5 l. scale and model predictions. Construction of the plant starts in 1997. Grontmij consultancy has a patent pending on the process.

Introduction

New European legislation for reducing the total N-content in the effluent of wastewater treatment plants stimulates research towards new strategies for upgrading existing plants. Specific treatment of internal recycle flows gets much attention nowadays [6]. On request of the waterboard Zuid-Hollandse Eilanden en Waarden, our laboratory developed a completely new approach for treating recycled water from the sludge digesting unit at Sluisjesdijk, being part of the two stage A/B process (Absorption/Belebungsverfahren [2]) Dokhaven in Rotterdam (470.000 p.e.).This recycle flow accounts for about 20% of the ammonium load of the main plant. An evaluation of mass balances over the plant, that currently operates with an overloaded B-stage, showed that 85% ammonia reduction in this recycle flow will shift the operation of the B-stage from oxygen limited to ammonium limited, reducing total nitrogen in the effluent (currently 24 mg/l) by 25%.

Concepts of the SHARON process

The centrifuged effluent of the sludge digestion contains about 1 g NH_4^+ per liter, on molar basis a similar bicarbonate content, it has a temperature of 30 °C and a pH of 7.8-8.3. The SHARON process especially takes advantage of the high temperature, which enables high specific growth rates. Therefore, a certain biomass concentration level can be maintained in a single reactor of limited dimensions, without the need for sludge retention (figure 1). Experimentally, a maximum specific growth rate of 2.1 d^{-1} was measured for the ammonium oxidizing biomass at 35°C, so that, under actual conditions, around 1 day aerobic residence time is required. Note that sludge retention is here not essential to match certain effluent standards, as the effluent is fed back to the main process.

The high temperature has a second advantage. At ambient temperatures in wastewater treatment plants (in the Netherlands, typically 15 °C) nitrite oxidizers grow faster than ammonium oxidizers, which means that ammonium is completely oxidized to nitrate. However, the reverse is true at elevated temperatures, as can be seen in figure 2, which is based on temperature coefficients found by Hunik [5]. This means that by carefully selecting the residence time, nitrite oxidizers can be washed out, while ammonium oxidizers are retained in the reactor. From the reaction stoichiometry it follows that the oxygen uptake by the micro-organisms is reduced by 25% when the oxidation is stopped at nitrite:

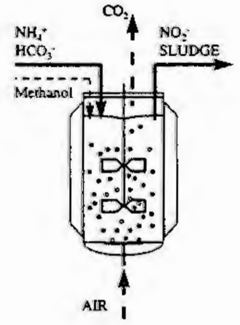


Figure 1. The SHARON process in a well mixed continuous flow reactor.

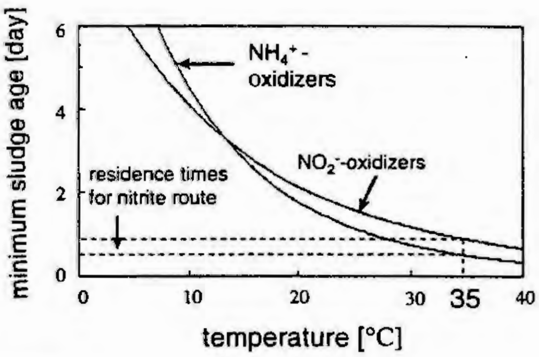


Figure 2 Maximum specific growth rates for ammonium and nitrite oxidizers as function of the temperature.

Nitrification

$NH_4^+ + 1.5\ O_2 \rightarrow NO_2^- + H_2O + 2\ H^+$ *(Nitrosomonas)* (1)
$NO_2^- + 0.5\ O_2 \rightarrow NO_3^-$ *(Nitrobacter)* (2)
$NH_4^+ + 2\ \ O_2 \rightarrow NO_3^- + H_2O + 2\ H^+$ (3)

A second measure for reducing variable costs is to use denitrification for pH control. During nitrification, 2 moles H^+ are produced for every mole NH_4^+ oxidized:

Denitrification

$NO_2^- + 0.5\ \ CH_3OH + 0.5\ \ \ CO_2 \rightarrow 0.5\ N_2 + HCO_3^- + 0.5\ \ \ H_2O$ (4)
$NO_3^- + 0.83\ CH_3OH + 0.167\ CO_2 \rightarrow 0.5\ N_2 + HCO_3^- + 1.167\ H_2O$ (5)

This type of influent contains typically almost equimolar amounts of ammonium and bicarbonate. When protons are released, the concentration of solved CO_2 increases due to the shift in the bicarbonate equilibrium. CO_2 can be stripped by the air used for oxygen supply. Under practical conditions, about 50% of the protons produced by nitrification can be neutralized by stripping CO_2. Commonly, this is not enough to maintain a sufficiently high pH for the required conversion, as will be explained later. Of course, base addition is an option. Here, denitrification is used instead, being more cost effective. From equation 4 it follows that for each mole of nitrite, one mole H^+ is consumed. The heterotrophic denitrification requires an organic energy and carbon source, which is only in very limited amounts available in the centrifugate. Therefore, here it was decided to use methanol as a relatively cheap "COD" source. If in addition to the equations 4 and 5, biomass formation is taken into account, full denitrification requires about 3.8 kg methanol per kg NO_3^--N, and only 2.3 kg per kg NO_2^-[7]. Due to the nitrite route, 40% savings on methanol supply are obtained. Compared to base addition, denitrification is about 40-50 % cheaper for the same pH decrease at the current world market price levels. To what extent denitrification is required depends on the buffering capacity of the medium and on the required pH (hence conversion rate). This is essentially a cost optimization problem.

Research program for process design

Because of the non-conventional process conditions, a research program has been carried out to determine microbial kinetics and stoichiometry. The high dilution rate puts a selection pressure upon the system for fast growing micro-organisms, rather than for organisms with a high affinity for substrate, which is the case in conventional installations operating at low nutrient concentration levels, so that literature data, especially on nitrifying microorganisms, are not necessarily applicable here. For reasons of low investment costs, a single reactor was chosen with intermittent aeration to enhance both nitrification and denitrification. As a result, the pH varies about 1 unit during operation, so that the dynamic response to pH variations had to be taken into account, especially since the non-dissociated forms of the N-compounds are considered to be metabolically relevant [1].

Experiments were carried out for approximately 2 years in 1-2 liter fermentors, operated in continuous mode with actual centrifugate from the sludge digesting unit. From the observed turnover, stoichiometric parameters could be obtained. For determining kinetic parameters, samples taken from these reactors were subjected to respirometric experiments [4,7]. For design of the full scale aerated reactor (with a volume of 1500 m^3) no additional (pilot scale) experiments were considered necessary, because reactor specifications for scaling up can be obtained from literature. Moreover, the biomass is growing in very small lumps of cells (not being selected for settling properties), which means that significantly different floc sizes at small and full scale are not foreseen. Estimations for the influence on oxygen supply and stripping of CO_2 (pH regulation) were obtained with some straightforward calculations. Literature data were also used to estimate the net heat production of the conversion.

During the experimental program, much emphasis was put on the accurate formulation of the kinetic expressions. For systems with varying pH, it is particularly important to identify the biochemical active form of compounds that appear both protonized or non-protonized. Via the chemical equilibrium equations, concentration levels of the active compounds can then be calculated as a function of the pH, which reduces the number of experiments considerably. It is important to notice that on the one hand experiments were used to identify model parameters, and that on the other hand the model formulation itself had a great impact on the (number of) experiments.

Eventually, the identified microbial kinetics and stoichiometry, combined with the chemical equilibrium reactions and gas-liquid mass transfer, were used in a dynamic process model, implemented in SIMULINK. The model was used among other things, for determining the necessary denitrification rate (hence methanol addition) at full scale operation, given the drastic daily changes in the influent load and using a straightforward control strategy.

The model

The process model consists of 13 non-linear differential equations. Three of them account for the accumulation of O_2, CO_2 and N_2 in the gas phase, the rest for liquid phase accumulation of (lumped) compounds. Both the liquid and the gas phase were considered to be well mixed and to have a constant volume. The differential equations then take the general form:

$$\frac{dC_i^J}{dt} = D^J * (C_{i,in}^J - C_i^J) + r_i^J - \frac{\phi_i^{JK}}{V_J} \tag{6}$$

The superscript J denotes the phase (gas or liquid), the subscript i the compound. The specific conversion rates r [mol/m^3/s] only have non-zero values in the liquid phase. D is the dilution rate [1/s] and ϕ^{JK} the mass transfer rate through the interphase of phase J to phase K [mol/s] (only relevant for O_2, N_2 and CO_2). In the liquid phase, only the relatively slow microbial conversions were expressed as differential equations. Therefore, the conversions of the compounds that take part in chemical equilibria (NH_3/NH_4^+, NO_2^-/HNO_2 and $CO_2/HCO_3^-/CO_3^{2-}$) were described in terms of the lumped compounds NH_{3H} (C_{NH3H}=C_{NH3}+C_{NH4}), NO_{2H} (C_{NO2H}=C_{NO2} + C_{HNO2}) and CO_{2HO} (C_{CO2HO}=C_{CO2}+C_{HCO3}+C_{CO3}).

From the concentrations of the three lumped compounds, the concentrations of the individual charged and non-charged components (being relevant in the kinetic expressions) were calculated iteratively solving the set of equations formed by the chemical equilibrium equations:

$$C_{HCO_3^-} = \frac{C_{CO_{2HO}}}{1 + K_{CO_3} / C_{H^+} + C_{H^+} / K_{CO_2}} \tag{7}$$

$$C_{CO_3^{2-}} = \frac{C_{CO_{2HO}}}{1 + C_{H^+} / K_{CO_3} + C_{H^+}^2 / K_{CO_3} / K_{CO_2}} \tag{8}$$

$$C_{NO_2^-} = \frac{C_{NO_{2H}}}{C_{H^+} / K_{HNO_2} + 1} \tag{9}$$

$$C_{NH_4^+} = \frac{C_{NH_{3H}}}{K_{NH_3} / C_{H^+} + 1} \qquad (10)$$

and the charge balance:

$$\Delta = -C_{H^+} + K_W / C_{H^+} + C_{HCO_3^-} + 2 * C_{CO_3^{2-}} + C_{NO_2^-} - C_{NH_4^+} - C_{Z^+} \qquad (11)$$

. Δ denotes the gap in the charge balance. The equilibrium constants for chemical equilibrium i are indicated with K_i. C_{Z+} is a net concentration of additional positively charged ions, present in the influent, that was calculated by solving these equations for the influent composition, with known pH. In fact it is assumed this way that the difference in true pH and the pH calculated for the acid-base equilibria, is due to the effect of strong acids/bases, which may not be completely true. The set of equations (7-11) was solved with the Newton-Raphson algorithm (the first derivative $d\Delta/dC_H$, essential for the algorithm, can easily be obtained analytically from the above equations). For a starting value for C_{H+} of 10^{-15}, Δ/C_{H+} reduces to values below 10^{-4} within 5-6 steps. This algorithm was programmed in C, and implemented as a SIMULINK S-function, accounting only for about 10% of the simulation time. (On a Pentium-60, two days of operation are simulated in about 1 minute.)

Microbial kinetics were modeled with the following general structure including Monod terms for substrate limitation and inhibition terms:

$$\mu^i = \prod_{j=1}^{n} \left(\frac{C_j}{K_j^i + C_j}\right) \prod_{k=1}^{m} \left(\frac{KI_k^i}{KI_k^i + C_k}\right) \qquad (12)$$

The superscript i denotes the microbial population. 5 kinetic expressions were used for ammonium oxidizers, nitrite oxidizers, denitrifyers for nitrate and nitrite, and heterotrophic biomass for the conversion of excess methanol under aerobic conditions. j denotes the substrate, K_j^i the Monod constant for population i, growing on substrate j, and KI_k^i the inhibition constant of population i for compound k.

The important equations for the growth of ammonium and nitrite oxidizers read explicitly as:

$$\mu^{amm} = \mu_{max}^{amm} * \frac{C_{NH_3}}{K_{NH_3}^{amm} + C_{NH_3}} * \frac{C_{O_2}}{K_{O_2}^{amm} + C_{O_2}} * \frac{K_{I,HNO2}^{amm}}{K_{I,HNO_2}^{amm} + C_{HNO_2}} \qquad (13)$$

$$\mu^{nit} = \mu_{max}^{nit} * \frac{C_{HNO_2}}{K_{HNO_2}^{nit} + C_{HNO_2}} * \frac{C_{O_2}}{K_{O_2}^{nit} + C_{O_2}} \qquad (14)$$

In equation 13, NH_3 is used as the actual substrate for ammonium oxidizers and HNO_2 as the inhibiting component [1]. Experimentally, it was shown that K_{NH3} is indeed constant at about 0.5 mol/m^3 for the pH range 6.5-8.5 at 30°C [4]. At 35°C the same value can be used [7]. At typical effluent (and thus medium) concentrations of 100 mg NH_4^+-N per liter the NH_3 concentration ranges from 0.11 mg/l at pH 6 to 10.1 mg/l at pH 8. No inhibition of ammonia was found up to concentrations of 6000 mg NH_4^+-N/l (pH 7, 40°C). The inhibition constant for nitrous acid was established at 0.2 mg HNO_2-N/l so that at higher pH (the HNO_2 concentration at nitrite concentrations of about 300 mg/l ranges from 0.53 to 0.00525 mg/l in the pH range of 6-8) the inhibition effect of HNO_2 is limited. As nitrite oxidizers use HNO_2 as well [1], with K_{HNO2}=0.26 mg HNO_2/l [9], their growth rate is in between 0.17 d^{-1} and 0.02 d^{-1} in the pH range 7-8. From these data it follows that a high pH should be maintained. pH 7 and above in combination with a liquid residence time of 1 day provides good conditions for a fast growing population of ammonium oxidizers and for repressing the growth of nitrite oxidizers. For K_{O2} a value of 1.45 mg/l was found, which is rather high, but probably caused by the fact that the experiments were carried out at high dissolved oxygen concentrations (>6 mg/l).

In this system with fast growing biomass, the influence of decay or maintenance was neglected, so that the conversion rates of all involved (lumped) compounds are stoichiometrically related to these 5 growth rates:

$$\underline{r} = A * \underline{\mu} \qquad (15)$$

quantity		influent	effluent
NH_4^+	[mg N/l]	972	130
NO_2^-	[mg N/l]	-	345
NO_3^-	[mg N/l]	-	0.9
HCO_3^-	[mmol/l]	72.3	7.3
HCO_3^-/NH_4^+	[mol/mol]	1.1	0.78
pH	[-]	8.3	7.4
Temp	[°C]	30	35
Act.biom.	[g/l]	-	0.33

Table 1. Predicted effluent composition for actual influent data at full scale (year averages)

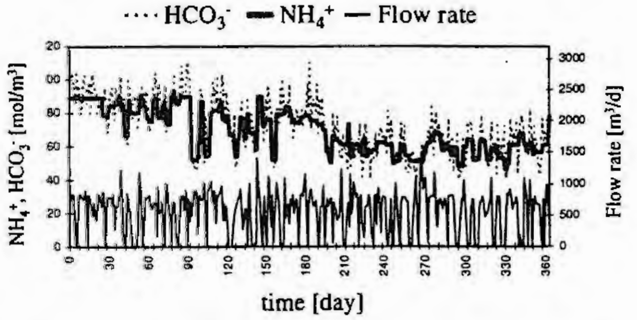


Figure 3. Day average influent compositions and flow rates (1994) Used for the predictions in table 1

The 10 components of vector $\underline{r}$ are: r_{CO2HO}, r_{NO2H}, r_{O2}, r_{CH3OH}, r_{NO3}, r_{N2}, r_{NH3H}, r_{Xden}, r_{Xnitr}, r_{Xamm}. The latter three express the conversion rates of denitrifying, and of nitrite and ammonium oxidizing biomass respectively. Most of the kinetic constants and the stoichiometric values composing matrix A were determined in our laboratory [4,7], and partly taken from literature [9]. A paper containing more details will soon appear.

Simulation results

Currently, the model was only used to investigate full scale process stability for actual influent data (fig. 3) with a simple control strategy. It should be noted that the bicarbonate concentrations were reconstructed from the year average values for concentration and standard deviation after discarding many spikes in the inaccurate measurements. The methanol supply rate was determined to maintain a sufficiently high pH level for getting 85% ammonium conversion on a yearly average basis. For practical reasons a reactor volume of 1500 m^3 will be used (a previous post thickening tank) which is currently slightly oversized. A cycle time of 4 hours was used, 2 hours for nitrification and 2 hours for denitrification. No special measures were taken to adjust the residence time with varying influent flow rates. Day average data for the influent over 1994 were used, with an average flow rate of 564 m^3/d. Methanol was supplied in fixed amounts (i.e. not related to the actual process load) in the beginning of the denitrification period. In experiments, hardly any performance loss of biomass was observed when the reactor was not fed for more than a week. Therefore, biomass degradation during periods without influent was not taken into account. Cooling is necessary, as under the mentioned conditions, microbial activity would have lead to a temperature increase of about 9°C. Relevant simulation results for a methanol supply of 1 kg/kg NH_4-N are shown in table 1 on a year average basis. Despite the large variations in influent flow rate and composition, irrelevant nitrate production is predicted. NH_4^+ conversion is predicted at over 86%. 71 % of the methanol was effectively used for denitrification.

On basis of these calculations, total costs were estimated at only $2-$3 per kg removed NH_4-N [3], being less than 50% of usual costs for ammonia removal. Variable costs for aeration and methanol dosage account for 70% of the total costs.

Summary

Full scale process design and operating conditions were established for the new SHARON process by combining laboratory scale experiments for microbial conversion with model predictions. In the kinetic expressions, the biochemical active form of components that are subjected to acid-base equilibria were included, so that with a minimum amount of experiments the pH dependency of the reactions could be modeled for this non-stationary process. In the model the equilibrium reactions were solved in an iterative procedure (assuming they are fast with respect to the microbial conversions) to calculate the pH and the concentration levels of these components, leading to very acceptable calculation times. The model was used so far to investigate process stability and conversion efficiency using a simple time-based control for aeration and methanol addition. With this simple, -yet sub-optimal- strategy, a methanol addition of 1 kg per kg NH_4-N in the influent (based on a year average influent load) was determined to reach an acceptable year average NH_4^+ conversion of 86 %. The model will be used for operator training and establishing a start-up procedure. Once data have become available from the full scale

process (1998), potential savings on aeration and methanol supply will be evaluated. If necessary, the model will be used to develop more advanced control strategies.

Acknowledgement. The SHARON project could only be successful thanks to very intense collaboration with the water board Zuid-Hollandse Eilanden en Waarden (ZHEW) and the Grontmij consultants. These organizations and the Dutch organization for water research (STOWA) sponsored the project. The authors especially appreciated the indispensable contributions of Andy Schellen (ZHEW), Lex van Dijk and Jan-Willem Mulder (Grontmij). Marc Brouwer and Serge Lochtman carried out the practical and theoretical work with great craftsmanship.

References

1. Anthonisen, A.C., Loehr, R.C., Prakasam, T.B.S., Srinath, E.G. (1976) Inhibition of nitrification by ammonia and nitrous acid, J. WCPF 48 (5): 835-852.

2. Böhnke, B. (1978) Möglichkeiten der Abwasserreinigung durch das "Adsorption-Belebungsverfahren". Verfahrenssystematik Versuchsergebnisse. GWA 29. Schriftenreihe des Instituts für Siedlungswasserwirtschaft der RWTH Aachen.

3. Brouwer, M., van Loosdrecht, M.C.M., Heijnen, J.J. (1996) STOWA report 96-01, Behandeling van stikstofrijke retourstromen op rioolwaterzuiveringsinstallaties, enkelvoudig reactorsysteem voor de verwijdering van nitriet.

4. Brouwer, M. Biologische stikstofverwijdering op Sluisjesdijk met het SHARON proces. BODL report, TU-Delft, April 1995.

5. Hunik, J.H. (1993) Engineering aspects of nitrification with immobilised cells. PhD thesis, Wageningen Agricultural University.

6. Kollbach, J.St., Grömping, M. Stickstoffrückbelastung: Stand der Technik 1996/1997; Zukünftige Entwicklungen. TK-verlag Karl Thomé-Kozmiensky, 1996.

7. Lochtman, S.F.W. Proceskeuze en -optimalisatie van het SHARON proces voor slibverwerkingsbedrijf Sluisjesdijk. BODL report, TUDelft, December 1995.

8. Riet, K. van 't, Tramper, H. (1991) Basic bioreactor design. Marcel Dekker Inc. New York.

9. Wiesmann, U. (1994) Biological nitrogen removal from wastewater. In Advances in Biochem. Eng./Biotechn. vol. 51. ed. by A. Fiechter. Springer-Verlag Berlin, New-York, pp 113-154.

MODELLING OF AN INTERMITTENTLY AERATED BIOREACTOR

Ján Derco, Alexander Kovács, Salima Shansab

Faculty of Chemical Technology STU, Department of Environmental Science,
Radlinského 9, 812 37 Bratislava, Slovak Republic

Abstract. An intermittently aerated completely mixing lab-scale reactor was used for the modelling of simultaneous nitrification and denitrification processes. The feed pulse change of organic component of synthetic wastewater was applied in order to simulate transient behaviour of the reactor. The COD, nitrate and biomass concentration responses of bioreactor were measured and evaluated. Different mathematical models for the description of the dynamic behaviour of simultaneous nitrification and denitrification processes carried out in this bioreactor have been developed and verified.

Introduction

The most common way of the removal of organic, nitrogen and phosphorus impurities from waste water are biological processes. There are usually lower capital and operational costs in comparison to physical and/or chemical processes of waste water treatment.

There are two environments in which biological processes of organic and nitrogen impurities removal are carried out.

In an aerobic environment dissolved oxygen is present in sufficient quantities as to not be rate-limiting. The concentration of dissolved oxygen is maintained at about 2 mg.l^{-1} in such a bioreactor. The content of organic pollutants in waste water can be minimised by heterotrophs. Similarly, ammonium nitrogen can be transformed into nitrite or nitrate. Inorganic carbon, i.e. carbon dioxide is utilised by nitrification bacteria.

In anoxic conditions dissolved oxygen is maintained at near zero levels. Thus the reduction of nitrite and nitrate to nitrogen oxide or nitrogen occurs. An organic carbon source is necessary for this process of denitrification. In addition to the above mentioned, the creation of anaerobic environment is inevitable in order to perform also enhanced biochemical phosphorous removal.

High removal efficiencies of nitrogen impurities removal are usually obtained by maintaining aerobic and anoxic zones in long ditch channel. On the other hand, it is difficult to form these different environmental conditions in a short ditch channel. The intermittent aeration method is available in these cases [1].

Intermittently Aerated Bioreactor

In this type of activated sludge process aerobic and anoxic periods alternate with each other by means of operating an aerator intermittently. Biological processes of nitrification and oxidation of organic substrates occur in the aerated period and denitrification occurs in the non-aerated period.

Enhanced phosphorous removal was observed in the fill-and-draw type activated sludge unit with the release of phosphorous during the non-aeration periods followed by the excess phosphorous uptake in the aeration periods [9].

Tne sludge solids reduction rate and the transformation of nitrogen and phosphorous compounds during the batch aerobic digestion of waste activated sludge carried out in the intermittent aeration were substantially equivalent to those in the continuous aeration, so the power cost for aeration can be saved by using the intermittent aeration method [8].

Experimental Modelling

Experimental modelling of biological processes of nutrients removal was carried out in an intermittently aerated completely mixing lab-scale reactor. The volume of the reactor was 4.2 dm^3. The synthetic waste water contained about 100 mg.l^{-1} of ammonium and 500 mg.l^{-1} of nitrates. Cyclohexanone was used as a carbon source for denitrification. The substrate concentration of COD was about 1200 mg.l^{-1}.

The cycle time was based on the hydraulic regime in the real plant and corresponded to 45 min. Technological parameters were similar to those of maintained in Carrousel bioreactor operated at Chemko, Strázske. The main reason was that the simulation of simultaneous nitrification and denitrification processes in the real Carrousel bioreactor in different lab-scale bioreactors, including completely mixing intermittently aerated reactor, was one of the principal objectives during this period of our study. The mean activated sludge horizontal velocity of 0.116 m.s^{-1} in the Carrousel channel for one aerator in function was taken into consideration.

The aeration period was altered from 10 to 30 min. in order to determine the optimal oxygen environment for organic and nitrogen impurities removal in this bioreactor. The reactor was aerated using air diffusers. The time of aeration was controlled by timer. Laboratory measurements were carried out at a temperature of 20 °C.

The analysis of COD, MLSS, ammonium and nitrate concentration were performed according to standard methods [5]. The dissolved oxygen concentrations were measured by Syland Dissolved Oxygen Meter. pH values were measured by Radelkis Aquacheck.

Mathematical modelling

Three mathematical models for the description of simultaneous nitrification and denitrification processes have been developed an verified in this work. All these models are based on the IAWPRC task group kinetic and stoichiometric concept [4].

Model 1 - the complete IAWPRC (International Association on Water Pollution Research and Control) kinetic and stoichiometric concept was adapted in the development of our first model of an intermittently aerated bioreactor (Model 1). Dissolved oxygen has been included as a substrate in the biochemical reaction rate terms in this model. Thus, the material balance of oxygen is the inevitable part of this model.

The following extension of this concept has been made in the development of this model. The evaporation process of organic volatile component of waste water was included in the model. The mathematical description of an organic compound evaporation is based on the material balance of this component of waste water. Similarly, the influence of organic component of waste water on oxygen transport rate, more precisely on the overall oxygen transport coefficient was considered in this model.

The dissolved oxygen was incorporated in this model by the inclusion of the material balance of this reactant. All the kinetics and stoichiometric parameters of the model were calculated with using of the optimisation procedure. Similar approach we applied also in our previous work [3].

Mathematical description of an organic compound evaporation is based on the material balance of this waste water component as follows:

$$q \cdot S_0 - q \cdot S - \frac{dS}{dt} \cdot V - K_1 \cdot S \cdot V - K_2 . S . p . V = 0 \tag{1}$$

where

Q - waste water flow rate [m.h^{-1}]

S_0 - organic component concentration at the beginning of the vaporising test measurements (COD) [kg.m^{-3}]

S - effluent organic component concentration (COD) [kg.m^{-3}]

K_1 - evaporation rate constant per unit volume due to mechanical agitation [h^{-1}]

K_2 - evaporation rate constant per unit volume due to aeration [h^{-1}]

V - tank volume [m^3]

Parameter p relates the time of aeration (t_a) and the cycle time (t_c), i.e.:

$$p = \frac{t_a}{t_c} \tag{2}$$

where

t_a - the time of aeration [h]

t_c - cycle time [h]

Oxygen transport rate can be described as follows:

$$V\frac{dC}{dt} = K_g . A . (C_s - C) \tag{3}$$

where

K_g - oxygen diffusion coefficient [$m.s^{-1}$]
A - area of gas diffusion [m^2]
C_s - saturation concentration of oxygen in solution [$g.m^3$]
C - concentration of oxygen in solution [$g.m^3$]

The integrated form of Eqn (3) is obtained by integrating between the limits of C_0 and C_t and 0 and t as follows ($C_0 = 0$):

$$C_t = C_s . \{1 - \exp(-K_L a . t)\} \tag{4}$$

where

$K_L a$ - overall oxygen transport coefficient per unit volume [h^{-1}].

Model 2 - our next approach to the modelling of simultaneous nitrification and denitrification processes is based on the measurements of the concentrations of COD, NH_4^+, NO_3^- and MLSS in steady-state before dynamic measurements. The formulas for the calculation of some model parameters were obtained based on the mass balances of selected components at steady-state conditions. The following material balances were used in order to obtain the above mentioned formulas:

1. mass balance of readily biodegradable substrate in order to compute the value of hydrolysis rate coefficient k_h
2. mass balance of nitrate nitrogen in order to compute the value of autotrophic yield coefficient Y_A
3. mass balance of autotrophic biomass in order to compute the value of autotrophic decay coefficient b_A
4. mass balance of heterotrophic biomass in order to compute the value of heterotrophic decay coefficient b_H
5. mass balance of particulate biodegradable organic nitrogen in order to calculate of the value of nitrogen fraction in particulate products coefficient i_{xp}

The values of the above given parameters, obtained by the evaluation of steady state measurements, were applied in the evaluation of dynamic measurements. It is obvious that this model represents one of the possible way how to reduce the model parameters number, which have to be determined experimentally or evaluated by the optimisation procedure.

Model 3 (simplified model) - this model is based on the inclusion of the aerobic and anoxic environments (i.e. the approximation of aerobic and anoxic portion of the bioreactor volume) by the 'switching functions' s_1, s_2, s_3 in the form as follows:

$$s_1 = \frac{1}{n_m} \cdot \sum_{i=1}^{n_m} \frac{O_i}{K_{OH} + O_i} \tag{5}$$

$$s_2 = \frac{1}{n_m} \cdot \sum_{i=1}^{n_m} \frac{K_{OH}}{K_{OH} + O_i} \tag{6}$$

$$s_3 = \frac{1}{n_m} \sum_{i=1}^{n_m} \frac{O_i}{K_{OA} + O_i} \tag{7}$$

where

O_i - dissolved oxygen concentration in bioreactor [$g.m^{-3}$]
K_{OA} - oxygen saturation coefficient for autotrophs [$g.m^{-3}$]
K_{OH} - oxygen saturation coefficient for heterotrophs [$g.m^{-3}$]
n_m - number of measured points of oxygen profile

This approach practically represents the approximation of aerobic and anoxic volumes of the biorecator. The values of the 'switching functions' were obtained by the evaluation of oxygen profile data measured in the bioreactor. These functions were incorporated into material balances of individual process components though the modified expressions of the biochemical reaction rates. Thus the material balance of oxygen is not necessary in this case. On the other hand, the influence of the organic component of the synthetic waste water on oxygen transport rate is included in this model implicitly through the values of these functions.

Solution and computing methods

The explicit fourth-order Runge-Kutta-Merson method [7] was employed for the set of differential equation solutions. The values of the steady-state before shock loading were taken as the starting values for solving. Values of model parameters were determined by the least square method, using the modified relaxation [10] and Nelder-Mead [2] algorithm for finding a minimum of the objective function.

Results and discussion

The measurements of the evaporation due to the volatility of the organic component (cyclohexanone) of the synthetic waste water were carried out both for mixing and aeration operation period of completely mixing reactor operated intermittently. The parameter values $K_1 = 0.0116\ h^{-1}$ and $K_2 = 0.0401\ h^{-1}$ of Eqn (1) were obtained by the evaluation of experimental data.

Significant influence of cyclohexanone concentration on oxygen transport rate resulted from the experiments. The following relation between the value of overall oxygen transport coefficient K_La and the value of cyclohexanone concentration (expressed as COD) was obtained by evaluation of experimental data:

$$K_La = 96.60 - 63.73\ e^{(-0.0121\ COD)}$$

where

COD - chemical oxygen demand [g.m^{-3}]

Oxygen transport coefficient values for various cyclohexanone concentrations were obtained by applying Eqn. (4).

The values of oxygen saturation concentrations at different cyclohexanone concentrations measured in water both, without and in the presence of activated sludge, are shown in Fig. 1. From Fig. 1 it can be seen the decrease of oxygen saturation concentration values at cyclohexanone concentration lower than about 100 mg.l^{-1}. In addition to this, only small changes in oxygen saturation concentration values at higher concentrations of cyclohexanone are evident from Fig. 1. It can be concluded that the presence of sludge also influences rates of oxygen transport.

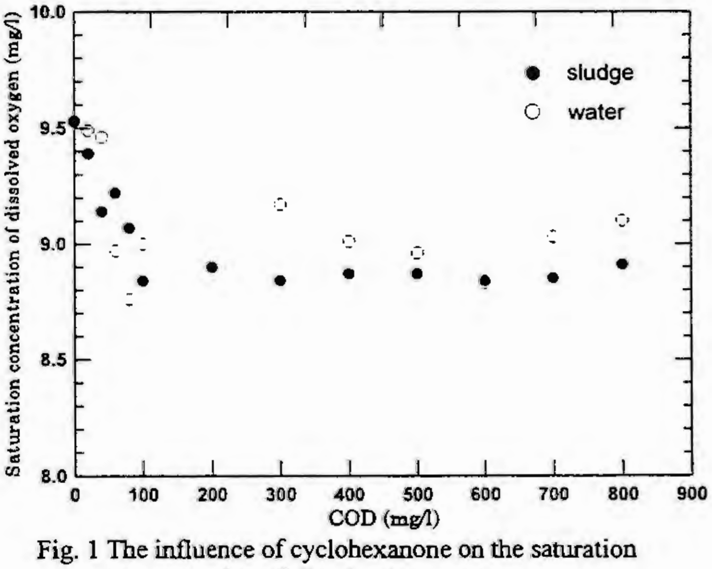


Fig. 1 The influence of cyclohexanone on the saturation concentration of dissolved oxygen

The COD, NO_3^- and MLSS responses of completely mixing intermittently aerated bioreactor on the feed pulse change of cyklohexanone concentration in the bioreactor were measured and evaluated. Good fits between experimental and calculated COD and NO_3^- values were obtained by applying the above described models (Fig. 2 and 3). The best agreement between experimental an calculated values of COD and nitrate were obtained by applying the Model 3, i.e. by the evaluation of ´switching functions´ values based on oxygen concentration measurements in the bioreactor. Dynamic changes of the switching functions (simplified model - Model 3) during the experiment are shown in Fig. 5. On the other hand, from Fig. 4 is evident higher increase of experimental values of biomass concentration during experiment in comparison to calculated values. The typical biokinetic parameter values for domestic waste water [6] were used as starting values for optimisation procedure.

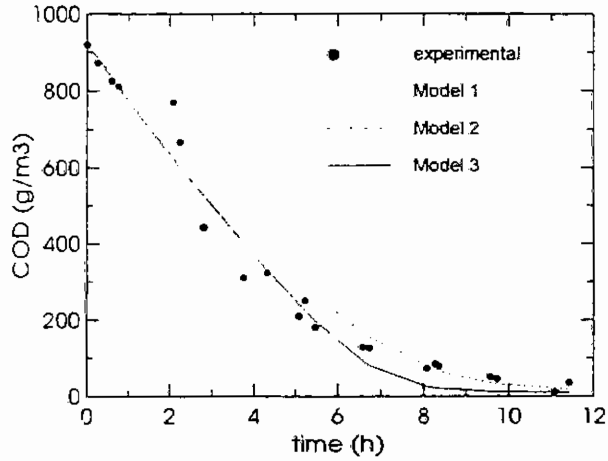


Fig. 2 COD responses to shock loading

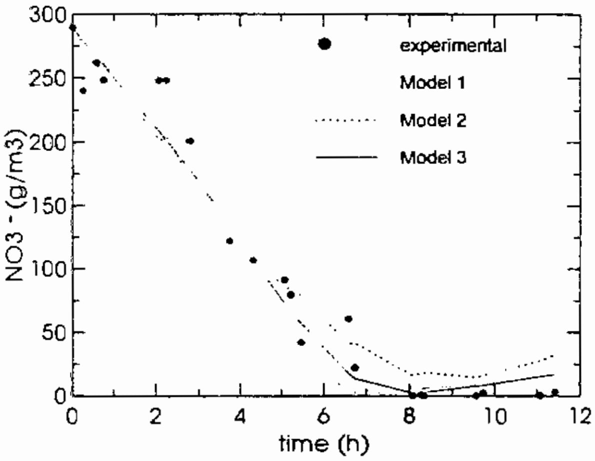


Fig. 3 NO_3^- responses to shock loading

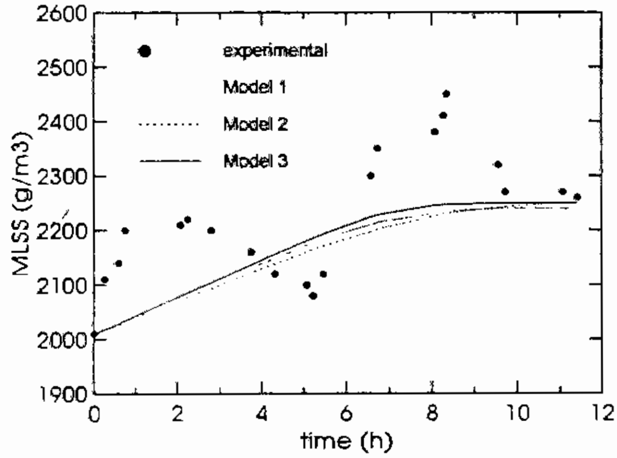


Fig. 4 MLSS responses to shock loading

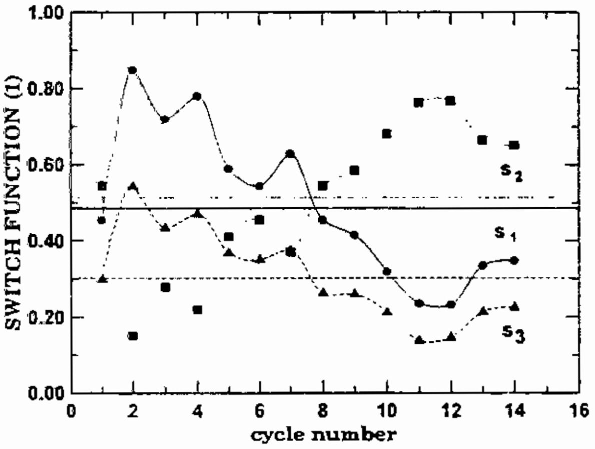


Fig. 5 Values of the switch functions

Conclusions

We have found very convenient laboratory equipment of completely mixing reactor with intermittent aeration for experimental modelling of simultaneous nitrification and denitrification processes carried out in high recycled systems. This is due to the simplicity of the equipment and the operational simplicity in comparison to tank-in-series bioreactor. Moreover the continuous oxygen profile is maintained in this reactor.

Acceptable agreement between experimental and calculated values of COD, NH_4-N, NO_3-N and MLSS was obtained by applying the mathematical model of completely mixing hydrodynamic in combination with modified and simplified IAWPRC kinetic and stoichiometric concept. The best description of experimental responses of the system to pulse feed change of organic component of wastewater was achieved by applying of 'switching functions' approach, which implicitly includes also the influence of organic component concentration on oxygen transport in the bioreactor. On the other hand, the utilisation of steady state measurements for the evaluation of some biokinetic parameter values reduces the parameter number to be estimated experimentally or by optimisation procedure. This is advantageous particularly in the case of industrial waste water treatment, where is usually a lack of kinetic and stoichiometric parameter values in the literature available for computer simulation using commercial simulation programs. Thus we have assessed that this model approach can be applied also for the description of the dynamic behaviour of a real system with similar hydraulic regime.

The simplified model of completely mixing intermittently aerated reactor based on the switching functions approach is advantageous particularly for highly recirculated bioreactor, i.e. Carrousel system of an industrial WWTP. Not only waste water composition but also temperature influence on oxygen transport rate can be included in this model trough the switching function values obtained by the evaluation of dissolved oxygen profiles monitored directly in the bioreactor.

All parameter values obtained in this work either by experimental or by optimisation methods were in the range of those published for domestic waste water. On the other hand, the combination of experimental methods and optimisation procedure for model parameters determination is convenient in order to allow the simulation of these processes.

References

1. Araki, H., Koga, K., Inomae, K., Kusuta, T., Away, Y., Intermittent Aeration for Nitrogen Removal in Small Oxidation Ditches. Wat. Sci. Tech., 34, (1990), 131-138.
2. Bounday, D. B., Basic Optimization Theory. Edward Arnold Publ., London, 1984.
3. Derco, J., Králik, M., Kovács, A., Berešíková, Z., □arnovský, L., Modelling of a Carrousel Plant by an Intermittently Aerated Activated Sludge Process. Pol. Jour. Env. Stud., 3, (1994), 25-30.
4 Grady, C. P. L., Gujer, W., Henze, M., Marais, G. v. R., Matsuo, T., A Model for Single - Sludge Wastewater Treatment Systems. Wat. Sci. Tech., 18, (1986), 47-91.
5. Greenberg, A. E., Clesceri, L. S., Eaton, A., Standard Methods for the Examination Water and Wastewater. American Public Health Association, Washington, DC 20005, 18th Edition, 1992.
6. Henze, M., Grady, C. P. L., Gujer, W., Marais, G. v. R., Matsuo, T., A General Model for Single - Sludge Wastewater Treatment System. Wat. Res., 21, (1987), 505-515.
7. Kubíček, M., Numerical Algorithms of Chemical Engineering Problems Solution. Publishers of Technical Literature, Prague 1983 (Czech).
8. Matsua, A., Ide, T., Fujii, S., Behaviour of Nitrogen and Phosphorus During Batch Aerobic Digestion of Waste Activated Sludge-Continuous Aeration and Intermittent Aeration by Control of DO. Wat. Res., 12, (1988), 1495-1502.
9. Osada, T., Haga, K., Harada, Y., Removal of Nitrogen and Phosphorous from Swine Wastewater by the Activated Sludge Units with the Intermittent Aeration Process. Wat. Res., 25, (1991), 1377-1388.
10. Pierre, D. A., Optimization Theory with Applications. J. Willey and Sons. New York, 1969.

A REDUCED ORDER MODEL FOR CONTROL OF A SINGLE REACTOR ACTIVATED SLUDGE PROCESS

S. Julien[1], **P. Lessard**[2], **J.P. Babary**[1]
[1] Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS/CNRS)
7, avenue du Colonel Roche, F-31077 Toulouse Cedex, France
[2] Département Génie Civil, Pavillon Pouliot, Université Laval
Québec, Canada, G1K 7P4

Abstract. The well-known IAWQ models are often used to simulate the behaviour of activated sludge processes. The IAWQ model n°1 slightly modified by adding a nitrogen limiting growth function for heterotrophic microorganisms, has been validated on a sequencing single sludge wastewater reactor doing nitrification and denitrification. However, this model is too complex to be used for control design. The main purpose of this paper is to show how it is possible to simplify the full model from biological and mathematical considerations. Properties of observability and controllability have been analyzed. Simulation results are compared with experimental data.

Introduction

The advent of more severe effluent criteria for wastewater treatment plants, specially regarding nitrogen removal, increases the need for more adequate control. The use of mathematical models thus becomes necessary to help in the elaboration of control strategies as well as for real time control and optimization of process performance.

Many models have been proposed to simulate the behaviour of the activated sludge process, culminating in the publication of the now widely referenced IAWQ model n°1 [2]. This mechanistic model, modified simply by adding a nitrogen limiting growth function for the heterotrophs, has been validated from experimental data. Sets of data were obtained from a lab scale pilot plant simulating **a single reactor activated sludge doing nitrification/denitrification through intermittent aeration** [7]. However, the use of such a model for real time control seems impossible due to its complexity and to the lack of measurable state variables. Very few researches have been done on the use of simple phenomenological models (called here reduced order models) for controlling activated sludge process [12], [4], leaving place for good and innovative research in that area.

A study was thus undertaken to elaborate a reduced order model to facilitate the use of control algorithms in real time control for an activated sludge process. The methodology developed hereafter could be summarized as follows:

- reduction of the *reference model* (I.A.W.Q. model *n°*1)
- adaptation of the model to make sure that the properties of ***observability*** and ***controllability*** are fulfilled
- calibration of the *reduced order model* from experimental data.

This paper presents the methodology used to simplify the model as well as a comparison between the results obtained with the reduced order model and the reference model.

Description of the process

The studied activated sludge process (low organic load) comprises a mixer, an aeration tank and a settler (Figure 1). Biomass is recycled from the settler to the aeration tank, excess sludge is wasted from the latter and oxygen is supplied intermittently in the reactor to create nitrification and denitrification conditions. Carbon and nitrogen are removed easily through this process which however is not optimized in terms of operation.

Two control variables can be considered: (a) the addition of an external carbon source to optimize the denitrification process, and (b) the aeration flowrate to optimize the cycle lengths (nitrification/denitrification) to meet the required effluent criteria and minimize eventually the energy consumption. On-line measurements available for these controls are oxygen and nitrate concentrations. Off-line measurements of ammonia concentration are also available and necessary for off-line identification of model parameters.

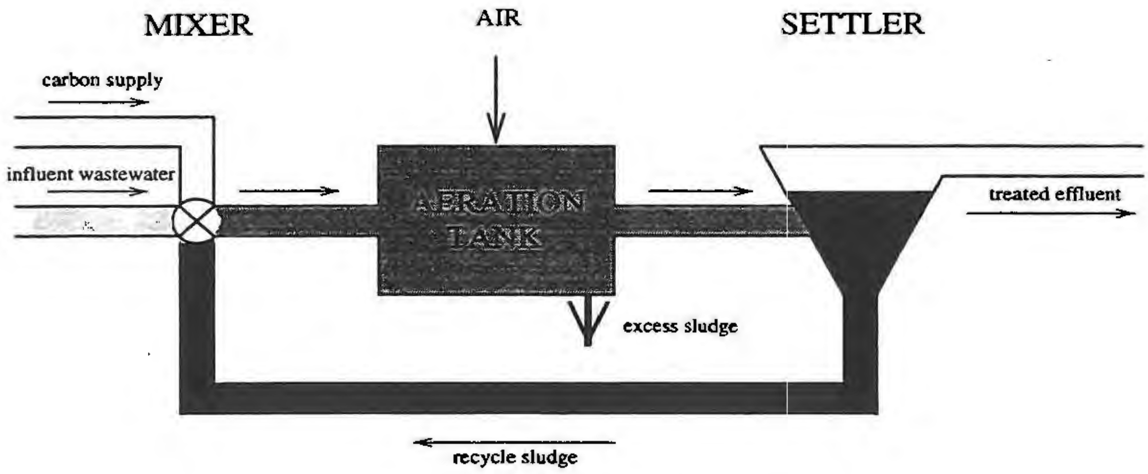


Figure 1: Activated sludge process

Mathematical modelling

The reference model which describes the elimination of nitrogen and carbon is the I.A.W.Q. model n°1 [2] in which some modifications have been made :

- ▷ the alkalinity which does not appear in kinetic reactions of other components has been omitted,
- ▷ the particulate products arising from biomass decay have been introduced in the definition of the particulate inert organic matter (like in the I.A.W.Q. model n°2 [3]),
- ▷ a new parameter K_{NH4H} has been introduced to take into account the ammonia limitation for the aerobic and anoxic growth of heterotrophic biomass.

The mathematical model takes into account the aerobic growth of heterotrophic and autotrophic biomass, the anoxic growth of heterotrophic biomass, the decay of both types of hiomass, the ammonification of soluble organic nitrogen, the hydrolysis of entrapped organics and organic nitrogen.

The reference model contains eleven state variables and twenty parameters, and has been written down under the following assumptions :

- the aerator and the mixer are supposed to be perfectly mixed,
- the biological reactions take place only in the aeration tank,
- the settler is considered perfect : the treated effluent does not contain particulate products and concentrations of soluble reactants are the same at inlet and outlet of the settler.

After parameter identification, simulation results show a good fitting with experimental data [6, 7] obtained from four series of experiments.

The considered reactor being a low organic loading activated sludge process, the removing of carbon does not state a problem ; so, it may be interesting to get a simplified state model characterized by three state variables S_{NO3}, S_{NH4} and S_{O2} defining concentrations of nitrate, ammonia and oxygen respectively.

Reduced order model

In order to get a reduced order model, a certain numher of approximations has to be done from biochemical and mathematical considerations. This can be achieved by observing the evolution of state variables in different experimental conditions and by simplifying the reaction kinetics ; but the obtained reduced order model must fulfil ohservahility and controllahility properties.

The following biochemical approximations have been made :

1. some state variahles (e.g. the soluble and particulate inert organic matters) do not intervene in state equations corresponding to the rest of variahles : they are removed,

2. some functions of biomass concentrations in the hydrolysis kinetics can be considered as constants,
3. some state variables (e.g. the heterotrophic and autotrophic biomasses) do not vary much within one nitrification/denitrification cycle ; they are replaced by their mean values,
4. some input concentrations (e.g. the influent nitrate and oxygen concentrations) are very small : they are considered as zero,
5. the sludge containing a low organic loading (small value of carbon concentration), the Monod-type kinetics related to the carbon are approximated by a linear function,
6. the Monod-type kinetics related to ammonia for heterotrophic growth can be approximated by a constant,
7. knowing that the concentrations of nitrate, ammonia and oxygen are the only state variables of the reduced order model, the concentration of the readily biodegradable substrate is expressed as a fraction of its amount in the influent wastewater, of the supplied external carbon and of the hydrolysis of organics.

The reduced order model can be splitted into two submodels : one submodel in aerobic conditions (submodel n°1) and one submodel in anoxic conditions (submodel n°2). The switching from the aerobic model to the anoxic model occurs when the oxygen concentration is near zero ; the switching from the anoxic model to the aerobic model occurs when the air is supplied. This model must fulfil the observability and controllability properties. Nonlinear methodology approaches [9] have been applied for each submodel.

- *Observability* - In the first case, it has been shown that the submodel n°1 is observable ; in the second case, the submodel n°2 is observable only if the reduction assumption n°6 is omitted : therefore, the ammonia limitation of the heterotrophic growth has been considered again.

- *Controllability* - A controllability study has been made in aerobic conditions by considering two types of control (the supplied external carbon only or combined with the air flowrate) : the submodel n°1 is controllable in both cases. In anoxic conditions, by considering the supplied external carbon only, it is possible to control the nitrate concentration only (the ammonia concentration is imposed by the ammonia concentration at the inlet of the reactor).

The activated sludge process operates in continuous mode with sequencing phases according to aerobic ($S_{O2} > 0$) or anoxic ($S_{O2} = 0$) conditions ; it is then described by two differential equation systems :

- aerobic conditions

$$\begin{aligned}
\dot{S}_{NO3} &= -(D + D_c) * S_{NO3} - \tfrac{1}{2.86} * k_{22} * (S_{Sin} + \tfrac{D_c}{D}.S_{Sc} + \tfrac{k_5}{D}) * \tfrac{K_{O2H}}{S_{O2}+K_{O2H}} \\
&\quad * \tfrac{S_{NO3}}{S_{NO3}+K_{NO3}} + k_3 * \tfrac{S_{O2}}{S_{O2}+K_{O2AUT}} * \tfrac{S_{NH4}}{S_{NH4}+K_{NH4AUT}} \\
\dot{S}_{NH4} &= D * S_{NH4in} - (D + D_c) * S_{NH4} - b * (S_{Sin} + \tfrac{D_c}{D}.S_{Sc} + \tfrac{k_5}{D}) \\
&\quad *(k_{12} * \tfrac{S_{O2}}{S_{O2}+K_{O2H}} + k_{22} * \tfrac{K_{O2H}}{S_{O2}+K_{O2H}} * \tfrac{S_{NO3}}{S_{NO3}+K_{NO3}}) \\
&\quad -k_3 * \tfrac{S_{O2}}{S_{O2}+K_{O2AUT}} * \tfrac{S_{NH4}}{S_{NH4}+K_{NH4AUT}} + k_4 \\
\dot{S}_{O2} &= -(D + D_c) * S_{O2} + kla * (S_{O2sat} - S_{O2}) - k_{12} * \tfrac{S_{O2}}{S_{O2}+K_{O2H}} \\
&\quad *(S_{Sin} + \tfrac{D_c}{D}.S_{Sc} + \tfrac{k_5}{D}) - 4.57 * k_3 * \tfrac{S_{O2}}{S_{O2}+K_{O2AUT}} * \tfrac{S_{NH4}}{S_{NH4}+K_{NH4AUT}}
\end{aligned} \tag{1}$$

- anoxic conditions

$$\begin{aligned}
\dot{S}_{NO3} &= -(D+D_c)*S_{NO3}-\frac{1}{2.86}*k_{22}*\frac{S_{NO3}}{S_{NO3}+K_{NO3}}*\frac{S_{NH4}}{S_{NH4}+K_{NH4H}} \\
&\quad *(S_{Sin}+\frac{D_c}{D}.S_{Sc}+\frac{k_5}{D}*\eta_{NO3h}*\frac{S_{NO3}}{S_{NO3}+K_{NO3}}) \\
\dot{S}_{NH4} &= D*S_{NH4in}-(D+D_c)*S_{NH4}-b*k_{22}*\frac{S_{NO3}}{S_{NO3}+K_{NO3}}*\frac{S_{NH4}}{S_{NH4}+K_{NH4H}} \qquad (2) \\
&\quad *(S_{Sin}+\frac{D_c}{D}.S_{Sc}+\frac{k_5}{D}*\eta_{NO3h}*\frac{S_{NO3}}{S_{NO3}+K_{NO3}})+k_4 \\
\dot{S}_{O2} &= 0
\end{aligned}$$

where

- $D=\frac{Q_S}{V}$ and $D_c=\frac{Q_c}{V}$ with Q_S the input flowrate, Q_c the external carbon flowrate and V the volume of the reactor,
- S_{Sc} is the concentration of the supplied external carbon,
- kla is the coefficient of oxygen transfer,
- S_{O2sat} is the saturation concentration of oxygen,
- S_{NH4in} and S_{Sin} are the concentrations of ammonia and soluble carbon in the input wastewater,
- $(k_{12}, k_{22}, k_3, k_4, k_5, K_{O2H}, K_{O2AUT}, K_{NH4AUT}, K_{NO3}, b)$ and $(k_{22}, k_4, k_5, K_{NO3}, K_{NH4H}, \eta_{NO3h})$ are the model parameters respectively in aerobic and anoxic conditions.

Experimental data from a pilot unit being rather limited, only the most influent parameters should be estimated. One sensitivity study of parameters in aerobic and anoxic conditions has therefore been realized ; the results are as follows :

- in aerobic conditions, parameters b, k_{12}, k_{22}, k_3, k_4 and k_5 must be identified ; other parameters have the numerical values obtained by identification of the I.A.W.Q. model n° 1 [7],
- in anoxic conditions, only parameters k_{22}, k_4, k_5 and η_{NO3h} are influent on the solution of the model.

Consequently, seven parameters are to be identified in the reduced order model ; these parameters can be estimated only if the model is identifiable. An identifiability study is then necessary.

Structural identifiability

There exist few available methods concerning the structural identifiability for nonlinear systems. The model structure of the bioreactor being highly nonlinear and on-line measurements of ammonia being impossible, the methods used have been the generating series approach [11] and the similarity transformation approach [1, 10, 11].

To apply the similarity transformation approach, the model is assumed to be observable and controllable, which has been previously verified only for the submodel n°1. The application of this approach has shown that all parameters of this submodel are identifiable [5].

In anoxic conditions, the submodel n°2 being observable but not controllable, it has been shown, by using the generating series approach, that only parameters k_{22}, k_4, K_{NO3}, K_{NH4H} and $k_5*\eta_{NO3h}$ are identifiable. Indeed, parameter k_5 being identifiable in the submodel n°1, parameter η_{NO3H} can also be determined.

Globally speaking, all parameters of the reduced order model are identifiable.

Parameter identification and simulation results

The parameter identification has been implemented by using the simplex method of Nelder and Mead [8], because it does not need the computation of the sensitivity functions. The performance index to be minimized is a quadratic function of errors between experimental data (concentrations of nitrate, ammonia and oxygen) and the solution of the reduced order model. Experiments were performed in different conditions of flowrates (influent wastewater and air) and influent substrates concentrations. A parameter identification has been realized for each experiment. It has been stated that four parameters $(k_{12}, k_5, \eta_{NO3H}, b)$ have practically the same value from one experiment to another one, which is not the case for the other parameters (k_3, k_4, k_{22}).

Figure 2 represents the evolution of concentrations of nitrate, ammonia and oxygen during six hours. The influent flowrate has been increased by 76% at time t = 3 hours. This figure allows a comparison between the solution of the reduced order model with the solution of the reference model and with experimental data. The discrepancies are relatively small at any time.

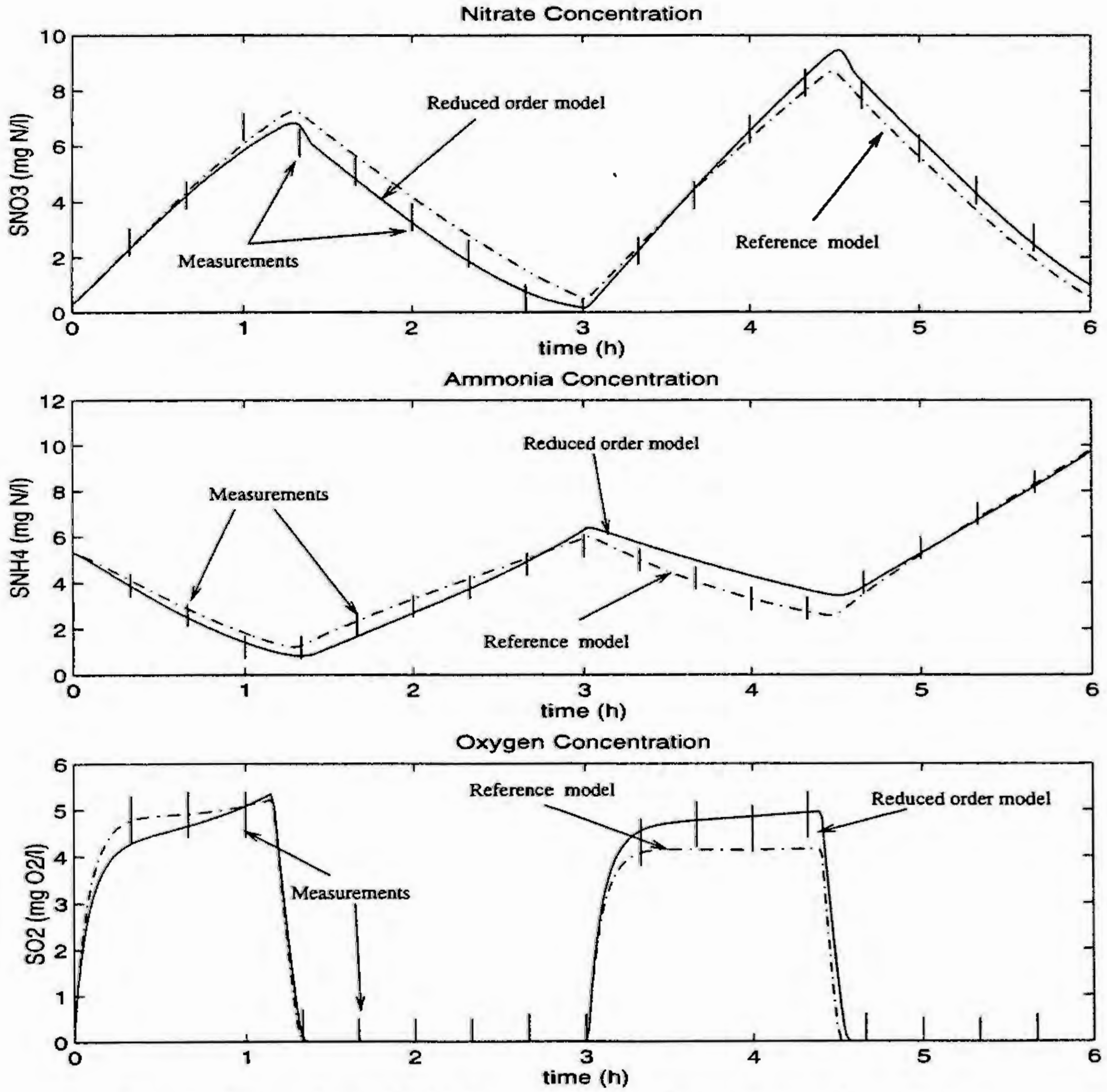


Figure 2: Observed and simulated concentrations of nitrate, ammonia and oxygen (experiments of 19.12.1995)

Conclusion

It has been shown in this paper that the complex mathematical I.A.W.Q. model can be simplified by considering biological and mathematical assumptions in the case of a low organic loading activated sludge process. The simplification leads to a reduced order model useful for designing control algorithms.

An identifiability study, prior to the practical parameter identification, allowed to determine which parameters are identifiable for each phase (aerobic or anoxic). Nevertheless, by considering both phases together, it has been possible to estimate all parameters of the reduced order model.

Simulation runs allow to compare solution of the general model, solution of the reduced order model and experimental data. A good fit between experimental and simulated values has been observed.

References

[1] Chappell, M.J. and Godfrey, K.R., Global Identifiability of the Parameters of Nonlinear Systems with Specified Inputs : a Comparison of Methods. Mathematical Biosciences, 102 (1990), 41 – 73.

[2] Henze, M., Leslie Grady Jr, C.P., Gujer, W., Marais, G.v.R., and Matsuo, T., A General Model for Single-Sludge Wastewater Treatment System. Wat. Res., 21(5) (1987), 505 – 515.

[3] Henze, M., Gujer, W., Mino, T., Matsuo, T., Wentzel, M.C. and Marais, G.v.R., The Activated Sludge Model n°2 Biological Phosphorus Removal. In : IAWQ Specialised seminar : Modelling and Control of Activated Sludge Processes, 1994.

[4] Jeppsson, U., On the Verifiability of the Activated Sludge System Dynamics. PhD thesis, Dept. of Ind. Elec. Eng. & Automation, Lund Institute of Technology, Sweden, 1993.

[5] Julien, S., Lessard, P., and Babary, J.P., Theoretical and Practical Identifiability of a Reduced Order Model in an Activated Sludge Process doing Nitrification and Denitrification. In 7th IAWQ Workshop on Instrumentation, Control & Automation of Water & Wastewater Treatment & Transport Systems, Brighton (UK), 1997 (submitted).

[6] Julien, S., Lessard, P., Babary, J.P., and Paul, E., Modelling and Simulation of a Single Reactor Activated Sludge Process. In Automatics and Informatics'96, Sofia (BULGARIA), 1996 : 200 – 203.

[7] Julien, S., Lessard, P., Babary, J.P., Plisson-Sauné, S., and Paul, E., Simulation d'un Procédé par Boues Activées à Faible Charge : Validation du modèle de l' I.A.W.Q. n°1. In 18ème Symposium International sur le Traitement des Eaux Usées, Montréal (CANADA), 1995, 139 – 154.

[8] Nelder, J.A. and Mead, R., A Simplex Method for Function Minimization. Computer Journal, 7 (1965), 308 – 313.

[9] Nijmeijer, H. and Van Der Schaft, A.J., Nonlinear Dynamical Control Systems. Springer-Verlag, New York, 1990.

[10] Vajda, S., Godfrey, K.R., and Rabitz, H., Similarity Transformation Approach to Identifiability Analysis of Nonlinear Compartmental Models. Mathematical Biosciences, 93 (1989), 217 – 248.

[11] Walter, E. and Pronzato, L., On the Identifiability and Distinguishability of Nonlinear Parametric Models. In Proc. Symp. Applications of modelling and control in agriculture and bioindustries, IMACS, Bruxelles (BELGIUM), 1995, V.A.3-1 – V.A.3-8.

[12] Zhao, H., Isaacs, S.H., Soeberg, H., and Kümmel, M., An Analysis of Nitrogen Removal and Control Strategies in an Alternating Activated Sludge Process. Wat. Res., 29(2) (1995), 535 – 544.

A FUZZY SELF-TUNING ALGORITHM FOR POWER SYSTEM MANAGEMENT IN NEXT GENERATION VEHICLES

Xavier J.R. Avula[1] **and Mukund Sridhar**[2]
[1]Department of Mechanical and Aerospace Engineering and Engineering Mechanics
University of Missouri-Rolla, Rolla, Missouri 65409-0050, U.S.A.
[2]Detroit, Michigan, U.S.A.

Abstract. Due to increasing concern with environmental pollution created by the utilization of petroleum derivatives in the operation of conventional vehicles, electric and hybrid vehicles are becoming prominent among the next generation vehicles. Development of electric vehicles is plagued by limited driving range and inconsistent battery behavior. Some of the difficulties encountered in the electric vehicle technology can be better understood using performance models. However, these performance models involve a high degree of uncertainty. As fuzzy logic has been proved to be a valuable tool in modeling systems with high degree of uncertainty, this approach has been adopted for the study of power system management of next generation vehicles described here. In this work, using a gradient descent technique, fuzzy linguistic states pertinent to next generation vehicle power systems parameters have been described and then tuned to the system behavior. A self-tuning algorithm based on error back propagation has been generalized to tune a class of fuzzy models with a high degree of accuracy. The application of the algorithm has been extended to electric vehicle load forecasting due to varying charging pattern and traffic behavior.

Introduction

Awareness of the environmental deterioration caused by automobile air pollution and petroleum fuel consumption has stimulated an intensified search for alternative power sources to the automobile internal combustion engine (ICE). An energy efficient, pollution free electric power source is a logical candidate due to several reasons. An electric vehicle produces only one tenth of the pollution emitted by ICE-powered car, even after figuring in the emissions from the power plants used for recharging the cars. This is in addition to freedom from the carcinogenic agents in the air. There is no noisy exhaust, valve train clatter, or knocking in electric vehicles as compared to their conventional counterparts. With the increasing consumption of gasoline and the instability in the gasoline prices, electric vehicles can reduce a country's trade deficit as well as avert an economical setback resulting from an oil crisis or an embargo. A fall in demand for petroleum products would also reduce the oceanic oil spills and the horrendous amount of crude and refined fuel that leaks from storage tanks into ground and pollutes the soil and the subterranean water resources.

The limited range and often inadequate power for rapid acceleration offered by the state-of-the-art, rechargeable batter powered electric vehicle (EV) severely restricts its usefulness. The main drawback of electric vehicles is the limited battery capacity and the high discharge rates which result in a limited driving range. The recharge time of the currently available batteries are significantly high as compared to the instant refueling capability of the gasoline engines. This causes a severe drawback for continuous driving. Due to the absence of any infrastructure for manufacturing, the initial costs are higher than their IC engine counterparts. These may considerably fall once the electric vehicles capture a significant market.

In order to circumvent these shortcomings, many modeling-assisted studies involving candidate technology comparisons have been performed to provide information for planning and research decisions. Modeling has also been extensively used in engineering activities including preliminary and final design optimization. The reliability of the results of computer simulation depends upon the validity and accuracy of its component models. Design experience has made it relatively easy to predict the performance of some of the major vehicle and drive component models.

Fuzzy Logic Approach to Performance Studies

Fuzzy logic, with its rule based approach has been effectively implemented in both modeling and practical applications [1]. It has particularly been useful in cases where an "expert" rather than a mathematical model has been effective in predicting/controlling a system behavior.

The introduction of a fleet of electric vehicles in the market will result in an increase in electric power demand. Hence it is imperative for the power industries to form an effective means to predict the power demand due to electric vehicles. The charge requirement at any time of the day depends on a number of factors like charging pattern, number of vehicles being charged, state of charge requirement and the usage pattern. In this work, fuzzy logic is used to forecast the demand taking various factors into account. Modeling the complex behavior of batteries presents an intriguing challenge. The second application deals with the use of fuzzy logic in modeling the battery characteristics related to voltage monitoring as well as predicting energy densities in the electric vehicle batteries.

Generalized Self-Tuning Algorithm (GSTA)

The underlying principles of fuzzy logic were first introduced in Professor Zadeh's first paper on fuzzy set theory in 1965[2]. Since then there have been major advances in the theory, applications, and implementations of fuzzy logic systems.

The theory of fuzzy sets allows a type of uncertainty due to vagueness or fuzziness rather than due to randomness alone. It its most basic sense, a fuzzy set is a set where objects have gradual rather than abrupt transition from membership to non-membership. The use of quantitative analysis (traditional models, e.g., a set of differential equations or statistical analysis) has been growing as a part of an effort to provide a more rational basis for many engineering problems. It is very difficult, if not impossible, to analyze engineering experience or human ability to draw conclusions which are based on analogy, using conventional statistical analysis. The fuzzy set theory is based on the premise that the key elements in human thinking are not numbers but words. The most important feature of human thinking is the inability to form a standard technique to extract from a collection of masses of data only such items of knowledge which are relevant to the task at hand.

Perception in the real world usually consists of summary representations of reality and is therefore imprecise and subjective. Hence these interpretations use concepts which do not have sharply defined boundaries, such as large, fast, very wide. In fuzzy logic, such concepts are represented as fuzzy sets, i.e., classes of objects in which transition from membership to non-membership is gradual rather than abrupt.

The numerical parameters and linguistic states that are to be chose for a fuzzy model become a challenge. One may have to go through endless iterations to hit upon the optimum choice of the numerical parameters governing the problem. Several algorithms have been proposed to tune the member functions. One of the most effective methods as suggested by Nomura et al. [3], deals with tuning of the fuzzy antecedent part and real consequent part based on the gradient descent technique. This technique has been implemented for an obstacle avoidance problem for the control of steering angle of a mobile robot using distance and angle with respect to the obstacle as its inputs. This algorithm suffers from a lack of generalized approach in dealing with fuzzy consequents. Hence the gradient descent technique has been extended to the problems involving fuzzy consequents. The mathematical derivation using the error backpropagation technique is shown in the following sections and implemented for electric/hybrid vehicle performance models. The first stage in the algorithm is the parametric representation of a fuzzy model.

Fuzzy Parametrization. Consider a m-parameter input and a r-parameter output case to study the parametric representations of the fuzzy rules. The inputs are represented by $x_1, x_2, x_1 ... x_m$ and the output by y.
The inference rule of simplified fuzzy reasoning can be represented as follows.
Rule i
IF x_i is A_{il} and x_2 is A_{i2} and ....and x_m is A_{im} THEN y_1 is OA_{il} and...y_r is OA_{ir} where is the rule number, A_{il}, A_{i2}, ...A_{im} are the fuzzy sets of the *antecedent* part and OA_{il}, OA_{i2}...OA_{ir} are the fuzzy sets of the *consequent* part. Since we consider triangular membership functions, the antecedent part is represented by an isosceles triangle. The two parameters representing the membership function are the center mp_{ij} and the width bs_{ij}. The output of the fuzzy reasoning can be derived by the following set of equations.

$$\mu_{ij}(x_j) = 1 - \frac{2\left[x_{j-mp_{ij}}\right]}{bs_{ij}} \tag{1}$$

$$\mu_{ci} = \mu_{il}(x_1) \cdot \mu_{i2}(x_2) \ldots \mu_{im}(x_m) \tag{2}$$

$$\Delta_{ir} = 0.5obs_{ir}\mu_{ci}(2 - \mu_{ci}) \tag{3}$$

$$y_r = \sum_{i=1}^{n} \frac{\Delta_{ir} omp_{ir}}{\sum_{i=1}^{n} \Delta_{ir}} \tag{4}$$

i = 1 to n (number of rules)
j = 1 to m (number of inputs)
k = 1 to K (number of data sets)
r = 1 to R (number of outputs)
p = 1 to P (number of parameters)

mp_{ij} and bs_{ij} are the midpoint and base width corresponding to the ith rule and the jth input, while omp_{ir} and obs_{ir} correspond to the midpoint and base width corresponding to the ith rule and the rth output. μ_{ij} represents the membership function corresponding to the jth input and ith rule. μ_{ci} is the result of the operation on the fuzzy rules using the multiplication operator. Δ_{ir} represents the area corresponding to a consequent output OA_{ir} with a weightage corresponding to Δ_{ir}, y corresponds to the rth fuzzy output.

The four parameters defining the fuzzy rule are mp_{ip}, bs_{ip}, omp_{ip} and obs_{ij}. These parameters corresponding to all the rules are to be tuned to a desired pattern or behavior. The fuzzy algorithm for the tuning consists of two basic stages:

1) An initial setting of the reference rules is conducted first and mp_{ij}, bs_{ij}, omp_{ij}, and obs_{ij} are set so as to allow the overlap over the whole domain. Consequent parameters omp_{ij} and obs_{ij} are tuned by substituting the output of the fuzzy reasoning y, the membership value, and the output data y^k.
2) Antecedent parameters mp_{ij} and the bs_{ij} are then tuned by substituting the changed parameters of the consequent part omp_{ij} and obs_{ij} and performing backpropagation based on the cumulative least squared error (CLS error) estimate of the output of the fuzzy reasoning y and the output data y^k.

Tuning by Gradient Descent Method. The objective of the gradient descent method is to seek for the vector Z which decreases an objective function $E(Z)$, where Z is a p-dimensional vector $Z = (Z_1, Z_2, Z_3, ..., Z_p)$ of the tuning parameters. In this method, the vector which decreases the value of objective function is represented by

$$\left(-\frac{\delta E}{\delta z_1}, -\frac{\delta E}{\delta z_2}, \ldots, -\frac{\delta E}{\delta z_p}\right) \tag{5}$$

and the learning rule is expressed by the following formula

$$z_1(t+1) = z_i(t) - K \cdot \frac{\delta E(Z)}{\delta z_1} \, (i = 1, \ldots p) \tag{6}$$

where t is the number of iteration of learning and K is a constant. Altering Z according to the learning rule, the value of the objective function converges to a local minimum. In the present method, the inference rules are tuned so as to minimize the objective function E which is defined by the following.

$$E = \sum_{i=1}^{n} \sum_{i=1}^{n} \left(y_r^k - y_{dr}^k\right)^2 \tag{7}$$

where y_r is the desired output data obtained from the expert. The objective function represents the inference error between the desirable output y_{dr}^k and the output of fuzzy reasoning y_r^k.

Since the shape of the membership function governing the set A_{ij} is defined by the center value mp_{ij}, the objective function E consists of the tuning parameters mp_{ij}, bs_{ij}, omp_{ij}, and obs_{ij} $(I=1,2,...n; j=1,2,...m; r=1,2,...R)$. Therefore, the present method can be an application of the descent method by which the optimum vector Z to minimize the objective function $E(Z)$ can be derived when the vector Z is defined as follows.

$$(z_1, z_2, \ldots, z_p) = (mp_{11}, \ldots mp_{nm};\ bs_{11}, \ldots bs_{nm};\ omp_{11}, \ldots, omp_{nR};\ obs_{11}, \ldots obs_{nR}) \tag{8}$$

where subscript $P=2n(m+R)$. R is the number of consequents, m is the number of antecedents and n represents the number of rules. Equations denoting the parameter updating show the respective $(t+1)th$ values of the tuning parameters. K_a, K_b, and K_c are constants. The comprehensive update equations when substituted by the error gradients are given in the Appendix.

The learning algorithm adaptively changes the tuning parameters for a direction to minimize the objective function E. Thus using the learning rules, the tuning parameters of the inference rules are optimized to minimize the inference error between the desirable output y_{dr}^k and the output of the fuzzy reasoning y_r^k.

Power Load Forecasting Due to Charging of Electric Vehicles

Formation of Fuzzy Rules. Since the objective in our case is to obtain the power demand forecasting, consequent part of the fuzzy system becomes the power demand or the electrical load. The antecedent part consists of the time of the day as we wish to obtain the daily distribution. The fuzzy rules are formed based on the utilizing capacity of the customers we have addressed. The linguistic states for the antecedent part are formed by the assumed behavior that the peak demand occurs primarily during two time periods. The first time period is mid-day period between 10 AM and 4 PM when the parked vehicles of the office goers get charged. The second time period is the night time between 10 PM and 6 AM when the utilization is very low and the charging rate is high causing the peak demand to occur. The traffic could be high or low during the rest of the period depending upon the nature of the day (i.e. whether it is a working day or a holiday etc.) causing uncertainty in the traffic behavior.

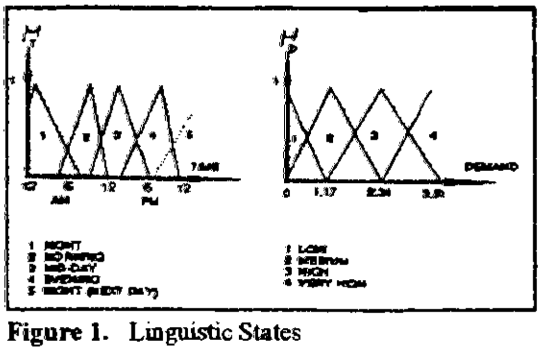


Figure 1. Linguistic States

The nature of a day in a week is divided into four categories: working day, holiday, working day before a holiday or a weekend, and holiday before a holiday or weekend. The rules for the fuzzy model are framed based on the expert's opinion on the behavior of the power demand for a day belonging to a particular category. The set of seven categorized rules are as follows:

WORKING DAY *If* TIME *is* MORNING *then* DEMAND *is* MEDIUM

If TIME *is* MID-DAY *then* DEMAND *is* HIGH

If TIME *is* EVENING *then* DEMAND *is* LOW

WORKING DAY (excluding the day before a holiday)

If TIME *is* NIGHT *then* DEMAND *is* VERY HIGH

DAY BEFORE A HOLIDAY

If TIME *is* NIGHT *then* DEMAND *is* LOW

HOLIDAY

If TIME *is* MORNING *or* MID-DAY *or* EVENING *then* DEMAND *is* LOW

HOLIDAY (excluding holiday before a holiday)

If TIME *is* NIGHT *then* DEMAND *is* VERY HIGH

The rules are framed on the basis of assumptions listed above. The sixth rule is a combination of the three rules and they are taken so during computation. The data set used for training was taken from the results obtained in [4] and the criteria for the formation of the fuzzy rules were based on the proposed power generation behavior in [5]. The defuzzification is carried out by training the model using the data set for a normal week with five working days.

Discussion of Results

Generalized Fuzzy Self-Tuning Algorithm. Different systems with considerable degree of non-linearity were tuned to a good error estimate using the tuning algorithm. The CLS error estimate for another data set was compared using the tuned rule base and also the expected system behavior (Figures 2, 3). The parameters K_a, K_b, $K_{\alpha a}$, $K_{\alpha b}$ were chosen so as to achieve the best convergence for the given data set and error estimate. The Centroid method was used for refuzzification and multiplication operator was used for the AND operations of the rules. The interpretation of these results shows that the initialization of the parameters in the back propagation network may be closer to the desired parameter set in a fuzzy network than in neural network. Initialization in the fuzzy algorithm is not done randomly as in the conventional neural network, but is governed by the rule base.

Figures 2 and 3 give the three dimensional representation of the fuzzy and the desired system. Figure 2 depicts that, if the initial rule base contains some erroneous rules (intentionally provided to the system shown), the untuned system may provide a different behavior than what is intended. Also, if the rule base contains a lot of inaccurate rules, GSTA does not converge.

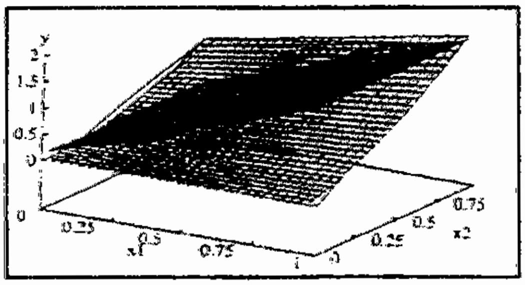


Figure 2. Untuned Fuzzy Model vs. Desired System ($y=x_1 \cdot x_2+x_2^2$)

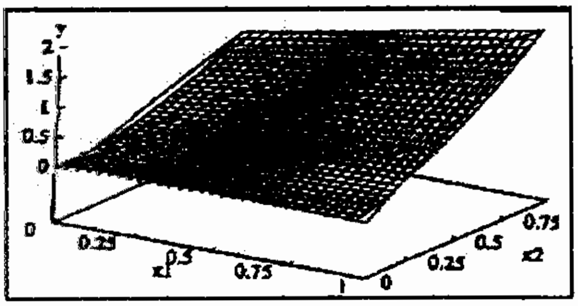


Figure 3. Tuned Fuzzy Model vs. Desired System ($y=x_1 \cdot x_2+x_2^2$)

Power Load Forecasting. The power load demand for a normal week with five working days and two holidays were used as a data set for tuning the rules. The data contained power load samples for every two hours. The tuning was performed till a cumulative least squared error of less than 0.003 was attained. Figures 4 and 5 show the tuning behavior and the error behavior respectively. The effect of the rule base was tried using a different input set which involved a week patterns of a Tuesday holiday and a long weekend. The results indicate relatively lower charging rate on Monday nights in Figure 6 and Thursday, Friday nights in Figure 7. The flattened peaks indicate the paucity of the data which is unavailable due to apparent lack of electric vehicles in today's market. The forecasting model may be updated every 3 months or even online. Online updating is preferred till the time the electric vehicles stabilize in the market.

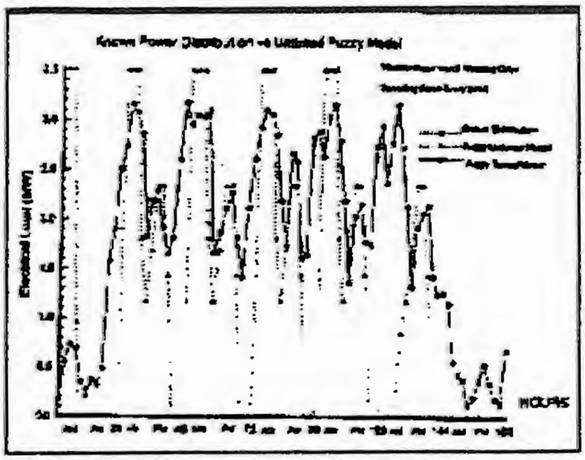


Figure 4. Tuning Behavior for the System

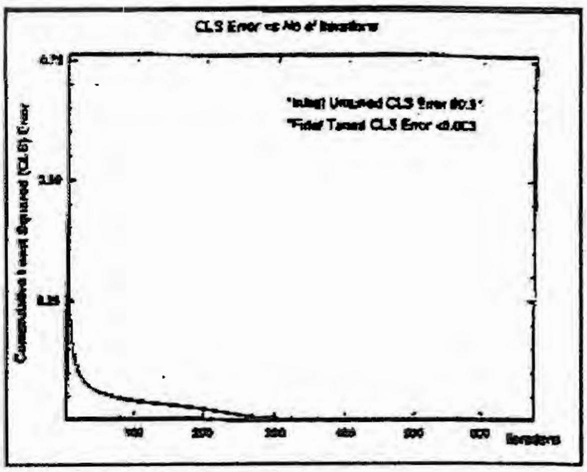


Figure 5. CLS Error Estimates

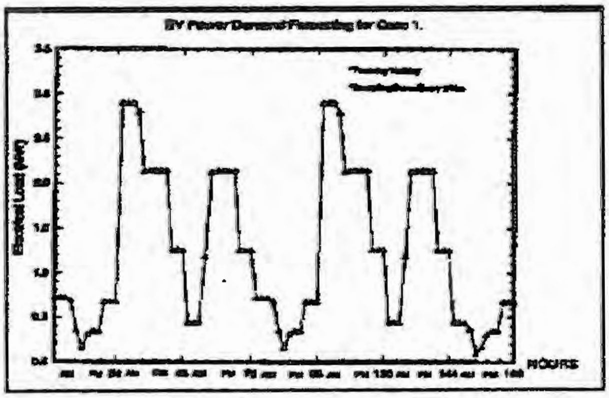


Figure 6. Power Load Forecasting for Case I

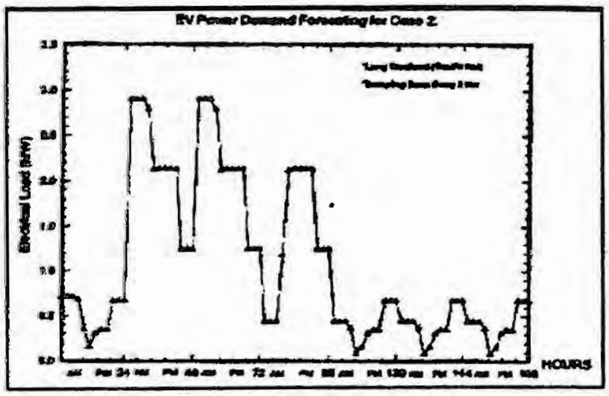


Figure 7. Power Load Forecasting for Case II

Conclusion

Fuzzy logic has been proved to be a good tool as a decision maker in problems with high degree of uncertainty. The gradient descent method used in the generalized self tuning algorithm has been effective in tuning the parameters governing the linguistic states. Power load forecasting due to electrical vehicle battery charging has been performed using the self tuning algorithm. The closeness of results with the test data indicate that a fuzzy rule base can be made to learn a preset pattern or behavior. This can provide an effective technique in data analysis and forecasting problems.

References

1. Isik C., "Fuzzy logic: Principles, applications and perspectives," SAE 1991 transactions, Journal of Aerospace, Paper No. 911148.

2. Zadeh, L.A., "Fuzzy sets," Information Control, 8, pp. 338-353, 1965.

3. Nomura, Hiroyoshi, Isao Hayashi, Noboru Wakami, "A learning method of fuzzy inference rules by gradient descent method," IEEE International Conference on Fuzzy Systems, pp. 203-210, 1992.

4. Rahman, S. And G.B. Shreshta, "An investigation into the impact of electrical vehicle load on the electric utility system," IEEE Transactions on Power Delivery, Vol. 8, No. 2, pp. 591-597, April 1993.

5. Collins, Michael M., "Impact of electric vehicles on electric power generation," EVC No. 8146, EVC Symposium VI, Baltimore, Maryland, October 21-23.

MODELS OF MIGRATION OF ORGANIC POLLUTANTS IN SOILS

Andrzej URBANIAK[1], **Michał WIKA**[1] **and Adam VOELKEL**[2]
[1]*Institute of Computing Science,*
Poznań University of Technology
ul. Piotrowo 3A, 60-965 POZNAŃ, Phone: (+48) (61) 790 790, fax: (+48)(61) 771525
E-mail: urbaniak@put.poznan.pl
[2]*Institute of Chemical Engineering and Technology,* **Poznań University of Technology**

Abstract. The mathematical migration models of the chosen organic pollutants in soils are considered. The motivation of the research is the migration analysis of oil derivatives in the soil environment. It is very important problem connected with the ground water pollution. The mathematical models besides the soil properties description include also physicochemical data of pollutants and take into account the possibility of their reactions. The analysis of the soil models is provided for different levels of accuracy (three levels). First level uses only the general equations with several parameters such as:

- soil composition,
- water content and the character of water resource.

In the second level the ground configuration is taken into account. The expansion of these two models is the model in which many hydrological and geological parameters are introduced. The physicochemical characteristics of system components are also considered. We assumed the point sources of the pollutions in the models and several others restrictions for clarity and solvability of the problem. The proposed analysis is necessary for optimal location of the fuel stations due to their direct influence on the nearest surroundings. Thus the location of the fuel station ought to be decided using the different criteria. In our models the environmental protection criteria are most important ones. The selected parameters of the models are determined on the basis of empirical results. As a tool in the modelling process the specialized software is used. The parts of modelling procedures were prepared separately according to special features of modelled environment. The elaborated models were prepared for PC with standard configuration. In the authors opinion presented models are important for petrochemical industry and especially for oil distribution.

Introduction.

Nowadays, there are many different pollutions migration processes' models. It is caused by the different point of view in modeling process, kind of presumed principles or limitations. However all these differences are determined by a model expected application, and therefore by its expected detail level and preciseness. It is very important to notice that adding more limitations will increase inaccuracy of a model. It is also very important to check if made limitations did not completely change theoretical meaning of processes and mechanisms which are being modeled. Modeling is a very complicated process, that shows the Fig.1 [5].

Below there are described these differences and also likenesses of pollutions migration processes' models. The first kind of the models division is delimitation because of a quantity of dimensions taken into consideration in a model. It means an axis of a 3 dimensional space, but not a time. A time parameter is necessary in all models because of a dynamic character of modeled processes and mechanisms. And therefore there are one, two and three dimensional models. Obviously one dimensional model is a kind of the most simple model in this division. In spite of preciseness and credibility of these models, they have only limited applications. Because it is impossible to simulate real processes on these models with great exactness, they may be only applicated in estimating these processes. The results of estimating are not very exact. They belong to the most simple models. But they find an application from the scientific point of view. These models are basic in scientific researches concerning more complicated real processes. They are also fundamental in development of existing models in the purpose of searching and creating new and more actual models. For example in 1968 Lindstrom proposed a mathematical model describing leaching of pesticides through column of soil.

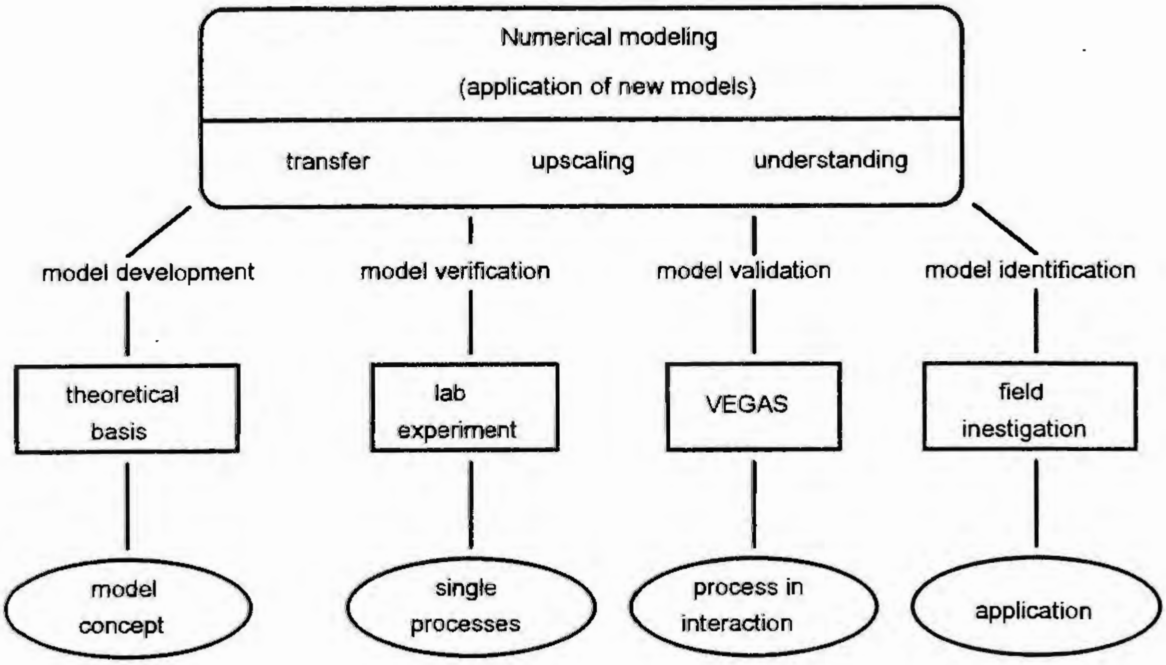


Fig. 1 A structure of modelling process

Other feature delimiting existing models is the way by which the model describes the type and the structure of the soil at which get the pollutions. It is very difficult and almost impossible to create the model, which on the one hand would be as much universal as could describe all kinds of the soil, but on the other hand would be as much exact as could be real. The most different limitation in modeling process is to presume that the soil is completely homogeneous. It means presuming that the migrating pollutions get at the same type and structure of the soil all the time. This is a very difficult limitation and this way it limits exactness of received results. The better and more exact limitation in this point of view is to presume layer structure of the soil. In this case we assume that all layers are homogeneous, but the ground consists of many different layers with different proprieties. But for heterogeneous layers modeling process is very difficult.

There are many other various types of limitations which have a different influence on modeling results. They concern an underground water flow type, adsorption and desorption processes or degradation. Some of more simply models do not take into consideration sorption processes [2]. Some models do not describe the degradation process [1,6]. All these limitations have strong influence on exactness of model, but they have also influence on its application. Another main criterion of models division is a kind of injected pollutions. It is very important what type of compounds pollutions belong to. It determines what kind of processes model should describe and how strong changes and reactions influence the soil or groundwater. As it has been shown there is many points which differentiate existing models. In the paragraph below there are described some of them to demonstrate differences mentioned above.

Models of pollutions migration processes.

First of all there will be described some of more simply models. Restrictions in application of this models are very serious. However the great limitation of a very complicated real situation makes it very often easier to understand the mechanisms of pollutions migration. The other very important thing is possibility of applying simplificated methods without access to the computer.

First example of a simple model is a piston flow model in a radial flux [3]. An injection in this model is short-lived. An application is only in estimating. The model presumes the following limitations: 1) an aquiferous layer is not limited, isotropic and homogeneous; 2) a motion of groundwater is constant; 3) pollutions do not undergo a sorption, and do not react in any other way with a soil medium; 4) a dispersion of pollutions in pore medium may be overlooked. The other simple model is one dimensional model with radial flux of groundwater with taking into consideration a longitudinal dispersion. This model inquire a continuous type of injection. Pollutions in this model also do not undergo the sorption processes. They also do not undergo a degradation. A fundamental difference is caused by inquiring dispersion process.

The less simple is the model with two dimensional hydrodynamic field for an extensive focus of the pollution injection [3]. In this model there exist only a few limitations. It presume that: 1) a motion of groundwater is constant; 2) pollutions do not undergo the sorption process and degradation; 3) but pollutions undergo transversal and oblong dispersion in aquiferous layer; 4) a total dispersion is proportional to the filtration velocity what means that hydrodynamic dispersion predominate in an absolute way. More exact models take into consideration real processes such as adsorption or desorption. In this model type a limitation concerning these processes is changed. It is presumed the analyzed pollutions undergo sorption process and ion exchange. Obviously there is a limitation in this case. The model presumes equilibrium between a solution and a soil stabilize at once. The model overlooks an influence of sorption processes and ion exchange kinetics. An adsorption follows linear isotherm. An ion exchange has an equivalent character. This model is not so simple as models describe before, and it is obvious that its preciseness is greater than others. As an example of a model which undergo sorption process Van Genuchten and coworkers (in 1974 - 1977) described a model for predicting pollutions migration in soil. The pollution substance was a pesticide. They also investigated the effect of a nonequilibrium adsorption.

The model described by Jury, Spencer and Farmer in 1983 includes the effects of the volatilization, leaching, and degradation of a soil-applied organic chemicals [4]. This model presumed follow limitations: 1) uniform soil properties; 2) a linear, equilibrium adsorption isotherm; 3) a linear, equilibrium liquid-vapor partition; 4) an uniform initial incorporation; 5) a loss of the pesticide and water to the atmosphere limited by a gaseous diffusion. This model is to describe the major loss pathway of soil-applied organic chemicals as a function of specific environmental variables and soil conditions. Three major loss pathways were examined: degradation, mobility and volatilization. Authors described the application of this model as an application not for predicting a chemical's concentration distribution but only for grouping chemicals according to their behavior in the environmental screening tests. Primarily this model has been used in pesticides injection, but it is applicable to other organic chemicals as well. Authors showed how the model might be used to identify the major loss pathways for a given chemical and to determine the relative mobility, volatility or persistence of a group of chemicals.

The next model is a VIP model. This microcomputer model named VIP (Vadose Zone Interactive Processes) was developed at Utah State University to provide a mathematical description of a land treatment system [1]. The model simulates vadose zone processes including the volatilization, degradation, adsorption / desorption, advection and dispersion. Four physical phases in the vadose zone are considered including water, oil, soil grains and soil-pore air (unsaturated pore space). The bad point in this model is dimensionallity. It limits the applications for it because of its limited exactness. This model will be detaily described in the next section. There will be also described a trial to extend this model to two or even three dimensions. This trial has on the purpose the application abilities increasing of VIP model described before in simulating real processes of the pollutions migration.

Vadose Zone Interactive Processes Model (VIP) and its extensions.

This model simulates vadose zone processes such as volatilization, degradation, adsorption and desorption, advection and dispersion. VIP model describes the fate of the hazardous organic substances in the soil. Because the model is one dimensional, it describe processes in a column of soil. In land treatment point of view this soil column is made up of two different layers: 1) the zone of incorporation, 2) the lower treatment zone. Applicated abbreviations of this zones are often named as: for first layer is ZOI, and for the other is LTZ. The ZOI (also called the plow zone) is the first top layer at which the pollution get during injection. Typically it is presumed that this first layer is about 15 cm deep. The second layer (LTZ) extends below zone of incorporation and typically is about 1.5 m deep. This zone contains substances which have been mobilized and transported from the first layer (ZOI). This model presumed that pollution substance has homogeneous chemical properties.

Pollution may be a pure compound or it also may be a mixture consists of a few different compounds as long as the behavior of the mixture can be adequately described by composite substance parameters [2].

The fate of the pollution in the soil depends on mobilization, volatilization and decomposition rate. The model VIP takes into consideration that injected pollution substance may be mobilized by three mechanisms: advection, dispersion and migration between or among phases. Mechanisms of mobilization by advection or dispersion may be significant for the air and water phases. The soil grain phase is not mobile. Mechanisms of mobilization by migration of pollutions between or among phases are modeled as the sorption processes. This model assumes that the sorption can occur directly between two of any phases being in contact. Modeling of pollutions migration between more than two phases at the same time would be extremely difficult and it would complicate the equations describing these processes in the model. This model assumes that pollution migration from one phase to another must pass through the water phase. However volatilization is represented in the model by two processes: mass flux into the air phase and advection/dispersion.

Another process described in the model is degradation process. This process describes the loss of the pollution substance from the ground without regard to mechanism. It may be caused by volatilization for volatile chemical substances. Because the pollution substance may degrade at different rates in different phases, separate coefficients describe each phase in the model. The processes described before are represented by six following equations:

$$1)\ \frac{\partial C_w}{\partial t} = -V_w \frac{\partial C_w}{\partial z} - \mu_w C_w - \frac{\kappa}{\theta_w}(\theta_a K_{aw} + \theta_o K_{ow} + \rho K_{sw})C_w + \frac{\kappa}{\theta_w}(\theta_a C_a + \theta_o C_o + \rho C_s)$$

$$2)\ \frac{\partial C_a}{\partial t} = -V_a \frac{\partial C_a}{\partial z} + D_a \frac{\partial^2 C_a}{\partial z^2} - \mu_a C_a + \kappa(K_{aw} C_w - C_a) - C_a \frac{\partial \theta_a}{\partial t}$$

$$3)\ \frac{\partial C_o}{\partial t} = -\mu_o C_o + \kappa(K_{ow} C_w - C_o) - C_o \frac{\partial \theta_o}{\partial t}$$

$$4)\ \frac{\partial C_s}{\partial t} = -\mu_s C_s + \kappa(K_{sw} C_w - C_s)$$

$$5)\ \frac{\partial \theta_o}{\partial t} = -\gamma_o \theta_o$$

$$6)\ \theta_o = \phi - \theta_w - \theta_o$$

where the subscripts w, a, o, s indicate the water, air, oil and soil grain phases respectively. Meanings of symbols are as follows:

C - concentration of the pollution substance in the phase described by subscript (g/m^3),
V - vertical pore velocity of the phase (m/day),
μ - first order decay rate for the pollution within the phase (1/day),
κ - a mass transfer rate coefficient (1/day),
θ - volume of the phase within the control volume (m^3 phase/m^3 control volume),
K_{aw}, K_{ow}, K_{sw} - linear partition coefficients,
ρ - soil bulk density,
D - dispersion coefficient for the phase (m^2/day),
ϕ - soil porosity,
γ_o - degradation rate of the oil phase,
z - depth, positive downwards (m),
t - time (days).

For extending this model to three dimensional model there are required some changes in equations. The first two equations will be changed. They will be look like follows:

1) $$\frac{\partial C_w}{\partial t} = -V_{wx}\frac{\partial C_w}{\partial x} - V_{wy}\frac{\partial C_w}{\partial y} - V_{wz}\frac{\partial C_w}{\partial z} - \mu_w C_w - \frac{\kappa}{\theta_w}(\theta_a K_{aw} + \theta_o K_{ow} + \rho K_{sw})C_w + \\ + \frac{\kappa}{\theta_w}(\theta_a C_a + \theta_o C_o + \rho C_s)$$

2) $$\frac{\partial C_a}{\partial t} = -V_{ax}\frac{\partial C_a}{\partial x} - V_{oy}\frac{\partial C_a}{\partial y} - V_{az}\frac{\partial C_a}{\partial z} + D_a(\frac{\partial^2 C_a}{\partial x^2} + \frac{\partial^2 C_a}{\partial y^2} + \frac{\partial^2 C_a}{\partial z^2}) - \mu_a C_a + \\ + \kappa(K_{aw} C_w - C_a) - C_a \frac{\partial \theta_a}{\partial t}$$

In this case the model not only include vertical pore velocity of phase, but must include three components of pore velocity adequately to three dimensions of model. This extension added in this model will increase the quantity of possible applications in studying and observing processes of pollutions migration. This model may be used in simulating real processes which occur in real soil environment.

A computer model of migration processes.

The equations of extended VIP model are programmed in PSI/c. PSI/c is an interactive, expression-oriented and block-structured simulation program for studying the behaviour of dynamic continuous and discrete systems. There are many different function types which may be used to define a model. There are dynamic continuous and discrete, linear and non-linear, logic and boolean functions. PSI/c has many different facilitise which improve the modeling process such as full screen editor with menu commands bar or optimization facility. The equations of VIP model are defined as follows:

1) $$C_{wi} = INT\left[\frac{A_1}{2\Delta x}(C_{wi-1} - C_{wi+1}) + \frac{A_2}{2\Delta y}(C_{wi-1} - C_{wi+1}) + \frac{A_3}{2\Delta z}(C_{wi-1} - C_{wi+1}) + BC_{wi} + C\ \right]$$

2) $$C_{ai} = INT\left[\frac{D_1}{2\Delta x}(C_{ai-1} - C_{ai+1}) + \frac{D_2}{2\Delta y}(C_{ai-1} - C_{ai+1}) + \frac{D_3}{2\Delta z}(C_{ai-1} - C_{ai+1}) + \\ + \frac{E_1}{\Delta x^2}(C_{ai-1} - 2C_{ai} + C_{ai+1}) + \frac{E_2}{\Delta y^2}(C_{ai-1} - 2C_{ai} + C_{oi+1}) + \frac{E_3}{\Delta z^2}(C_{ai-1} - 2C_{ai} + C_{ai+1}) + FC_a\ \right]$$

3) $C_o = INT(GC_o + H)$

4) $C_s = INT(IC_s + J)$

5) $\theta_o = INT(K\theta_o)$

6) $\theta_a = L$

where letters from A to L are constants which represent all coefficients which are included in equations of model.

Model verification and validation.

Verification of the numerical model is very often based on analytical solutions for problems without gravity and capillarity. In general however, analytical solutions that consider fully efects of gravity and capillarity in multiphase flow through porous media are not tractable. That is why the model which is verified against analytical solutions can still give wrong results. In such cases the only way to verify model's credibility is to verify by comparison of the model and an experimental results [2]. Therefore the extended VIP model will be verify by comparison between numerical simulations and experimental results.

The much more difficult and complicated problem of process interpretation (validation) is to prove that the numerical model provides a valid representation of physical reality of modeled processes. The object of validation is to examine whether the model is a good description of reality in terms of its behavior and its intended applications. It is possible to perform the validation only for specific limited conditions [7]. The difficulty in validation process of extended VIP model results from extending dimension of the model.

Conclusions

1. The mathematical model presented in the paper based on the existing VIP model was developed for more real application, especially for three dimensions.
2. The computer model was perform using PSI/c software and there was also realized numerical verification process.
3. The verification process is just making and its results will present at the conference lecture.

References.

1. Canter L.W., Knox R.C., Ground water pollution control;Lewis Publishers, Inc. Michigan 1986

2. Grenney W. J. , C. L. Caupp, R. C. Sims, A Mathematical Model for the Fate of Hazardous Substances in Soil: Model Description and Experimental Results, Hazardous Waste & Hazardous Materials, no. 3, 4 (1987).

3. Jordan H.P., A. S. Kleczkowski, J. Silar, W. M. Szestakow, S. Witczak, Ochrona wód podziemnych, Wydawnictwa Geologiczne Warszawa 1984.

4. Jury W. A., W. F. Spenser, W. J. Farmer, Behavior Assessment Model for Trace Organics in Soil, Journal of Environmental Quality, no. 4, 12 (1983).

5. Kobus H., B. Barczewski, H.-P. Koschitzky (Eds.), Groundwater and Subsurface Remediation, Springer-Verlag Berlin Heidelberg 1996.

6. Ne-Zheng Sun, Mathematical modelling of groundwater pollution, Springer, Los Angeles 1989

7. Sabljic A., Quantitative Modeling of Soil Sorption for Xenobiotic Chemicals, Environmental Health Prospectives, 83, (1989), 179-190

Modelling of Batch Reactors for Simulation, Optimization and Process Control

M. V. Le Lann*, M. Cabassud and G. Casamatta
Ecole Nationale Supérieure d'Ingénieurs de Génie Chimique, Laboratoire de Génie Chimique UMR CNRS 5503,
18, Chemin de la Loge 31078 TOULOUSE Cedex FRANCE
email : MarieV.LeLann@ensigct.f, Fax : +33 5 62 25 23 18, Phone : +33 5 62 25 23 68
*: author to whom correspondence should be addressed

Abstract

Simulation, optimization and control needs modelling. But, what kind of model for each objective ? Without the pretension to have the universal answer, we give the main results of our studies performed in these three domains in the particular case of the batch reactor.

Introduction

With the emergence of the generic drugs, pharmaceutical industries which were not used to be in a severe concurrence are now pushed to fight in order to be the first on the market. So, they are now interested in any technique or methodology which permits them to reduce the time to market. The heart of a drug manufacturing process is the batch reactor which is still widely used in fine and pharmaceutical industries. It is often characterised as a flexible and multipurpose equipment. That means that a same apparatus is used to carry out different reactions and operations under various operating conditions involving chaining of sequences.

Operation of a batch reactor covers a wide domain from optimization of operating conditions to on-line control and monitoring. In each of these domains, computational procedures are generally based on a process model. The main difference between models lies in their complexity. This is mainly due to the difference in the objective to be reached. There is not an unique answer to the question : "What is the best model ?". Keeping in mind that if the model has to be representative of the process, it is clear that its development has to be shorter as possible. In this paper, we want to focus on the different points to look at when one develops a model.

For instance, if simulation is concerned, a complete dynamic model of the batch reactor has to be elaborated. Such a model is necessarily based on mass and energy balances on the reaction mixture and the jacket and must contain an accurate description of the kinetic laws as well. Moreover, when used for control strategies studies, this model has to perfectly describe the behaviour of the penalising plant element, i.e. either the jacket or the heating-cooling system, which is commonly neglected in the literature. For example in the case of multifluid system, we have felt necessary to give a special care to model the flow hydrodynamics inside the jacket.

On the contrary, concerning the optimization of operating conditions, most often the objective function includes only chemical requirement as yield, selectivity, and the common trend is to focus only on the kinetic model. This generally allows to determine optimal operating conditions such as : initial concentrations, temperature and feeding reactive profiles, batch time., ... But these conditions often reveal themselves to be unrealistic and are not feasible on industrial plants. As it will be shown , by including a simple description of the heat exchanges and/or using constraints, it is possible to determine feasible solution applicable on a real industrial reactor.

In a third part, we want to focus on the use of models in control. A parallel can be made with the two previous domains. Generally, the temptation has been to use linear control tools as there were the most popular (in the industrial area) and as they offer guaranty of stability. Nevertheless, this solution which is probably satisfactory in the case of continuous processes, cannot fully answer the objective for discontinuous processes which don't exhibit steady-state and which requires the development of a controller as flexible as the batch reactor itself. Based on our experience in model-based control, several points will be emphasised. Our aim is to show, through experimental and industrial results how the structure of the model can be crucial to get good results.

Finally, as a conclusion, perspectives will be given on the approach we are developing in the field of experimental strategy and further how to capitalise information into an unique complete model, which would be simplified according to the specific application.

Modelling for simulation of batch reactor

Batch reactor is mainly used for production of fine chemicals or polymers. The common reactor is composed of the different parts : the reactor tank with its mixing device, the jacket, the condenser. Modelling of this unit involves the description of material evolution inside the reactor and the heat transfers between the different parts. Several assumptions have been adopted to establish the model equations (homogeneous temperature T_r and molar compositions x_r^i, chemical reactions occurring only in liquid phase, negligible vapor hold-up, thermodynamic equilibrium between the vapor phase and the reacting mixer if boiling, no reaction in the condenser, total condenser).

The mass and energy balances have been established on the different parts.

$$\frac{du_r}{dt} = cf + \Delta n\, u_r - cv + rx\, cl \tag{1}$$

with : - u_r : number of mols in the reactor (mol)

- cf, cv, cl : molar reactant feed , molar vapor stream reflux flow rates respectively (mol.s^{-1})
- rx : reflux ratio (rx = 1 : total reflux - rx = 0 : no reflux)
- Δn : production of total mols by the nr reactions (s^{-1})defined by the relations : $\Delta n = \sum_{i=1}^{nc} \sum_{j=1}^{nr} cs_{ij} Rr_j$ (2)

with : - cs_{ij} : stoichiometric coefficient of component i in the reaction j

- Rr_j : rate of the reaction j (s-^{1}) : $Rr_j = k0_j \exp(\frac{-E_j}{RT}) \prod_{i=1}^{nc} C_i^{\,or_{ij}}$ (3)

which can also be expressed as : $Rr_j = k0_j \exp(\frac{-E_j}{RT})(\frac{1}{Vm_r})^{(\sum_{i=1}^{nc} or_{ij})-1} \prod_{i=1}^{nc} x_r^{i\,or_{ij}}$ (4)

The material balance on component i can be written as :

$$\frac{d(x_r^i u_r)}{dt} = cf\, x_f^i + Rg_i\, u_r - cv\, y_v^i + rx\, cl\, x_c^i \qquad i = 1, nc \tag{5}$$

with : x_r^i , x_f^i, y_v^i, x_c^i: molar composition s of component i in the reacting mixture, reactant feed, vapor stream, the reflux stream respectively.

An energy balance on the reacting mixture yields :

$$\frac{d(h_r\, u_r)}{dt} = U_{rw}\, a_{rw}(T_w - T_r) + U_{rwb}\, a_{rwb}(T_{wb} - T_r) + U_{rs}\, a_{rs}(T_s - T_r) \tag{6}$$

$$+ cf\, mh_f(T_f, p, x_f^i) - cv\, mH_v(T_b, p, y_v^i) + rx\, cl\, mh_l(T_l, p, x_c^i) + q_r\, u_r$$

with : - h_r : molar liquid enthalpy of the reacting mixture (J.mol^{-1})

- U_{rw} ,U_{rwb}, U_{rs} : coefficients of heat transfer between the reacting mixture and the internal wall, the bottom internal wall, ambient surroundings respectively(W.m^{-2}.K^{-1})
- a_{rw}, a_{rwb}, a_{rs}: the corresponding exchange areas (m^2)
- T_r , T_w, T_{wb} , T_s: reacting mixture and external wall , external wall bottom, ambient surrounding temperatures (°C)
- mh_f , mh_l : molar liquid enthalpy of the reactant feeding stream and of the reflux (J.mol^{-1})
- mH_v : molar enthalpy of the vapor stream (J.mol^{-1})
- q_r : heat generated by the nr reactions (J.mol^{-1}.s^{-1}) defined by : $q_r = \sum_{i=1}^{nr} \Delta Hr_i\, Rr_i$ (7)

where ΔHr_i represents the heat generated by the reaction i (J.mol^{-1})

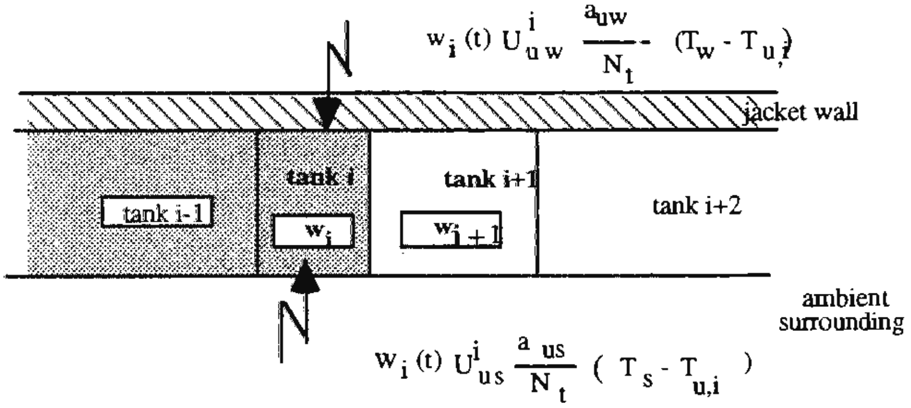
Generally, the evolution of the temperature in the jacket is modelled with a single heat balance by assuming that the temperature is homogeneous inside the jacket. The model obtained by this way does not take into account the hydrodynamic behaviour of the fluid in the jacket. In fact, the main question to be answered during the modelling step is: To what complexity extent has the model to be developed to correctly represent the behaviour of the process? . There is not a universal answer. It mainly depends on the objectives and for what purposes the model has been built? In the case of batch reactor, if the model is used to study scale-up problem, most often a simplify heat transfer balance on the jacket will be enough, but the information given by the model will not include the real dynamic as the jacket is supposed to be perfectly mixed. For safety studies, such a model will not give a complete information as the operation conditions are normally extreme and mainly dependent of the hydrodynamic behaviour (small flow rate which implies bad heat transfers). In the same way, when such a model is used to validate a control strategy, it has to include the most penalising dynamics of the process. In practice it can be noticed that the weak side of such a process is its heating-cooling system. Two main configurations are well known. The monofluid system consists in a unique fluid circulating in the jacket at a constant flowrate which the temperature is modified by external devices as heat exchangers, power heater, ... The multifluid system uses alternatively several utility fluids and the control of the heat transfer between the fluid and the reactor content is ensured by acting on the flowrate. So, it is clear that in the second case, hydrodynamics has the penalising dynamics which moreover depends on the flowrate changes. So, to perfectly describe the behaviour of such system, the solution that we adopted is a lumped of the jacket consisting of a succession of Nt perfectly mixed tanks. This number is related to the Peclet number for a given flowrate by the relation :

$$Nt = Pe/2 \tag{8}$$

This solution is very useful to describe the filling step during a changeover of utility fluids. These changeovers commonly occur in the case of the multifluid heating-cooling system during a classical batch reactor operation involving generally several steps as preheating, reaction phase, cooling. An energy balance on each tank i gives :

$$m_{u,i} Cp_{u,i} \frac{d T_{u,i}}{dt} = U^{i}_{us} (\frac{a_{us}}{Nt})(T_s - T_{u,i}) + U^{i}_{uw} (\frac{a_{uw}}{Nt})(T_w - T_{u,i}) + f_{u,i} \rho_{u,i} Cp_{u,i} (T_{u,i-1} - T_{u,i}) \tag{9}$$

In the case of a changeover of fluid, a tank can be filled with two different fluids. For this particular tank, we define a filling-up coefficient $w_i(t)$ which represents the fraction of the tank occupied by one of the utilities as shown on figure 1.



$w_i(t)\, U^{i}_{uw} \frac{a_{uw}}{N_t} (T_w - T_{u,i})$
jacket wall
tank i-1
tank i
w_i
tank i+1
w_{i+1}
tank i+2
ambient surrounding
$w_i(t)\, U^{i}_{us} \frac{a_{us}}{N_t} (T_s - T_{u,i})$

Figure 1 : Jacket representation during a changeover of fluid

A heat balance is established on each part of the tank filled with the two different utility fluids. For the first part:

$$w_i(t) \frac{V_j}{Nt} Cp_{u,i} \rho_{u,i} \frac{d T_{u,i}}{dt} = w_i(t) U^{i}_{us} (\frac{a_{us}}{Nt})(T_s - T_{u,i}) + w_i(t) U^{i}_{uw} (\frac{a_{uw}}{Nt})(T_w - T_{u,i}) + f_{u,i} \rho_{u,i} Cp_{u,i} (T_{u,i-1} - T_{u,i}) \tag{10}$$

A similar expression is obtained for the other part by replacing $w_i(t)$ by $w_{i+1}(t)$ and the physical properties, the heat transfer coefficients by those corresponding to the other utility fluid.

In the case of the monofluid heating-cooling system which doesn't involve changeover of fluid, only the equation (9) is used. The heat transfers across the jacket wall are thus modelled by :

$$\frac{d(m_w h_w)}{dt} = U_{rw} a_{rw} (T_r - T_w) + \sum_{i=1}^{Nt} U^{i}_{uw} (\frac{a_{uw}}{Nt})(T_i - T_w) \tag{11}$$

The temperature T_r of the reaction mixture is obtained by solving the equation : $h_r - mh_r(T_r, p, x^i_r) = 0$ by a call to a thermodynamic model or a thermodynamic data bank. In the case of boiling of the reaction mixture, the equation solved is $h_r - mh_r(T_b, p, x^i_r) = 0$ where T_b is the bubble point.

In this case a molar balance is used on the condenser : $$\frac{dc}{dt} = cv - cl \tag{12}$$

With cv obtained by:

$$cv = \{U_{rw} a_{rw} (T_w - T_r) + U_{rwb} a_{rwb} (T_{wb} - T_r) + U_{rs} a_{rs} (T_s - T_r) + cf\, mh_f(T_f, p, x^i_f) + q_r u_r - \frac{d(h_{rb} u_r)}{dt}\} / mH_v(T_b, p, y^i_v) \tag{13}$$

in the filling-up step of the condenser and otherwise by :

$$cv = \{U_{rw} a_{rw} (T_w - T_r) + U_{rwb} a_{rwb} (T_{wb} - T_r) + U_{rs} a_{rs} (T_s - T_r) + cf\, mh_f(T_f, p, x^i_f) + q_r u_r - \frac{d(h_{rb} u_r)}{dt}\} / \{mH_v(T_b, p, y^i_v) - rx\ mh_l(T_l, p, x^i_c)\} \tag{14}$$

A partial balance is written on each component i : $$\frac{d(x^i_c c)}{dt} = cv\, y^i_v - cl\, x^i_c \text{ .... } i=1, nc \tag{15}$$

The liquid-vapor equilibrium is solved by a call to the data bank : $y_v^i - mk_i(x_r^i, y_v^i, T_b, p)x_r^i = 0$

In the case of the non boiling reaction mixture : cv=cl=0.

This set of differential and algebraic equations is solved using the GEAR method. This model has been validated on both pilot and industrial plants [1].

Modelling for optimisation of batch reactor

Optimisation of batch reactor consists in the determination of temperature or additional reactant rate profiles, or the determination of these two profiles simultaneously. In all the cases. The different variables of the problem are lower and upper bounded. These bounds represent some physical limits (for example, concerning the temperature, the lower bound can represent the minimum temperature of the reaction, the upper bound the boiling point of the solvent).

A classical optimisation problem with bounded and constrained variables takes the form :

$$\begin{cases} \text{Min}\, f(x) & x \in R^m \\ C(x) = 0 & C: R^m \longrightarrow R^p \quad (m \le p) \\ l \le x \le u & l, u \in R^m \end{cases} \tag{16}$$

where C is a vectorial linear/non-linear continuous application, l and u are the lower and upper bounds.

To treat the optimal control problem by nonlinear programming, a finite number of discrete parameters representing the control function u(t), $t \in [t_0, t_f]$ (temperature or feed rate profiles), has to be defined.

For this purpose, the interval $[t_0, t_f]$ is divided into a finite number nint of subintervals. In each subinterval, the control function is represented by a function f as :

$$u(t) = f(t, z_j) \qquad t \in [t_{j-1}, t_j] \qquad j=1,\text{nint} \tag{17}$$

The profiles are defined by the parameters z_j and the switching times t_j. In order to avoid too much sophisticated profiles, f is assumed to be a linear function versus time in each subinterval. The temperature and feed rate variations in the range $[t_{j-1}, t_j]$ are given by : :

$$u(t) = z_{j-1} + (t - t_{j-1})\left(\frac{z_j - z_{j-1}}{t - t_{j-1}}\right) \tag{18}$$

Note that this formulation allows for a discontinuity and change of functional shape in each subinterval. The optimisation procedure needs a model of the reactor. In the first studies the model used was restricted to the mass balance assuming no change of state, only modelling the kinetic aspect (Eq. 1 to 5 with cv=cl=0). Nevertheless to get realistic operating conditions it is necessary to consider the limitations of the heating and cooling capacities of the system fitted out the real industrial reactor. These limitations can be expressed by two ways : constraints on the rate of change in the temperature and a constraint on the rate of the heat generated by the reactions (computed by Eq. 7). Purity constraints can be used as well.

- physical constraints of the process : to limit the rate of heating and cooling with respect to the plant capacities. If we note b_{max} and b_{min} the upper and lower bounds of these variations, the constraints can be expressed by :

$$(z_j - z_{j-1}) - b_{max}(t_j - t_{j-1}) < 0 \text{ and } (z_{j-1} - z_j) + b_{min}(t_j - t_{j-1}) < 0$$

- constraint to control the amount of introduced reactant :

- if the amount of reactant is fixed and equal to M $\quad \sum_{j=1}^{nint} \int_{t_{j-1}}^{t_j} \phi_j(t, z_j)dt = M$

- or if the amount of reactant has to be optimised $\quad \sum_{j=1}^{nint} \int_{t_{j-1}}^{t_j} \phi_j(t, z_j)dt \le M_{max}$

where M_{max} is the maximum amount of introduced reactant.

- constraints of purity restricting the formation of a by-product W under a threshold H_1: $C_W \le H_1$
- constraints on the formation of the desired product P in order to obtain a concentration above a threshold H_2: $C_P \ge H_2$
- constraints on the rate of the heat generated by the reactions in order to avoid runaway conditions :

$$Q_{max} \le H_3$$

The objective function, given by a simulation model, is an implicit function of the variables. It could be a simple objective function to be maximised, for example the concentration of the desired product. Other types of functions could be chosen, depending upon the nature of the problem : it could be the maximisation of the yield versus to the total quantity of reactant initially present in the reactor or versus to the total quantity of added reactant. Some other objective functions including economic factors may be interesting, for example, when the cost of the reactants is the controlling factor or when it is economically advantageous to minimise the operation time or the operating costs.... The problem is now to compute the variables z_j (temperature and feed rate) at each bound of each subinterval, that minimise the criterion subject to the bounds and imposed constraints. The optimal control problem is solved by nonlinear programming. A sequential quadratic programming (SQP) procedure [3] has been used as it is considered as the most effective method for NLP problems. This procedure has been used for optimal conditions of industrial reaction involved in the production of a key component produced by the pharmaceutical company Sanofi [7]. An example of results is given in figures 2 and 3. Figure 2 shows the temperature and the feed rate profile found by the optimisation procedure with a constraint on the heat generation rate of 200 kW which corresponds to the maximum cooling capacity of the industrial reactor. In the most part, this heat generation is always close to this limit (Fig. 3).

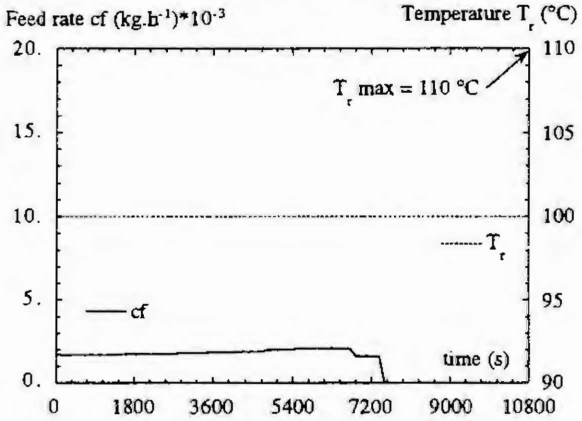


Figure 2 : Optimal operating conditions
Safety constraint on the thermal flux : 200 kW

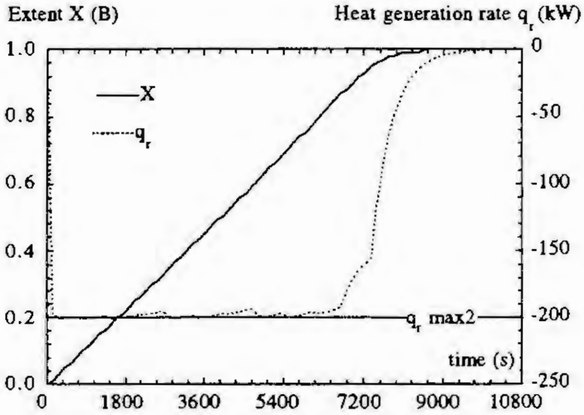


Figure 3 : Time evolution of the reaction extent and of the heat generation

These conditions leads to divided the operating time by 4 with keeping the same yield 84 %.

Present studies deal with incorporating the heat balance, at least on the reaction mixture to take into account the transfers between the mixture and the thermal fluid. This allows to determine optimal operating conditions for a particular operating mode such as isoperibolic corresponding to have a thermal fluid in the jacket flowing at a given temperature. If the heat transfers are well modelled (heat balance on the jacket)we can go further and determine the temperature profile of the thermal fluid instead of the reactor temperature. This will leads to a pure optimal control solution. By this way, this could yield a preliminary set-point profile susceptible to be given to the controller which would reacts against only plant model mismatches, but the most important part of the "work" should be made by the optimisation procedure.

Modelling for control of batch reactor

The temptation in control of batch reactor has been to use the simplest model i.e. the classical input-output model (called also black box model). In the first works related to thermal control of batch reactor performed in our laboratory, we applied the adaptive Generalized Predictive Controller with Reference model [2,4] to pilot and industrial reactors (1000 l glass-lined reactor) fitted out with several thermal systems [5] In the case of pilot plants the results were very satisfactory as we were able to operate the reactor for different conditions (temperature profile, chemical reactions, ...) with the same controller design parameters [5]. We should have thought that we reached our objective which was to develop a controller as flexible as the batch reactor is. But, in the case of the industrial reactor fitted out with the multifluid system the results was not so good. It has revealed that the black box model didn't model the transfers between the reaction mixture and the jacket especially the "bain-marie" effect (effect corresponding to a zero utility flowrate but with a jacket full of fluid). Let us recall that in the case of the multifluid system the manipulative variable is the fluid flowrate. This heat transfer between the jacket and the reaction mixture represents the main limiting dynamics in an industrial-scale reactor. So, we developed a "realistic model-based controller" [5]. This controller is always based on the GPC structure but the nonlinear model used is obtained from the heat balance on the reaction mixture and the jacket.

$$m_r Cp_r \frac{dT_r}{dt} = U_{rj} a_{rj} \left(T_j - T_r\right) + \Delta H' + F_c Cp_c \left(T_c - T_r\right) \tag{19}$$

$$m_j Cp_j \frac{dT_j}{dt} = F_j \rho_j Cp_j \left(T_{in} - T_j\right) - U_{rj} a_{rj} \left(T_j - T_r\right) \tag{20}$$

Furthermore, such a model shows that the manipulative variable is not a classical one. Effectively as the temperature of the fluid is constant whatever the thermal fluid flowrate is the reaction mixture in the case without chemical reaction reaches the temperature of the fluid. The value of the flowrate doesn't influence this final state but affects the "speed" to reach the final state. In other words, the manipulative variable is the time constant of the process called also the residence time. This is true for all processes such as the manipulating variable is the flowrate (i.e. the residence time).

For comparison, figure 4 and 5 give the results obtained for both models, respectively for the black box model and the realistic model. In the latter case, the use of a realistic model makes possible to develop a model based-supervision which overcome the temperature overshoot previously observed by predicting the time at which the valve of hot water has to be shut-off. One can see the real benefit to use a model which respects the structure of the process. Moreover, it is much easier to initialise the model parameters as they are related to physical coefficients that, even if they are not known precisely , we have an order of magnitude.

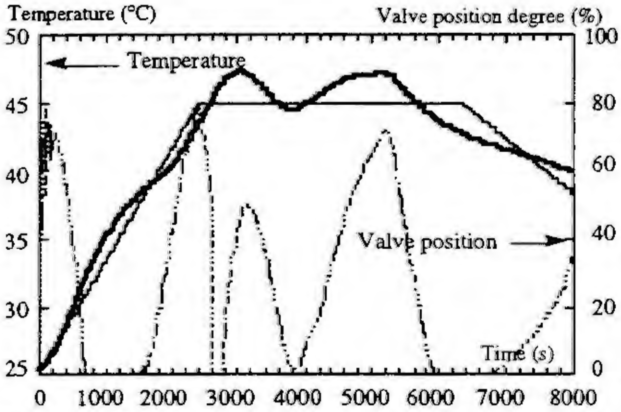


Figure 4 : Experimental results with input-output model

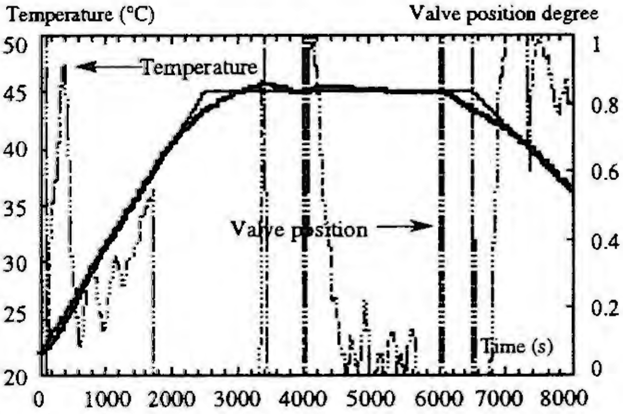


Figure 5 : Results with realistic model+supervision

Recent studies [6] have shown that it is possible to have a linear model of the process by considering as manipulative variable the heat to be transferred through the jacket. This control variable is then used in a cascade structure with as slave controller, a heat transfer model of the jacket which computes the flowrate or the temperature (depending of the configuration of the heating-cooling system) to apply to get this control value. This structure is very interesting also to perform automatically the changeovers of fluids. They are done only when the heat transfers could not be done by the actual fluid and not by predefined time or temperature.

Our studies are now devoted to nonlinear control by taking as model a set of three differential equations: Eq. 19, Eq. 20 and one relative to the description of the hydrodynamics of the jacket. This latter one will allows to take the filling-up and the purge steps as operation steps of the process as the controller will be able to act on the flowrate during these steps contrary as what it was done previously (the flowrate is kept constant and so the reactor is no more under control). A special care is brought to get a "rough" model of the kinetic as a relation linked the heat generated during the reaction to the reactant feeding rate. This model will be included in the prediction model to better anticipate on the reaction behaviour.

References

1. Cabassud M. , M.V. Le Lann, B. Ettedgui and G. Casamatta A general simulation model of batch chemical reactors for thermal control investigations, *Chem. Engng Technol.*,17 (4), 255-268, 1994
2. Clarke D.W, C. Mohtadi, and P.S. Tuffs,, Generalized Predictive Control, *Automatica,* **23** (2), 126-137, 1987
3. Gill P.E.,W. Murray, M.A. Saunders and M.H. Wright,1985, *Software and its relationship to methods in numerical optimization*, P.T. Boggs, R.H. Byrd and R.B. Schnabel, Eds. SIAM, Philadelphia, 1985.
4. Irving E., C.M. Falinower and C. Fonte, Adaptive Generalized Predictive Control with multiple reference model, Proc. of "*2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing*", Lund, Sweeden, 26-06-86, 1986
5. Le Lann M.V., M. Cabassud and G. Casamatta, Adaptive Model Predictive Control, in *R. Berber (ed.) Methods of Model Based Process Control,* Kluwer Academic Publishers, Dordrecht, pp. 426-458, 1995
6. Louleh Z., M. Cabassud, M.V. Le Lann, A. Chamayou, G. Casamatta, A new heating-cooling system to improve controllability of batch reactors, *Chemical Engineering Science*, **51** (11), 3163-3168, 1996.
7.Toulouse C., J. Cezerac, M. Cabassud, M.V. Le Lann et G. Casamatta, Optimisation and scale-up of batch chemical reactors impact of safety constraints, *Chemical Engineering Science,* **51** (10), p. 2243-2252,1996.

OBSERVERS SYNTHESIS METHODOLOGY FOR CHEMICAL REACTORS

A.M. Gibon-Fargeot, F. Celle-Couenne and H. Hammouri

LAGEP UPRES A Q CNRS 5007 - Bât. G - 3^{e}étage Université claude Bernard - CPE Lyon
43, bd du 11 Novembre 1918 - 69 622 Villeurbanne cedex FRANCE
Tel : (33) 04 72 43 18 45, Fax : (33) 04 72 43 16 82 e-mail : couenne@lagep.univ-lyon1.fr

Abstract. We propose a methodology for observer synthesis for a class of chemical reactions processes. By using high gain observers in the mono output case and Kalman observers and also some asymptotic estimations coming from some attractivity property for this class of processes , we obtain in many cases, the estimation of the whole state even if some part of the model is not observable. For this purpose, we use cascades of the previous observers and estimators. Moreover we present the methodology via some examples involving the main reaction schemes.

1 Introduction

By the use of simple observers for nonlinear systems we propose solutions for estimation of concentrations for a class of chemical reactors. The methodology is presented on the basis of simple examples which permit to enlighten the main reaction schemes : series, parallel, reversible and some associations of these latter. Moreover we forecast the cases where the system is partially observable.

For non observable states (or in the case classical observers will not work for these states) we show that when the reactor temperature is measured, it is often possible to obtain an asymptotic estimation of these states. The originality of this note comes from the use of this attractivity property inherent in this class of processes to palliate the lack of observers and from the mixing of these different methods in order to obtain an asymptotic estimation of the whole state vector.

In section 2 we present the general model of the chemical processes that we consider. In section 3 we review the main class of observers that we will use and also we present the attractivity property that we will also use to estimate concentrations. In section 4 we give the results for cascade estimators and we present the main reaction schemes. Following the measure, we propose one solution for estimation problem.

2 Modelling of the chemical process

This modelling concerns the process formed by the reactor and the chemical reactions taking place in this reactor. We consider homogeneous ideal stirred tank. The reactor volume V is constant.
R independant chemical reactions with N ($N \geq R$) species (reactants and products) take place in the reactor. From mass and energy balances, we obtain the following model ([6]) :

$$\left\{ \begin{array}{ll} \dfrac{dC}{dt} & = \frac{D}{V}(C_{in}(t) - C(t)) + K\varphi(C,T) \\ \dfrac{dT}{dt} & = \frac{D}{V}(T_{in}(t) - T(t)) - \frac{\Delta H^T}{\rho C_p}\varphi(C,T) - Q(T(t) - T_j(t)) \end{array} \right. \tag{1}$$

where C (dim$C = N$) is the concentration vector, T (Kelvin degree) is the temperature in the reactor. The dimension of the state of (1) is then equal to $n = N + 1$. C_{in} is the inlet concentration vector (dim$C_{in} = N$). T_{in} is the inlet reactor temperature and T_j the coolant temperature. D is the reactor flow rate. K (dim$K = N \times R$) is the stoichiometric coefficient matrix. The entries k_{ij} of K are positive if the species j is a product for the reaction i, negative if it is a reactant and 0 if j does not appear in the reaction i. $Q = \frac{UA}{\rho V C_p}$ with U the overall heat transfer, A the heat transfer area, ρ the reactor contents density and C_p the reactor contents heat capacity. ΔH (dim $\Delta = R$) is the heat reaction vector. The reaction rate vector φ (dim$\varphi = R$) which is a nonlinear function of C and T verifies the following properties ([9]) :

1. The physical positive quadrant $\mathcal{P} = \{ \begin{pmatrix} C \\ T \end{pmatrix} \; T \geq 0 \; C_i \geq 0 \; i = 1, \cdots, N\}$
2. $\phi_i(C,T) \geq 0 \; \forall \begin{pmatrix} C \\ T \end{pmatrix} \in \mathcal{P} \; i = 1, \cdots, R$
3. $\varphi(C,0) = 0$
4. $\psi(C) \geq \|\varphi(C,T)\|$ for all physical T with $\psi(C)$ is a positive bounded function

In this case one can easily show ([9]) that all the trajectories $\begin{pmatrix} C \\ T \end{pmatrix}$ of the system (1) issued from the physical domain $\mathcal{P}$ are bounded when applied inputs are bounded.

3 Observer synthesis

In this section we review the most representative class of global (in the sense that the estimation error tends to zero whatever the initialization of the observer is made) observer and also the attractivity property intrinsic to the class of processes under consideration. This second point is especially relevant in the case the system is not observable or global observers cannot be synthesized.

Kalman-like observer ([1] [3] [4] [8])

This class of observer works for observable systems of the form (with $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$):

$$\left\{ \begin{array}{ll} \dot{x}(t) & = A(u(t), y(t))x(t) + g(u(t), y(t)) \\ y(t) & = Cx(t) \end{array} \right. \tag{2}$$

For simplicity we consider only direct measures of concentrations or temperatures (C is a constant matrix). Moreover A and g are continuous with respect to u and y and $|u\|_{L^\infty} < u_{max}$. The observer is given by :

$$\left\{ \begin{array}{ll} \dot{\hat{x}}(t) & = A(u(t), y(t))\hat{x}(t) + g(u(t), y(t)) - S^{-1}C^TQ(C\hat{x} - y) \\ \dot{S}(t) & = -\theta S(t) - A^T(u(t), y(t))S(t) - S(t)A(u(t), y(t)) + C^TQC \\ S(0) & = S_0 \end{array} \right. \tag{3}$$

(3) is an exponential observer as soon as the enlarged input $w = \begin{pmatrix} u \\ y \end{pmatrix}$ is regularly persistent, $\theta > 0$ and S_0 and Q are chosen symmetric positive definite.

Remark 1 *An admissible input w is said to be regularly persistent if the corresponding noncontrolled linear time varying system is uniformly completely observable (see [1]).*

Example 1 *[8]*
This observer can be applied for the estimation of C for endothermic or exothermic reaction scheme of the form : $C_1 \overset{\alpha_1=1}{\longrightarrow} C_2$ with the reaction rate of first order $(= \alpha_1)$ and the process kinetics obeying the Arrhenius law and the on-line measure of T.
Or for $C_1 \overset{\alpha_1=1}{\longrightarrow} C_2 \overset{\alpha_2=1}{\longrightarrow} C_3$ for the estimation of C_1 and C_2 with Arrhenius law and on-line measures of T.

High gain observer ([7] [8] [10])

We restrict our attention to the single output case ($y \in \mathbb{R}$) for control affine systems :

$$\left\{\begin{array}{l} \dot{x}(t) = f(x(t)) + \displaystyle\sum_{1\leq j\leq m} u_j(t).g_j(x(t)) \\ y(t) = h(x(t)) \end{array}\right. \quad x(t) \in \mathbb{R}^n, y(t) \in \mathbb{R} \tag{4}$$

where f and g_j are sufficiently differentiable.

This second class of observer works for uniformly observable systems, that is to say observable systems such that all admissible input $u(.)$ is universal [7]. Moreover such systems can be written thanks to some coordinate change $\Psi(x) = (h(x), L_f(h(x)), \cdots, L_f^{n-1}(h(x)))^T$ to the form (with $z \in \mathbb{R}^n$)

$$\left\{\begin{array}{l} \dot{z}(t) = Az(t) + B(z(t)) + \displaystyle\sum_{1\leq j\leq m} B_j'(z(t))u_j(t) \\ y(t) = Cz(t) \end{array}\right. \tag{5}$$

with : $z(t) = \begin{pmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{pmatrix} \in \mathbb{R}^n$, $y(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}^m$ and $A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}$,

$B = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ b(z) \end{pmatrix}$, $\forall j \in \{1, \ldots, m\}$ $B_j' = \begin{pmatrix} b_{1,j}'(z_1) \\ b_{2,j}'(z_1, z_2) \\ \vdots \\ \vdots \\ b_{n,j}'(z) \end{pmatrix}$ and $C = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}$.

In this case, if moreover $B(z)$ and $B_j(z)$ $\forall j \in \{1, \cdots, m\}$ are globally Lipschitzian function then :

$$\left\{\begin{array}{ll} \dot{\hat{z}}(t) = A\hat{z}(t) + B(\hat{z}(t)) & + \displaystyle\sum_{1\leq j\leq m} B_j'(\hat{z}(t))u_j(t) + \Delta K(C\hat{z}(t) - y(t)) \\ \hat{z}(0) = \hat{z}_0 & \qquad \textit{the initial condition} \end{array}\right. \tag{6}$$

with $\Delta = \begin{pmatrix} \theta & 0 & \cdots & \cdots & 0 \\ 0 & \theta^2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \theta^{n-1} & 0 \\ 0 & \cdots & \cdots & 0 & \theta^n \end{pmatrix}$ et $K = \begin{pmatrix} k_1 \\ \vdots \\ k_n \end{pmatrix}$ is an exponential observer for (5) for sufficiently large θ and for K such that the matrix $(A + KC)$ be stable.

Remark 2 *Notice that if the trajectories remain in a compact Ω the observer will work as soon B and the Bj's can be modified outside Ω such that they are globally Lipschitzian on $\mathbb{R}^n$. The observer for the former system is then given by :*

$$\left\{\begin{array}{ll} \dot{\hat{x}}(t) = f(\hat{x}(t)) & + \displaystyle\sum_{1\leq j\leq m} u_j(t).g_j(\hat{x}(t)) + \left(\frac{\partial\Psi}{\partial x}(\hat{x}(t))\right)^{-1} \Delta K(h(\hat{x}(t)) - y(t)) \\ \hat{x}(0) = \hat{x}_0 & \qquad \textit{the initial condition} \end{array}\right. \tag{7}$$

Example 2 *[8]*
This will work for endothermic or exothermic reaction scheme $C_1 \overset{\alpha_1\neq 1}{\longrightarrow} C_2$ for estimation of C_1 with Arrhenius law and on-line measure of T.

Open-loop estimation ([6] [10] [11])

In this section we present an open-loop estimation based on the knowledge of the on-line measures, the model and some intrinsic property of this class of processes. This property will permit to reduce the complexity of the problem of estimation in order to use the observers described above and also to get round the unobservability of some part of the state of the process. This property is connected to the result presented in [6] but more general.

Definition 1 *A dynamical system is said to be open-loop attractive (OLA) if the deviation between two trajectories issued from any couple of different initial conditions tends asymptotically to zero when t tends to infinity when the same inputs are applied.*

That is to say if a system is OLA, it suffices to know the evolution of the inputs u on a sufficiently long time to know the exact values of the states.

We will use this property not directly on the system (1) but only on some of the nonmeasured states of (1). First let us consider

$$\dot{\tilde{C}} = f(\tilde{C}, w) \tag{8}$$

with the augmented input vector $w^T = (u^T, y^T)$ and $\tilde{C}$ is a part of C . This system (8) will be referenced as the fictitious system relatively to $\tilde{C}$ and issued from (1). Now suppose that (8) is OLA and it exists an additionnal differential equation on C_i (C_i is not a component of $\tilde{C}$) which depends on w, $\tilde{C}$ and C_i. If we exclude autocatalytic reactions, C_i appears in its differential equations in the following manner :

$$\dot{C}_i = \Phi(\tilde{C}, w) - \Gamma(C_i, \tilde{C}, w) - D/VC_i \tag{9}$$

where Φ and Γ are positive functions. Since the system (8) is OLA, from some time $t \geq \bar{T}$ the equation (9) gives :

$$\dot{C}_i = \bar{\Phi}(t, w) - \bar{\Gamma}(C_i, t, w) - D/VC_i$$

1t follows the obvious proposition :

Proposition 3.1 *The fictitious system formed by (8) and (9) and whose state is given by $\begin{pmatrix} \tilde{C} \\ C_i \end{pmatrix}$ is OLA if $\frac{\partial \Gamma}{\partial C_i} > 0 \; \forall w, t$. In this case the attractivity rate is bounded by $k \exp -\frac{D_{min}}{V}$ for some k and $t \geq \bar{T}$ with $D(t) \geq D_{min} \; \forall \; t$*

Remark 3 *the proof follows from the fact that inputs and states are bounded. This comes from properties given in section 2.*

Corollary 3.2 *With reaction rate coming from Arrhenius law, any one-dimensionnal fictitious system of the form $\dot{C}_i = f_i(C_i, w)$ is OLA as soon the orders of reaction with respect to C_i are greater than 0.*

Corollary 3.3 *Any OLA fictitious system of the form (8) is an asymptotic estimator for the concentration $\tilde{C}$.*

Example 3 *Consider the reaction scheme $C_1 \xrightarrow{\alpha_1 = 2} C_2 \xrightarrow{\alpha_2 = 1} C_3$ with kinetics given by $\varphi_1 = k_{01} C_1^2 \exp \frac{E_1}{RT}$ and $\varphi_2 = k_{02} C_2 \exp \frac{E_2}{RT}$. The on-line measure is only T and we desire to estimate all the concentrations. One can easily verify that the fictitious system formed by C_1, C_2, C_3 with the measure of T is OLA. Notice that C_1 is OLA and also C_1, C_2.*

4 Methodological synthesis for observer for chemical reactors ([10][11])

Cascade estimators ([5])

Consider the following system (with $x_1 \in \mathbb{R}^{n1}$ and $x_2 \in \mathbb{R}^{n2}$) :

$$\left\{ \begin{array}{ll} \left. \begin{array}{ll} \dot{x}_1 & = f_1(x_1, u) \\ y_1 & = x_{11} \in \mathbb{R} \end{array} \right\} (10).1 \\ \left. \begin{array}{ll} \dot{x}_2 & = f_2(x_1, x_2, u) \\ y_2 & = x_{21} \in \mathbb{R} \end{array} \right\} (10).2 \end{array} \right. \tag{10}$$

for which we suppose all the components of the state are bounded.

Proposition 4.1 *Suppose that (10).1 is either observable and observer synthesis can be made with Kalman-like or high-gain observer or is OLA. Suppose that (10).2 with the knowledge of x_1 is either observable and observer synthesis can be made with Kalman-like or is high-gain observer or OLA then the cascade of the two estimators will give an asymptotic estimator for (10).*

Methodology on the basis of examples

In any case we suppose to do the on-line measure of the reactor temperature and orders of reaction are correct with respect the previous results.

Example 4 $C_1 \overset{\longleftarrow}{\longrightarrow} C_2 \longrightarrow C_3.$

The equations are given by :

$$\left\{\begin{array}{ll} \dot{C}_1 & = -k_{01}\exp-\frac{E_1}{RT}C_1^{\alpha_{11}} + k_{02}\exp-\frac{E_2}{RT}C_2^{\alpha_{22}} + \frac{D}{V}(C_{1_{in}} - C_1) \\ \dot{C}_2 & = k_{01}\exp-\frac{E_1}{RT}C_1^{\alpha_{11}} - k_{02}\exp-\frac{E_2}{RT}C_2^{\alpha_{22}} - k_{03}\exp-\frac{E_3}{RT}C_2^{\alpha_{32}} + \frac{D}{V}(C_{2_{in}} - C_2) \\ \dot{C}_3 & = k_{03}\exp-\frac{E_3}{RT}C_2^{\alpha_{32}} + \frac{D}{V}(C_{3_{in}} - C_3) \end{array}\right.$$

- *C_3 is measured. The system is uniformly observable.*
- *C_2 is measured.C_3 is OLA with the measure of C_2. The system $\begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$ is uniformly observable.*
- *C_1 is measured.$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$ is uniformly observable. C_3 is OLA with the estimation of C_2.*

Example 5 $C_1 \longrightarrow C_2 \overset{\longleftarrow}{\longrightarrow} C_3$

The process is described by :

$$\left\{\begin{array}{ll} \dot{C}_1 & = -k_{01}\exp-\frac{E_1}{RT}C_1^{\alpha_{11}} + \frac{D}{V}(C_{1_{in}} - C_1) \\ \dot{C}_2 & = k_{01}\exp-\frac{E_1}{RT}C_1^{\alpha_{11}} - k_{02}\exp-\frac{E_2}{RT}C_2^{\alpha_{22}} + k_{03}\exp-\frac{E_3}{RT}C_3^{\alpha_{33}} + \frac{D}{V}(C_{2_{in}} - C_2) \\ \dot{C}_3 & = k_{02}\exp-\frac{E_2}{RT}C_2^{\alpha_{22}} - k_{03}\exp-\frac{E_2}{RT}C_3^{\alpha_{33}} + \frac{D}{V}(C_{3_{in}} - C_3) \end{array}\right.$$

- *C_3 is measured. The system is uniformly observable.*
- *C_2 is measured.C_3 is OLA with the measure of C_2. The system $\begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$ is uniformly observable with the injection of the estimation of C_3.*
- *C_1 is measured. $C_2 + C_3$ is OLA.*

Example 6 $C_1 \xrightarrow{R_1} C_2 \xrightarrow{R_2} C_3 \cdots \xrightarrow{R_{n-1}} C_n$

When the measure is C_i the system formed by C_1 to C_i is uniformly observable and the system formed C_{i+1} to C_n is OLA with the measure of C_i.

Example 7

$$\begin{tikzcd} C_1 \arrow[r, "R_1"] \arrow[dr, "R_{n_1}"'] & C_2 \arrow[r, "R_2"] & C_3 \cdots \arrow[r, "R_{n_1-1}"] & C_{n_1} \\ & C_{n_1+1} \arrow[r, "R_{n_1+1}"] & C_{n_1+2} \cdots \arrow[r, "R_{n-1}"] & C_n \end{tikzcd}$$

- *If the on-line measure is C_1 the systems formed by C_2 to C_{n_1} and by C_{n_1+1} to C_n are OLA with the measure.*
- *If the measure is C_i, $i \in \{2, \cdots, n_1\}$, the system formed by C_1 to C_i is uniformly observable. The system formed by C_{i+1} to C_{n_1} is OLA with the measure and the system formed by C_{n_1+1} to C_n is OLA with the estimation of C_1.*

Example 8 $C_1 \longrightarrow C_2$
$C_2 + C_3 \longrightarrow C_4$

- *if* $y = C_1$, $\begin{pmatrix} C_1 \\ C_2 - C_3 \\ C_3 \\ C_4 \end{pmatrix}$ *is OLA. So* $\begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{pmatrix}$ *is also OLA.*
- *if* $y = C_2$, $\begin{pmatrix} C_3 \\ C_4 \end{pmatrix}$ *is OLA and* $\begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$ *is uniformly observable with the injection of the estimation of* C_3.
- *if* $y = C_3$, $\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix}$ *is uniformly observable and* C_4 *is OLA with the injection of* C_2 *and the measure.*

5 Conclusion

In this note we propose on the basis of simple examples how to use global observers with some open-loop estimators. In the previous examples we do not explicitely present systems for which Kalman-like observers will work. This comes from the fact these latter work only for first order reaction with respect to the nonmeasured concentration. This can be easily verified on differential equations.

References

[1] R.E. Kalman, R.S. Bucy, New results in linear filtering and predicting theory, Journal of basic Engineering, 1960, 35-40.

[2] D.G. Luenberger, Observers for multivariable systems, IEEE Trans. on Automatic Control, 11, 1966, 190-197.

[3] G. Bornard, F. Celle, N. Couenne, Regularly persistent observers for bilinear systems, Lectures Notes in Control and Information Sciences, vol 22, 1988.

[4] N. Couenne, Synthèse d'observateurs de systèmes affines en l'état en l'état- Etude du flot singulier des équations d'Euler, Ph. D. thesis, Institut National Polytechnique de Grenoble, 1988.

[5] H. Hammouri, F. Celle and J. De Leon Morales, Cascade observers for some multioutput nonlinear systems, Proceedings of the first ECC, 1991.

[6] D. Dochain, M. Perrier and B.E. Ydstie, Asymptotic observers for stirred tank, Chemical Engineering Science, 1992, 47(15/16), 4167-4177.

[7] J.P. Gauthier, H. Hammouri and S. Othman, A simple observer for nonlinear systems: application to bioreactors, IEEE Trans. on Automatic Control, 7(12), 1992, 1970-1974.

[8] A.M. Gibon-Fargeot, H. Hammouri and F. Couenne, Nonlinear observers for chemical reactors, Chemical Engineering Science, 49(14), 1994, 2287-2300.

[9] J. Alvarez-Ramirez, Observers for a class of continuous tank reactors via temperature measurement, Chemical Engineering Science, 50(9), 1995, 1393-1399.

[10] A.M. Gibon-Fargeot, Synthèse d'observateurs pour le génie des procédés, Phd thesis, Université Claude Bernard Lyon 1, 1995.

[11] A.M. Gibon-Fargeot, F. Couenne-Celle, H. Hammouri, Global observers for CSTR's : a structural approach, Internal report, LAGEP, 1996.

DESIGN OF THE INITIATION OF BATCH POLYMERISATION REACTORS

S. Németh and P. Árva
University of Veszprém
Egyetem u. 10, H-8200 Veszprém

Abstract. One of the most important step of safe and profitable operation of batch polymerisation reactors is the initiation. In this paper a general model is formulated to describe the free radical polymerisation of several vinyl monomers. A method based on this model is introduced to choose the initiator and its initial concentration in order to minimise the polymerisation time in an industrial batch reactor. The method is demonstrated and tested by simulation.

Introduction

At the design and operation of batch polymerisation reactors several difficulties must be considered:

The physical properties of reacting system change highly during the polymerisation. These changes influence the rate of the polymerisation reactions, the properties of polymer products and the overall heat transfer coefficient of the reactor.

Polymerisation is mostly strongly exothermic reaction so great amount of heat has to be removed out of the reactor. The volumes of the batch polymerisation reactors are about 100 - 200 m^3. Increasing the size of the reactor decreases its surface area to volume ratio and renders the heat removal more difficult.

At higher conversion the polymerisation reactions become diffusion controlled. This limitation leads to a dramatic decrease in termination rate resulting in an autoacceleration of polymerisation process. Several empirical and semi-empirical relationship have been suggested in the literature to take into account this phenomena.

The polymerisation reactions are very sensitive for the temperature, the applied catalysator, initiator, and the inhibitors.

An additional difficulty is given by the fact that several process variables, that are indicate the properties of polymer product, can not be available in real time because of the lack of on-line sensors for rapid measurements.

Due to these difficulties the design and operation of batch polymerisation reactors have received much attention in the last years. Several papers deal with the optimal design, optimal control and the study of runaway characteristics of these types of reactors [1].

Fauske and Leung [2] introduced a method to design the emergency relief systems of polymerisation reactors. Maschio and Zanelli [3] studied the runaway characteristic of methylmethacrylate (MMA) polymerisation reactor and the effect of using different initiator. The introduced model has been an adequate tool for analysing the MMA polymerisation reactor. In our earlier paper suspension polymerisation of stryrene was studied [4]. Based on a rigorous mathematical model an investigation has been made to predict the runaway ability of this reactor, and the runaway index was defined to evaluate the runaway ability under different operation conditions.

Beside the safety operation, the tailor-made polymer production is also a requirement. The properties of the polymer are developed during the reaction. Thermal and mechanical properties of polymer product are correlated with the value of the molecular weight distribution which mainly depend on the temperature of the reactor. Scali [5] presented a method to control the quality of product in batch polymerisation. On the basis of the kinetic model an optimal temperature profile is predetermined to achieve the product with constant value of molecular weight during the course of the reaction.

The productivity of batch polymerisation reactors can be increased by the application of bifunctional initiators. The reaction rate decreases at the end of the polymerisation process when monofunctional initiators are applied. Applying bifunctional initiators the reactor productivity is increased by about 25 % [6].

In this paper a method, based on a general model of free radical polymerisation of vinyl monomers, is introduced to choose the initiator and its initial concentration in order to minimise the polymerisation time in an industrial batch reactor.

Development of a general model for free radical polymerisation of vinyl monomers

Free radical mechanism of polymerisation of vinyl monomers consists of four main elementary steps, initiation, propagation, chain transfer and termination (Table 1.).

Table 1. Typical reaction mechanism for free radical polymerisation of vinyl monomers.

Initiation	
	$I \xrightarrow{k_I} 2I^{\bullet}$
	$I^{\bullet} + M \xrightarrow{k_i} R_1$
Propagation	$R_i + M \xrightarrow{k_p} R_{i+1}$
Chain transfer	
	$R_i + M \xrightarrow{k_{trM}} R_1 + P_i$
	$R_i + S \xrightarrow{k_{trS}} R_1 + P_i$
Termination	
	$R_i + R_j \xrightarrow{k_{td}} P_j + P_i$
	$R_i + R_j \xrightarrow{k_{tc}} P_{i+j}$

Where I is the initiator, $I^{\bullet}$ is the free radical species, M is the monomer, R_i is the growing polymer chain, S is the chain transfer agent, P_i is the dead polymer species.

Decomposition of the initiator, an azo or peroxo compound, follows first order kinetics. The free radical can react very rapidly with a monomer molecule to form a monomer radical and sometime it disappears by side reactions. If a monomer molecule added to a radical it gives a larger radical which can react with a further monomer molecule. Radicals species can be transferred to monomer, solvent, chain transfer agents forming a long dead polymer and a monomer radical. If two radicals react whith each other then final polymer molecules will produced at the expense of radical. The chain termination reaction is inhibited by the so called gel or Trommsdorff effect. This phenomenon is described by a relationship depending on the temperature and the conversion.

In deriving simpler kinetic relations three simplifying assumptions are normally made:

The length of the polymer chain is large so the total rate of monomer consumption may equal to the rate of consumption in the propagation alone.

The rate constants of termination and propagation are independent of the chain length.

The quasi steady-state approximation of live radicals is applied.

On this basis, the dynamic simulation model for batch polymerisation can be derived. Applying these assumptions the component balance equation for initiator, monomer and the total amount of growing polymer chain $\left(R = \sum_{i=1}^{\infty} R_i\right)$ can be written:

$$\frac{dVI}{dt} = -Vk_I I; \quad t = 0,\ VI = (VI)^0; \tag{1}$$

$$\frac{dVM}{dt} = -Vk_p MR; \quad t = 0,\ VM = (VM)^0; \tag{2}$$

$$R = \sqrt{\frac{2fk_I}{k_t} I}; \tag{3}$$

where f is the initiator efficiency, V is the volume of the reaction mixture.

The volume of the reaction mixture decreases during the polymerisation course because of the density of the producing polymer is higher than the density of the monomer. Density of the reaction mixture is

$$\frac{1}{\rho(x)} = \frac{1-x}{\rho_M} + \frac{x}{\rho_P}, \tag{4}$$

while a volume of the reaction mixture is

$$V(x) = V_0\left(1 - \frac{\rho_P - \rho_M}{\rho_P}x\right) = V_0(1 - Bx), \qquad (5)$$

where ρ_M, ρ_P are the density of the monomer and polymer, V_0 is the initial volume of the reaction mixture, x is the monomer conversion defined as:

$$x = 1 - \frac{VM}{(VM)^0} \qquad (6)$$

The concentration of the initiator can be calculated from equation 1

$$I = \frac{V_0}{V} I_0 e^{-k_I t} = \frac{I_0}{1 - Bx} e^{-k_I t} \qquad (7)$$

Substituting the equations 3, 6, 7 in equation 2 the monomer conversion can be expressed:

$$\frac{dx}{dt} = k_P \frac{1 - x}{\sqrt{1 - Bx}} \sqrt{\frac{2fk_I}{k_t} I_0 y} \qquad (8)$$

$$\text{where } y = \frac{I}{I_0} = e^{-k_I t} \qquad (9)$$

Because of the gel effect the reaction rate constants is calculated as

$$\left(\frac{k_P^2}{k_t}\right)^{1/2} = \left(\frac{k_P^2}{k_t}\right)^{1/2}\Bigg|_{x=0} \cdot g(x) = k \cdot g(x) \qquad (10)$$

where the first term is the net rate constant of the polymerisation reactions at zero conversion, while g(x) is the self-acceleration factor.

Finally the model of batch polymerisation reactor can be written as:

$$\frac{dx}{dt} = k\frac{(1 - x)g(x)}{\sqrt{1 - Bx}}\sqrt{2fk_I I_0 y} = k \cdot F_1(x) F_2(I); \quad t = 0, x = 0; \qquad (11)$$

$$\frac{dy}{dt} = -k_I y; \quad t = 0, y = 1; \qquad (12)$$

In this model k is the net reaction rate constant at zero conversion. The values of the k, at given polymerisation system, depend only on the reactor temperature. Values of the second term ($F_1(x)$) depend only on the monomer conversion, while the values of third term ($F_2(I)$) depend on the type of the initiator (k_I) and its initial concentration (I_0).

This type of model can be applied for either homogeneous [5], and heterogeneous polymerisation processes[7].

Design of the initiation

The equation 12 can be solved analytically. Substituting the solution of equation 12 into equation 11, then it can be integrated:

$$Int(x) = \int_0^x \frac{dx}{F_1(x)} = k\sqrt{2fk_I I_0}\int_0^t e^{-\frac{k_I t}{2}} dt = 2k\sqrt{\frac{2fI_0}{k_I}}\left[1 - e^{-\frac{k_I t}{2}}\right] \qquad (13)$$

The polymerisation time (t) can be expressed as an explicit function of monomer conversion (x) by equation 13; it is given by

$$t = \frac{2}{k_I}\ln\left(\frac{2k\sqrt{\frac{2fI_0}{k_I}}}{2k\sqrt{\frac{2fI_0}{k_I}} - Int(x)}\right), \qquad (14)$$

so equation 14 can be used to predict the polymerisation time that is need to reach the predetermined final conversion (x^*).

To minimise the batch time the next optimisation problem can be formulated:

$$\min_{k_I, I_0} t = \frac{2}{k_I} \ln \left(\frac{\sqrt{\frac{2fI_0}{k_I}}}{\sqrt{\frac{2fI_0}{k_I} - \frac{\mathrm{Int}(x^*)}{2k}}} \right). \tag{15}$$

It can be seen that the batch time is minimal if the value of the $\sqrt{\frac{2fI_0}{k_I}}$ term is large, ie. the initial concentration of the initiator is large or k_I is small. In batch polymerisation, the reaction rate can not exceed a maximum value determined by the cooling capacity of the reactor. Let v_{max} is the allowable maximum value of the reaction rate. Substituting the expression of v_{max} in equation 15, the optimisation problem can be written as

$$\min_{k_I, x} t = \frac{2}{k_I} \ln \left(\frac{\frac{v_{max}}{k_I F_1(x)} + \frac{\mathrm{Int}(x)}{2}}{\frac{v_{max}}{k_I F_1(x)} + \frac{\mathrm{Int}(x)}{2} - \frac{\mathrm{Int}(x^*)}{2k}} \right) \tag{16}$$

In this case the batch time has to be minimised with k_I and x, because the conversion where the value of reaction rate is maximum is not known.

In some special case, it can be determined analytically. Let us consider the numerator of equation 16. As it was mentioned earlier the batch time is minimal if the value of the numerator in equation 16 is large. Now we have to determine whether the value of the numerator is increasing by the conversion or not. Take the first differential of the numerator of equation 16 with respect to conversion (x).

$$\frac{F_{1,x}(x)}{F(x)} < \frac{k_I}{2v_{max}} \tag{17}$$

where $F_{1,x} = \frac{dF_1(x)}{dx}$. It can be seen, that if the inequality 17 is always true in the interval $[0, x^*]$ (case a) then the numerator is increasing by the conversion, so the following optimisation problem is given:

$$\min_{k_I} t = \frac{2}{k_I} \ln \left(\frac{\frac{v_{max}}{k_I F_1(x^*)} + \frac{\mathrm{Int}(x^*)}{2}}{\frac{v_{max}}{k_I F_1(x^*)}} \right), \tag{18}$$

while in the opposite case (case b) the optimisation problem is formulated as:

$$\min_{k_I} t = \frac{2}{k_I} \ln \left(\frac{\frac{v_{max}}{k_I}}{\frac{v_{max}}{k_I} - \frac{\mathrm{Int}(x^*)}{2}} \right) \tag{19}$$

In the case (a) the reaction rate should reach its maximum value at x^*, while in the case (b) the maximum reaction rate is to be designed at the beginning of the process. So the rate constant of initiator decomposition and the initial concentration of the initiator can also be expressed as

case (a):
$$k_I = 2 \frac{F_{1,x}(x^*)}{F_1(x^*)} v_{max} \tag{20}$$

$$I_0 = \frac{1}{4f} \left(\frac{1 + F_{1,x}(x^*) \mathrm{Int}(x^*)}{k \cdot F_{1,x}(x^*)} \right) \frac{F_{1,x}(x^*)}{F_1(x^*)} v_{max} \tag{21}$$

case (b):
$$k_I > 2F_{1,x}(0) v_{max} \tag{22}$$

$$I_0 = \frac{1}{4f} \frac{v_{max}}{k^2 \cdot F_{1,x}(0)} \tag{23}$$

Results of simulation and discussion

The applicability of the method introduced in the previous section is demonstrated by simulation. Experimental and kinetic data for VC polymerisation were taken from the literature [7]. In that paper, the polymerisation of VC initiated by AIBN at different at different levels of temperature was studied. Fig. 1 shows the monomer conversion profile at different I_0, while the reaction rate curves are displayed on Fig. 2.

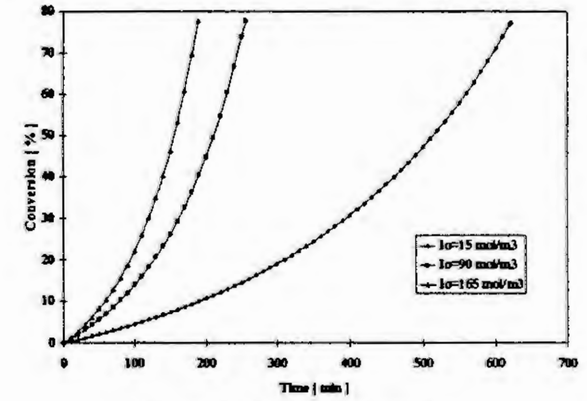


Figure 1. Conversion profile.

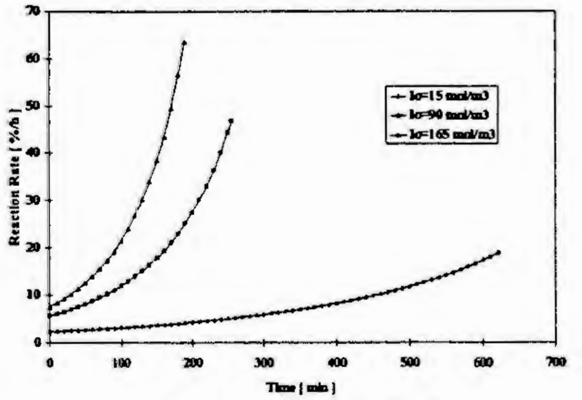


Figure 2. Reaction curves.

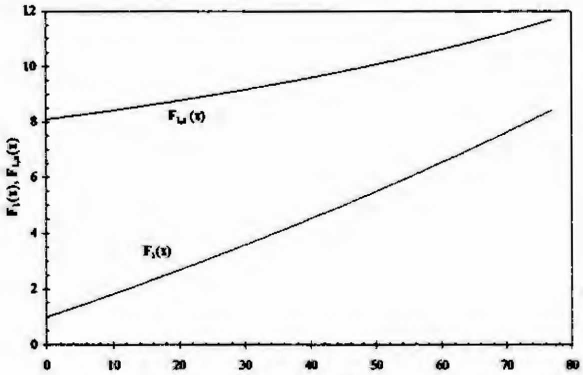


Figure 3. $F_1(x)$ and $F_{1,x}(x)$ functions of VC.

Fig 3. shows $F_1(x)$, and $F_{1,x}(x)$ curves of VC. It can be seen that both functions are increasing in the interval [0 - 77%]. Let us assume that the v_{max} is 20 %/h in industrial batch reactor. In this case the minimal batch time need to reach 77% conversion is 4.8 h. The designed initiator system is the following: k_I is 0.554 h^{-1} and I_0 is 2.54 mol m^{-3}. It can be seen that the value of the rate constant of the decomposition of the designed initiator is much higher than the value of AIBN. Consequently, the initial concentration of the designed initiator is also smaller. Moreover, applying the designed initiator the polymerisation rate is higher at the beginning of the process. Due to the high value of the rate constant of the decomposition the designed initiator nearly totally decomposes by the end of the process, when the gel effect becomes significant, so the net reaction rate of the polymerisation can not increase significantly.

Fig 4. shows the reaction rate curve of the designed polymerisation, while the designed conversion profile can be seen in Fig 5. It can be seen that the profile of the reaction curve is changed. In this case there is no runaway tendency of the polymerisation, so the reactor less sensitive that in the case displayed in Fig. I. and Fig 2. Moreover, the controllability of the reactor is also improved. This figure also shows the sensitivity of the reaction course. Effects of the reactor temperature, the initial concentration of the initiator and the rate constant of initiator decomposition are also investigated. The sensitivity of batch time and reaction rate can also be observed.

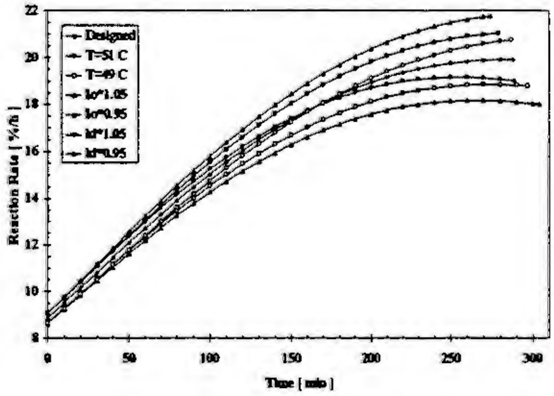


Figure 4. Reaction curves.

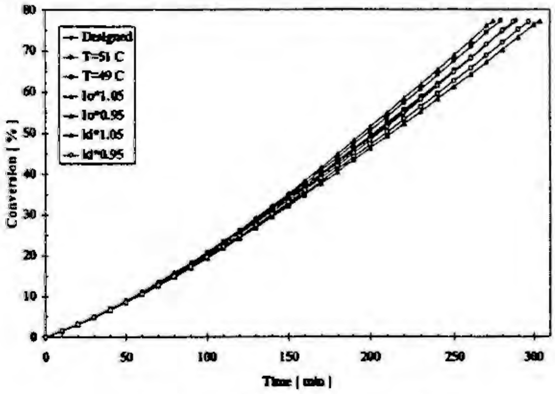


Figure 5. Conversion profiles.

If I_0 is higher than the designed value by 5% then the initial reaction rate is also higher during the polymerisation course and it exceeds the value of v_{max} around 150 min. Because of the higher reaction rate the batch time is smaller. If I_0 is smaller by 5% then the shape of the reaction curve is changed.

At higher level of temperature the initial reaction rate is higher, but at the end of the process it is smaller than in the designed case. At lower level of temperature the initial reaction rate is also lower, but the and of the process it is higher than in the designed case, because the decomposition rate of the initiator is also lower.

Similar results were obtained by the changing of the rate constant of initiator decomposition

Conclusions

One of the key element of the safe and profitable operation of batch polymerisation reactors is the initiation. The reaction rate in the industrial reactor is limited by the capacity of the cooling system. If the heat generation is higher than the maximum cooling capacity then the reactor temperature is increasing resulting in a significant increase in heat production that may lead to thermal explosion of the reactor.

Based on a general model a method was introduced to design the initiation of the batch polymerisation reactors. The method helps for the plant and development engineers to choose the best initiator and its initial concentration in order to minimise the batch time.

The method was tested by simulation.

Acknowledgements

This work was partially supported by the OTKA 2547 Project and Peregrinatio I. Foundation (Veszprém, Hungary). The authors wish to thank for the financial support.

References

1. Cheremisinoff, N., P., Encyclopedia of Engineering Materials, Part A: Polymer Science and Technology, Vol. 1. Synthesis and Properties, Marcel Dekker, New York, 1988.
2. Fauske, K., H., and Leung, J., C., New Experimental Technique for Characteristing Runaway Chemical Reactions, Chem. Eng. Prog. 81, (1985), 5-12.
3. Maschio, G. and Zanelli, S., Modelling of Radical Polymerisation Reactor: Runaway Phenomena in Batch Reactors; Modelling of Radical Polymerization Batch Reactor: The Influence of the Initiator. In: Polymer Reaction Engineering, (Eds.: Reichert, K., H. and Geiseler, W.) VCH Verlagsgesellschaft mbH, Weinheim, 1989, 94-104, 178-186.
4. Németh, S. and Thyrion, F., C., Study of Runaway Characteristic of Suspension Polymerisation of Styrene, Chem. Eng. Technol., 18, (1995), 315-323.
5. Scali, C., Ciari, R., Bello, T. and Maschio, G., Optimal Temperature for the Control of the Product Quality in Batch Polymerization: Simulation and Experimental Results, J. Appl. Polym. Sci., 55, (1995), 945-959.
6. Villalobos, M., A. and Hamielec, A, E., Bulk and suspension polymerization of styrene in the persence of n-pentane: an evalution of monofunctional and bifunctional initiation, J. Appl. Polym. Sci., 50, (1993), 327-343.
7. Abdel-Alim, A., H. and Hamielec, A., E., Bulk Polymerization of Vinyl Chloride, J. Appl. Polym. Sci., 16, (1971), 783-789.

ANALYTICAL SOLUTION OF THE NON UNIFORM HEAT EXCHANGE IN A REACTOR COOLING COIL WITH TIME VARYING FLUID FLOW

Ph. Bogaerts [1], **J. Castillo** [2] **and R. Hanus** [1]
Université Libre de Bruxelles
50 Av. F.-D. Roosevelt c.p.165-1050 Brussels (Belgium)
[1] Control Engineering Department (e-mail : pbogaert@labauto.ulb.ac.be)
[2] Chemical Engineering Department (e-mail : jcastill@labauto.ulb.ac.be)

Abstract. The analytical solution of the partial differential equation modeling the enthalpy balance of the cooling fluid in a batch exothermic reactor has been developed in the general case of a time varying flow. Due to this time dependance of the flow, the obtained solution is implicit. However, the solution can be derived explicitely in the case of a flow profile which can be analytically described. The interest of such a solution is illustrated with simulations aiming at comparing this true solution with the approximate segmentation and zeroth order methods. These simulations are performed in the case of a an exponential time decreasing flow.

1. Introduction

This paper is the second part of a study presented in [2]. Together with other papers (for instance [4]), it aims at proving the gains which can be obtained by trying to solve analytically a partial differential problem, and this as far as possible before using approximate numerical methods. The example proposed in this study consists of the mathematical model of a convective heat transfer between a uniform reacting mixture and the non uniform cooling fluid of a reactor internal coil. The non uniformity of the cooling fluid temperature leads to a distributed parameter problem which has to be solved analytically [2] and / or numerically [8]. In [2], we showed that the analytical solution of the enthalpy balance mentioned above could be obtained explicitely in the case of a constant fluid flow. As only the integral on position of this result (the cooling fluid temperature) is needed in the enthalpy balance of the reacting mixture, one is able to simulate the process with one independant variable (time) and without any approximation on the profile of the cooling fluid temperature. This solution was then used to study the accuracy which is reached with the approximate segmentation method. Actually, it must be kept in mind that a small number of finite volumes can lead to inaccurate results, although it is of common use [3, 6, 7, 9].

In the more general case of a time varying flow, another approximate solution, based on the previous rigorous result in the constant flow case, can also be used. This method, called zeroth order approximation, consists simply in allowing the flow to vary with time in the final solution of the constant flow case [2].

In this paper, the exact solution for the time varying flow case is presented and the obtained result is implicit. An explicit solution can however be obtained when the flow profile can be described analytically. This kind of result can then be used as the reference true solution with which approximate methods (like the zeroth order one) can be compared in this time varying flow case.

The second section briefly presents the process and its mathematical model. The partial differential equation, describing the cooling fluid enthalpy balance, is solved in the third section leading to an implicit solution. The explicit solution is derived from the latter one in the case of an exponential time decreasing fluid profile. The fourth section is devoted to an example of simulation in the case of the exponential fluid profile, together with some comparisons to the results obtained with the approximate methods of segmentation and of zeroth order. Conclusions are discussed in section five. The notations and numerical values for the simulations are given in appendix.

2. Description and mathematical model of the process

We consider here a batch reactor involving an exothermic first order reaction with two reactants (A and B) and one product (C). The reaction takes place in liquid phase and with constant volume. It is supposed that the mixture inside the reactor is instantaneously and perfectly mixed. A cooling coil is placed inside the reactor and contains cold water. The flow of this latter can be modulated with the valve aperture. It is assumed that this coil is always full of water, stagnant during a non cooling phase and circulating during a cooling one. More details about the mathematical models for this kind of reactor are given in [1] and [3]. The mathematical model, which will be used in the simulations of the next sections, is given hereafter.

Reaction rate (Arrhenius' law)

$$r = k e^{-\frac{E}{RT_m}} C_A^{0.5} C_B^{0.5} \tag{1}$$

See appendix for the significance of the symbols.

Mass balances

$$\frac{dC_i(t)}{dt} = \nu_i r \text{ where i = A, B, C} \tag{2}$$

Reacting mixture energy balance

$$\rho_m V_m c_{p_m} \frac{dT_m(t)}{dt} = U\pi D \int_0^{\ell} \left(T(x,t) - T_m(t)\right) dx - V_m r \Delta H_r \tag{3}$$

(accumulation of heat in the mixture = heat exchanged with the water of the cooling coil + heat produced by the reaction)

Cooling water energy balance

$$\rho c_p \frac{\pi D^2}{4} \frac{\partial T(x,t)}{\partial t} = U\pi D\left(T_m(t) - T(t)\right) - \rho c_p q(t) \frac{\partial T(x,t)}{\partial x} \tag{4}$$

(accumulation of heat in an infinitely small volume of water = heat exchanged with the uniform reacting mixture + enthalpy difference between the input and the output of the infinitely small volume)

This may be rewritten as

$$\alpha(t) \frac{\partial T(x,t)}{\partial x} + \beta \frac{\partial T(x,t)}{\partial t} + T(x,t) = T_m(t) \tag{5}$$

where

$$\begin{cases} \alpha(t) = \dfrac{\rho c_p q(t)}{\pi D U} \\ \beta = \dfrac{D \rho c_p}{4U} \end{cases} \tag{6}$$

This latter partial differential equation is analytically solved in the next section.

Note that the system of equations {(3), (4)} has to be replaced by the system {(7), (8)} given below in the case of a N finite volumes model.

$$\rho_m V_m c_{p_m} \frac{dT_m}{dt} = U \frac{S}{N} \sum_{i=1}^{N} \left(T_{fv_i} - T_m\right) - V_m r \Delta H_r \tag{7}$$

$$\rho \frac{V}{N} c_p \frac{dT_{fv_i}}{dt} = U \frac{S}{N}\left(T_m - T_{fv_i}\right) + \rho c_p q(t)\left(T_{fv_{i-1}} - T_{fv_i}\right) \tag{8}$$

where i = 1 to N and $T_{fv_0} = T_{IN}$ is the temperature at the input of the coil.

3. General solution of the partial differential equation

Using the bilateral Laplace transform (see [5]), the partial differential equation (5) can be reduced to an ordinary differential equation (with the time as independant variable). After solving this ODE, the inverse bilateral Laplace transform leads to the final solution given below :

$$\begin{aligned} T(x,t)\upsilon(x)\upsilon(t) = {} & e^{-\frac{t-\tau^*}{\beta}} \upsilon(t-\tau^*) T(0,\tau^*) \upsilon(\tau^*) + e^{-\frac{t}{\beta}} \upsilon(t) T\Big(x - \frac{1}{\beta}\int_0^t \alpha(\tau)d\tau, 0\Big) \upsilon\Big(x - \frac{1}{\beta}\int_0^t \alpha(\tau)d\tau\Big) \\ & + \frac{e^{-\frac{t}{\beta}} \upsilon(t-\tau^*)}{\beta} \int_{\tau^*}^{t} e^{\frac{\tau}{\beta}} T_m(\tau) \upsilon(\tau) d\tau \end{aligned} \tag{9}$$

where $\alpha(t)$ and β are given by (6), $\upsilon(.)$ are Heaviside functions $\left(\text{i.e. } \upsilon(x) = \begin{cases} 0 & \text{if } x<0 \\ 1 & \text{if } x>0 \end{cases}\right)$ and τ^* is implicitely defined by

$$\int_{\tau^*}^{t} \alpha(\tau)d\tau = \beta x \tag{10}$$

An explicit expression of τ^* can be obtained if the analytic function $q(t)$, and hence $\alpha(t)$, is known. For instance, in the case of a constant fluid flow $q(t) = q$, (10) leads to

$$\tau^* = t - \frac{\beta}{\alpha}x \tag{11}$$

which can be injected in (9), giving the same result as the one obtained in [2]:

$$\begin{aligned} T(x,t)\upsilon(x)\upsilon(t) = {} & e^{-\frac{x}{\alpha}}\upsilon(x)T(0,t-\frac{\beta}{\alpha}x)\upsilon(t-\frac{\beta}{\alpha}x) + e^{-\frac{t}{\beta}}\upsilon(t)T(x-\frac{\alpha}{\beta}t,0)\upsilon(x-\frac{\alpha}{\beta}t) \\ & + \frac{e^{-\frac{t}{\beta}}\upsilon(x)}{\beta}\int_{t-\frac{\beta}{\alpha}x}^{t} e^{\frac{\tau}{\beta}}T_m(\tau)\upsilon(\tau)d\tau \end{aligned} \tag{12}$$

In the case of an exponential time decreasing fluid flow given by

$$q(t) = q_{MAX}e^{-\gamma t} \tag{13}$$

equation (10) leads to

$$\tau^* = t - \frac{1}{\gamma}\ln\left(1 + \frac{\beta\gamma x e^{\gamma t}}{\alpha_{MAX}}\right) \tag{14}$$

where

$$\alpha_{MAX} = \frac{\rho c_p q_{MAX}}{\pi D U} \tag{15}$$

4. Example of simulation and comparison with approximate methods

In this section, a simulation of the reactor described in section 2 will be presented in the case of an exponential time decreasing flow of the form (13) with $q_{MAX} = 0.0018 m^3 s^{-1}$ and $\gamma = 10^{-4} s^{-1}$.

The result {(9),(14)} will be used in the enthalpy balance (3). At time t=0, the reactor is full of the reactants mixture and the temperature initial conditions are $T_m(0) = T(x,t)|_{t=0} = T_0 \quad \forall x$. Moreover, for any $t \geq 0$, the limit condition is $T(x,t)|_{x=0} = T_{IN}$.

As it can be seen in (3), the value of the integral $\int_0^{\ell} T(x,t)dx$ is required in order to compute the temperature $T_m(t)$.

This integral, in the case of an exponential time decreasing flow (13), is given below :

$$\begin{aligned} \int_0^{\ell} T(x,t)dx = {} & T_0 e^{-\frac{t}{\beta}}\upsilon(t)\left(\ell - MIN\left(\ell, \frac{1}{\beta}\int_0^t \alpha(\tau)d\tau\right)\right) + T_{IN}\frac{\alpha_{MAX}}{\beta\gamma - 1}e^{-\gamma t}\left(\left(MIN\left(1 + \frac{\beta\gamma e^{\gamma t}}{\alpha_{MAX}}\ell, e^{\gamma t}\right)\right)^{1-\frac{1}{\beta\gamma}} - 1\right) \\ & + \frac{e^{-\frac{t}{\beta}}}{\beta}\left(\ell F(t) + \frac{\alpha_{MAX}}{\beta}\left(G\left(t - \frac{1}{\gamma}\ln\left(1 + \frac{\beta\gamma e^{\gamma t}}{\alpha_{MAX}}\ell\right)\right) - G(t)\right)\right) \end{aligned} \tag{16}$$

where

$$F(t) = \int_0^t T_m(\tau)e^{\frac{\tau}{\beta}}d\tau \tag{17}$$

and

$$G(t) = \int_0^t F(\tau)e^{-\gamma\tau}d\tau \tag{18}$$

The simulation results are presented in the next page figures.

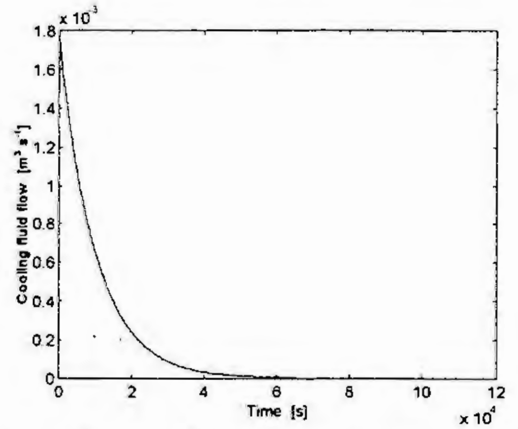


Fig. 1. Exponential time decreasing profile of the cooling fluid flow

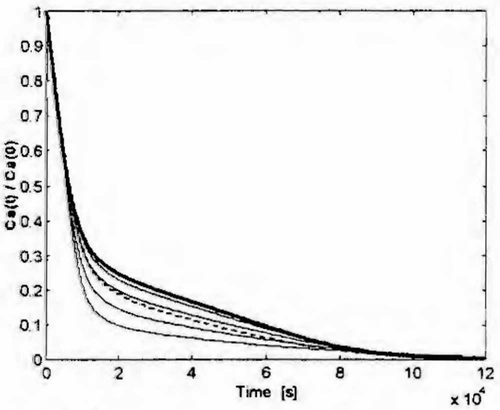


Fig. 2. Remaining proportion of reactant A (solid lines from top to bottom : analytical ("true") solution, 100, 36, 18 and 12 finite volumes; dashed line : zeroth order approximation method)

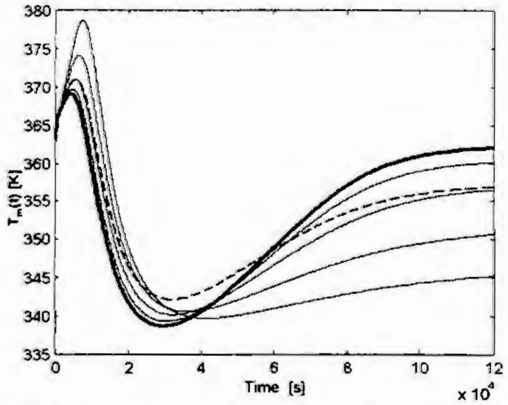


Fig. 3. Temperature in the reacting mixture (solid lines from bottom to top in the left part or from top to bottom in the right part : analytical ("true") solution, 100, 36, 18 and 12 finite volumes; dashed line: zeroth order approximation method)

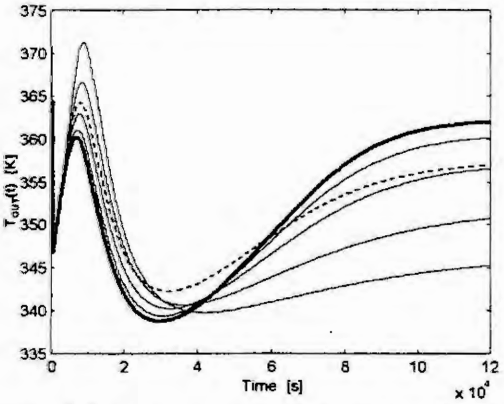


Fig. 4. Output temperature of the cooling fluid (solid lines from bottom to top in the left part or from top to bottom in the right part : analytical ("true") solution, 100, 36, 18 and 12 finite volumes; dashed ligne : zeroth order approximation method)

All the simulations are performed with the exponential time decreasing cooling flow presented in Figure 1. These simulations allow to compare the results obtained with different techniques, namely the rigorous analytical solution, the segmentation method (with various numbers of finite volumes) and the zeroth order approximation method. The comparisons are made on the remaining proportion of reactant A (Figure 2), on the temperature of the reacting mixture (Figure 3) and on the output temperature of the cooling fluid (Figure 4). It can easily be seen that a high number of finite volumes is required in order to reach a good accuracy. For instance, even in the case of 100 finite volumes, there exists a time interval (from about 2000 s to 4000 s) where the error on the residual proportion of reactant A is close to about 2 %. It must also be pointed out that no results are presented with a number of finite volumes less than 12. The reason is that the simulation error is so big that the output temperature of the cooling fluid exceeds the water boiling point. The model proposed in section 2 is then not valid anymore.

The zeroth order approximation method consists in allowing the flow to vary with time in the solution (12) of the constant fluid flow case. This method behaves slightly better than the segmentation method involving 18 finite volumes. However, it is interesting to note (see Figures 3 and 4) that the results are less satisfactory in the time interval where the various numbers of finite volumes lead to the lowest errors. Last but not least, the zeroth order approximation method has the advantage to require a lower computational load than the 18 finite volumes method.

5. Conclusions

The main result of this study consists of the analytical solution of the partial differential equation describing the enthalpy balance of the non uniform cooling fluid circulating in the internal coil within an exothermic uniform batch reactor. The general case of a time varying flow has been handled and the obtained solution is implicit. However this latter can be derived explicitely if the time varying flow can be described analytically. Examples of a constant flow and of an exponential time decreasing flow are proposed.

Such an explicit solution can be used to simulate the process with only one independant variable (time) and without any approximation on the position dependance. Moreover it can serve as the "true" reference in comparisons with approximate methods like the segmentation into finite volumes and the zeroth order method. This latter consists in the exact solution of the constant fluid flow case in which this constant flow is replaced by a time varying one. In the paper, this comparison is made in the case of an exponential time decreasing flow. The inaccuracy which results from using too low numbers of finite volumes is clearly evidenced. Concerning the zeroth order approximation method, more or less satisfactory results are obtained. They are similar with the ones corresponding to the segmentation into 18 finite volumes but with a lower computational load.

Obviously, it must be pointed out that the way to handle the partial differential equation could also be used in the caseof other physical systems which can be described with the same mathematical structure. For instance, a similar first order hyperbolic homogenous PDE allows to describe the mass balance of a reactant A in a plug flow reactor involving a first order reaction $A + B \rightarrow C$:

$$\frac{\partial C_A(x,t)}{\partial t} + \mathrm{v}(t)\frac{\partial C_A(x,t)}{\partial x} + k\,C_A(x,t) = 0$$

where $\mathrm{v}(t)$ is the axial fluid velocity. This enlarges the application field of the solution and conclusions proposed in this study.

Appendix : Notations (and numerical values for the simulations)

c_p : specific heat of the cooling water ($4184\ J\ kg^{-1}\ K^{-1}$)

c_{p_m} : specific heat of the reacting mixture ($2092\ J\ kg^{-1}\ K^{-1}$)

C_i : concentration of reactant (i=A or B) or product (i=C) ($mol\ m^{-3}$)

D : diameter of the coil ($0.104\ m$)

E : activation energy ($62760\ J\ mol^{-1}$)

k : frequency factor ($90000\ s^{-1}$)

ℓ : length of the coil ($90.8\ m$)

N : number of finite volumes

q : flow of the cooling water ($m^3\ s^{-1}$)

r : rate of reaction ($mol\ m^{-3}\ s^{-1}$)

R : constant of the perfect gases ($8.3143\ J\ mol^{-1}\ K^{-1}$)

t : time (s)

T : temperature of the cooling water (K)

T_{fv_i} : temperature of the cooling water in the i-th finite volume (K)

T_{IN} : temperature of the cooling water at the input of the coil (K)

T_m : temperature of the reacting mixture (K)

T_{OUT} : temperature of the cooling water at the output of the coil (K)

T_0 : temperature of the cooling water and the reacting mixture at t=0 (K)

U : global coefficient of convective heat exchange ($285\ J\ s^{-1}\ m^{-2}\ K^{-1}$)

V_m : volume of the reacting mixture ($10\ m^3$)

x : position (m)

ΔH_r : enthalpy of the reaction ($-104500\ J\ mol^{-1}$)

ν_i : stoichiometric coefficient ($\nu_A = \nu_B = -1$; $\nu_C = 1$)

ρ : specific mass of the cooling water ($1000\ kg\ m^{-3}$)

ρ_m : specific mass of the reacting mixture ($1000\ kg\ m^{-3}$)

References

1. Bogaerts, Ph., Cuvelier, A., Arte, Ph. and Hanus, R., Mathematical modelling of a chemical semi-batch reactor. In: Proc. of the EUROSIM' 95 Congress, Vienna, Elsevier Science B.V., 1995, 457-462.

2. Bogaerts, Ph., Castillo, J. and Hanus, R., Analytical solution of the non uniform heat exchange in a reactor cooling coil with constant fluid flow. Accepted for publication in: Mathematics and Computers in Simulation.

3. Cabassud, M., Le Lann, M.-V., Ettedgui, B. and Casamatta, G., A general simulation model of batch chemical reactors for thermal control investigations. Chem. Eng. Technol., 17 (1994), 255-268.

4. Castillo, J. and Bogaerts, Ph., Analytical solution approaches of non uniform temperature profiles. Application to heat exchanger simulations. Accepted for publication in: Proc. of the IMACS 2nd Mathmod, Vienna, Elsevier, 1997.

5. Hanus, R. and Bogaerts, Ph., Introduction à l'Automatique -Vol. 1 -Systèmes continus. De Boeck & Larcier, Paris, Bruxelles, 1996.

6. Juba, M. R. and Hamer, J. W., Progress and challenges in batch process control. In: Proc. of the Third International Conference on Chemical Process Control, (Eds.: Morari, M. and McAvoy, T.J.) Cache Elsevier, 1986, 139-183.

7. Luyben, W. L., Process modeling, simulation and control for chemical engineers. McGraw-Hill, Chemical Engineering Series, 1990.

8. Maffezzoni, C. and Ferrarini, L., A characteristic ligne based method to build finite-dimensional models of heat exchangers. Mathematical Modelling of Systems, Vol. 1 no. 3 (1995), 141-166.

9. Szeifert, F., Chovan, T. and Nagy, L., Process dynamics and temperature control of fed-batch reactors. Computers chem. Engng., Vol. 19, Suppl., (1995), S447-S452.

OPTIMAL EXPERIMENTAL DESIGN FOR PRACTICAL IDENTIFICATION OF UNSTRUCTURED GROWTH MODELS

K.J. Versyck, J.E. Claes and J.F. Van Impe
Department of Food and Microbial Technology, Katholieke Universiteit Leuven
Kardinaal Mercierlaan 92, B-3001 Heverlee (Belgium) Tel.: +32-16-32.15.85 Fax.: +32-16-32.19.97
E-mail: karina.versyck@agr.kuleuven.ac.be, jan.vanimpe@agr.kuleuven.ac.be

Abstract. In this paper, *optimal experimental design* for *parameter estimation* of unstructured microbial growth models during growth of biomass on a single limiting substrate in a fed-batch bioreactor is considered. The ratio of the largest to the smallest eigenvalue of the *Fisher information matrix* (i.e., the *modified E-criterion for optimal experimental design*) is used to evaluate the information content of several simulation fed-batch experiments, each with a different volumetric feed rate profile. The construction of optimal feed rate profiles is based on the following Conjecture: *A feed rate strategy which is optimal in the sense of process performance, is an excellent starting point for feed rate optimization with respect to estimation of those parameters with a large influence upon process performance.* The optimal value of 1 for the modified E-criterion is obtained for several feed rate profiles with different structures, after optimization of their corresponding degrees of freedom with respect to the information content of the experiment. For each profile a criterion evaluating the violation of *model validity* is calculated.

Introduction

Modeling of bioprocesses is based on the knowledge of mass balances, transport phenomena and reaction kinetics obtained by profound microbiological/biochemical studies. After the selection of an appropriate model structure, one encounters the issue of finding a unique set of corresponding model parameters (which have to be estimated from experimental data). In this case study, this step of the modeling procedure, known as the *parameter identification step*, is considered. Even if the *theoretical identifiability* of the parameters can be proved, very often the associated confidence space as calculated during experimental data fitting is in practice quite large (problem of *uniqueness*). For example, studies on the practical identifiability of the parameters in the Monod growth model showed that the parameters cannot be uniquely identified from noisy batch measurements [5], although they are theoretically identifiable from batch experiments [2]. Important improvements in parameter confidences are achieved using *optimal experimental design* techniques. Munack proved that the extension of the batch experiment by a fed-batch phase with time-varying feed rate leads to a higher accuracy of the parameter estimators [3],[4].

In this paper, we focus on the identifiability of the parameters of two unstructured models for growth kinetics: the non-monotonic *Haldane* kinetics and the monotonic *Monod* kinetics. A unique identification of the parameter set of a model is only possible if the available data are rich enough. The input must be designed in order to induce as much information as possible in the resulting state trajectories. In this case study, our goal is the design of a persistently exciting (time varying) feed rate profile for a *fed-batch bioreactor*. Optimal experimental design is applied for the *practical identification* (or *parameter estimation*) of couples of parameters (μ_m,K_p), (μ_m,K_i) and (K_p,K_i) for the Haldane kinetics. The same has been done for the parameters (μ_{max},K_S) of the Monod kinetics.

The paper is organized as follows. First, a general **Mathematical model** for fed-batch fermentations and the two models for the growth kinetics under study are presented. Then we provide a short introduction to the **Theoretical framework for optimal experimental design**. In the Section **Model validity** a criterion used for evaluation of the model validity is presented. The **Construction of profiles** and the corresponding **Results** for the practical identification of the Haldane and the Monod parameters are summarized in the subsequent section. The **Conclusions** are stated in the final section.

Mathematical model

We consider the identification of the kinetic parameters for the growth of a biomass X on one limiting substrate S. Biotechnological processes in a stirred tank reactor operated in fed-batch are generally described by:

$$\begin{aligned}\frac{dC_S}{dt} &= -\sigma\, C_X + \frac{u}{V}\,(C_{S,in} - C_S)\\ \frac{dC_X}{dt} &= \mu\, C_X - \frac{u}{V}\, C_X\\ \frac{dV}{dt} &= u\end{aligned} \qquad (1)$$

with C_S [g/L] the concentration of substrate and C_X [g DW/L] the biomass concentration, V [L] the volume of the liquid phase, $C_{S,in}$ [g/L] the substrate concentration in the volumetric feed rate u [L/h], σ [g/g DW h] the (overall) specific substrate consumption rate and μ [1/h] the (overall) specific growth rate.

The two specific rates σ and μ are interrelated by the following *linear law*:

$$\sigma = \frac{1}{Y_{X/S}}\mu + m \qquad (2)$$

with $Y_{X/S}$ [g DW/g] the biomass on substrate yield coefficient and m [g/g DW h] the (overall) specific maintenance demand. We assume a maintenance (exogenous) metabolism. We consider kinetic models in which the specific growth rate μ is a function of the substrate concentration C_S only.

- *(Non-monotonic) Haldane kinetics*:

$$\mu = \mu_m \frac{C_S}{K_p + C_S + \frac{C_S^2}{K_i}} \qquad (3)$$

  The parameter K_p [g/L] indicates how fast the optimum for the specific growth rate μ is reached. K_i [g/L] is the inhibition parameter.

- *(Monotonic) Monod kinetics*:

$$\mu = \mu_{max} \frac{C_S}{K_S + C_S} \qquad (4)$$

  with μ_{max} [1/h] the maximum specific growth rate and K_S [g/L] the so-called Monod constant.

During simulations, the following *nominal* parameter set has been used. For the Haldane kinetics: $\mu_m = 2.1$ 1/h, $K_p = 10$ g/L, and $K_i = 0.1$ g/L. For the Monod kinetics: $\mu_{max} = 0.1$ 1/h and $K_S = 1$ g/L. Other parameters are: $Y_{X/S} = 0.47$ g DW/g, $m = 0.29$ g/g DW h, $C_{S,in} = 500$ g/L. Further, the following operational and initial conditions have been used: $X(0) = 10.5$ g DW, $V_* = 7$ L, $V_{MAX} = 10$ L, which corresponds to $\alpha = 1500$ g (the total amount of supplied substrate). The covariances of the measurement errors of the substrate concentration C_S and the biomass concentration C_X are given by: $\sigma^2_{C_S} = 1\ \ 10^{-2}$ g^2/L^2 and $\sigma^2_{C_X} = 6.25\ \ 10^{-4}$ g^2/L^2 respectively.

Theoretical framework for optimal experimental design

Parameter estimation can be formulated as minimization of the following *identification functional* $\mathcal{J}_I$ by optimal choice of the parameter vector $\mathbf{p}$:

$$\mathcal{J}_I \triangleq \int_0^{t_f} (\mathbf{y}(\mathbf{p}) - \mathbf{y}_m)^T \mathbf{Q} (\mathbf{y}(\mathbf{p}) - \mathbf{y}_m) dt \qquad (5)$$

in which $\mathbf{y}_m$ is the vector of measured outputs, $\mathbf{y}(\mathbf{p})$ is the vector of model predictions by using the parameter vector $\mathbf{p}$, and $\mathbf{Q}$ is a user supplied square weighting matrix. For optimal identification purposes, an experiment with maximum information content should be performed. Such experiment can be designed by using the concepts and tools of optimal experimental design. The question encountered here, can be formulated as follows.

> *Which feed rate profile must be applied to obtain the maximum information out of the resulting trajectories?*

To analyze the information content of the state trajectories obtained in a certain experiment, the *Fisher information matrix* $\mathcal{F}$ can be used (see, e.g., [4]):

$$\mathcal{F} \triangleq \int_0^{t_f} \left(\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\right)^T \mathbf{Q} \left(\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\right) dt \tag{6}$$

Q is normally chosen as the inverse of the measurement error covariance matrix. Depending on the requirements imposed by the application, a specific scalar function of this Fisher information matrix is used as the performance index for optimal experimental design to increase the parameter identifiability. Different so-called *optimal design criteria* are discussed in literature [9]. In order to increase parameter identifiability, the following scalar cost function of the Fisher information matrix can be used:

$$J[u] = \Lambda(\mathcal{F}) = \frac{\lambda_{max}(\mathcal{F})}{\lambda_{min}(\mathcal{F})} \tag{7}$$

which is the so-called *modified E-criterion* for optimal experimental design (with λ an eigenvalue of $\mathcal{F}$). The ratio of the largest to the smallest eigenvalue of $\mathcal{F}$ (the condition number of $\mathcal{F}$) should be as near as possible to 1. This corresponds to circular lines of constant functional values and a cone-like functional shape of $\mathcal{J}_\mathcal{I}$ in the parameter space. For a detailed discussion reference is made to, e.g., [4].

Model validity

A unique identification of the parameter set of a model is only possible if the available data are sufficiently rich. The input must be designed in order to induce as much information as possible in the resulting state trajectories (*persistent excitation* of the system). However, if the feed rate exhibits enormous gradients it is questionable whether the unstructured growth models are still valid. An essential requirement for the validity of the model is *balanced growth*, a biological state during which the intracellular metabolic reaction network is operating in steady-state conditions. In [1] the deviation from steady-state conditions is quantified –similar to an output least-squares error functional– as:

$$\Phi(\mu) = \int_0^{t_f} \left(\frac{d\mu}{dt}\right)^2 dt \tag{8}$$

- For the non-monotonic Haldane kinetics the time derivative of the specific growth rate μ (3) can be written, by using the substrate balance (1), as follows:

$$\frac{d\mu}{dt} = \frac{\mu_m}{(K_p + C_S + \frac{C_S^2}{K_i})^2} (K_p - \frac{C_S^2}{K_i}) \cdot (-\,\sigma\, C_X \;+\; \frac{u}{V}\,(C_{S,in} - C_S))$$

- For the monotonic Monod kinetics (4), and by using (1), we find:

$$\frac{d\mu}{dt} = \frac{\mu_{max}\, K_S}{(K_S + C_S)^2} \cdot (-\,\sigma\, C_X \;+\; \frac{u}{V}\,(C_{S,in} - C_S))$$

If the aim is to optimize the time-varying feeding strategy, $u(t)$, with respect to improving the quality of parameter estimation while violating model validity as less as possible, an *extended* criterion for optimal experimental design for unstructured growth models can be created by combination of the modified E-criterion $\Lambda(\mathcal{F})$ (7) with the model validity functional $\Phi(\mu)$ (8):

$$J[u] \;=\; \Lambda(\mathcal{F}) \;+\; \mathcal{K}\;\Phi(\mu) \tag{9}$$

where $\mathcal{K}$ denotes a weighting factor penalizing violations of model validity. The minimization of this new criterion is out of the scope of this paper, but the simple model validity criterion (8) is calculated and discussed for the feed rate profiles presented in this paper.

Construction of profiles and results

In a preliminary research the modified E-criterion (7) has been calculated for the following three profiles. Each profile (with exception of the batch process) consists of two phases: a feeding phase (from time $t = 0$ to $t = t_2$) and a batch phase (from time $t = t_2$ to $t = t_f$). When the total amount of substrate has been supplied (at time $t = t_2$), the fermentation continues in batch mode until the stop criterion $dX/dt = 0$ is fulfilled at time $t = t_f$, this is when the specific growth rate μ, given by (3) or (4), and thus the substrate concentration C_S has become equal to zero. This stop criterion follows from the application of the Minimum Principle for optimization of the biomass yield [6].

1. Batch process: $u(t) = 0$ L/h. The total amount of substrate available, i.e., α [g] is supplied at time $t = 0$ h. This feed rate profile has no degrees of freedom.
2. Constant feed rate: $u(t) = u^*$ L/h with u^* a constant number and the initial substrate amount equal to $S(0) = 0$ g. Since the total amount of substrate available is a fixed value the following constraint must hold:
$$C_{S,in} \cdot u^* \cdot t_2 \;=\; \alpha$$
with t_2 the final time of the feeding phase. Hence, there is one degree of freedom (DOF): if $u(t)$ equals an arbitrarily chosen number u^*, the final time t_2 of the feeding phase is fixed. The optimal value u^*_{opt} is obtained by *parametric optimization* of this profile in the sense of maximum information content.
3. Constant substrate concentration: Feed rate that keeps the substrate concentration in the reactor constant, described by
$$u(t) = \frac{\sigma\, X}{C_{S,in} \;-\; C_S^*} \quad \text{with} \quad C_S(0) = C_S^* \tag{10}$$
which can be calculated from (1) and (2). It has one degree of freedom, namely C_S^*. Observe that this profile is the optimal control in the sense of the biomass yield for non-monotonic growth kinetics if C_S^* is selected equal to $C_{S,\mu} = \sqrt{K_p K_i}$ which maximizes μ [6]. By applying parametric optimization of this profile in the sense of maximum information content the optimal value $C^*_{S,opt}$ is obtained.

Table 1 (Rows 1 to 3) summarizes the values of the modified E-criterion $\Lambda(\mathcal{F})$ (Column 5) for the estimation of (μ_m,K_i) corresponding to these three profiles. The degrees of freedom (if there are any) of the profiles are optimized with respect to $\Lambda(\mathcal{F})$.

Optimal feed rate profiles for parameter estimation. Based on the assumption that the feed rate profiles obtained for optimal process performance are exciting those features with large influence on the performance, the following Conjecture can be formulated for the construction of the optimal parameter estimation profile.

> *A feed rate strategy which is optimal in the sense of process performance, is an excellent starting point for feed rate optimization with respect to estimation of those parameters with large influence upon process performance* [6].

In [8] theoretical evidence is given for the fact that profile #3 –which is optimal for process performance if $C_S^* = C_{S,\mu} \equiv \sqrt{K_p K_i}$– yields no satisfactory results with respect to practical identification. Although, inspired by the preceeding Conjecture and the proof given in [8], it is possible to propose a suitable adjustment of the feed rate profile #3 to circumvent the lack of information induced by applying this profile. By chosing the initial substrate concentration different from the constant concentration C_S^* during the feeding phase –with feed rate $u(t)$ described by (10)– we obtain one degree of freedom more –compared to profile #3– namely, the initial substrate concentration $C_S^*(0)$. This implies the addition of an extra phase (from time $t = 0$ to $t = t_1$), which preceeds the feeding phase (from time $t = t_1$ to $t = t_2$). When $C_S(0)$ is chosen below C_S^*, the substrate is supplied –in the first phase– at the maximum possible feed rate u_{MAX}, i.e., the maximum pump capacity (during simulations, $u_{MAX} = 1$ L/h is used). When $C_S(0)$ is higher than C_S^* of the second phase, the first phase is a batch phase. As for the previous profiles, parametric optimization has been applied for the two degrees of freedom ($C_S^*(0)$ and C_S^*) in the sense of maximum information content (profile #4, Table 1). By applying this profile, which is clearly based on profile #3, the ratio of eigenvalues of the Fisher information matrix equals 1, which means that this latter feed rate profile is truly optimal with respect to the modified E-criterion for optimal experimental design. The identification functional $\mathcal{J}_I$ (5) is cone-like.

The violation of model validity is quantified by $\Phi(\mu)$ (Column 6, Table 1). In the context of model validity it is preferable to stop the fermentation as soon as the whole amount of available substrate has been supplied, i.e., when the volume reaches its maximum value V_{MAX} (profile #5), instead of continuing in batch to consume the remaining substrate (profile #4): during the last batch phase, the model validity functional $\Phi(\mu)$ increases from a very small order of magnitude $\mathcal{O}(10^{-4})$ 1/h^3 at time $t = t_2$ to $\mathcal{O}(10^{-1})$ 1/h^3 at time $t = t_f$, and this for all the parameter couples under study in this paper. This unnecessary violation of the model validity is caused by the fast decrease of the specific growth rate μ, during this very short final batch phase. Two alternative optimal feed rate profiles –profiles #6 and #7, with two and three degrees of freedom respectively– have been constructed based on profile #3. As is clear from the corresponding plots in Table 1, the variation in time of the substrate concentration $C_S(t)$ (from time $t = 0$ to $t = t_1$) is for these profiles induced by a feeding phase with constant feed rate level u^*. For the profiles #6 and #7 –with the same stop criterion as for profile #4, i.e., $dX/dt = 0$– the violation of the model validity is also larger than for profile #5.

We have obtained similar results for the practical identification of the parameter couples (K_p,K_i) and (μ_m,K_p) of the Haldane kinetics, and the parameter couple (μ_{max},K_S) of the Monod kinetics. The results are not shown here because of the limited space.

Conclusions

The information content of several experiments with different feed rate profiles is evaluated (by numerical simulations) for the estimation of the parameters of unstructured microbial growth models. Inspired by the Conjecture that feed rate profiles which are optimal with respect to process performance are a good starting point for optimization of feed rate profiles with respect to parameter estimation, some alternative feed rate profiles are constructed based on the optimal profile for process performance. After parametric optimization, several profiles are obtained which are optimal with respect to the modified E-criterion as quantified by its optimal value $\Lambda(\mathcal{F}) = 1$. For other feeding strategies, mentioned in literature, this optimal value has not been attained. Besides to the information criterion, attention is paid to the violation of the condition for model validity induced by each profile. To dispose of multiple solutions to the problem of optimal experimental design offers the opportunity to take also into account other considerations, such as practical feasibility of the optimal profiles obtained at simulation level.

Acknowledgment. Author K. Versyck is a research assistant with the Fund for Scientific Research - Flanders (FWO). Author J. Claes is a research assistant with the Flemish Institute for Scientific and Technological Research in Industry (IWT). Work supported in part by NFWO Projects G.0141.95 and G.0286.96, and KULeuven Project OT/95/20. The scientific responsibility is assumed by its authors.

References

1. Baltes, M., Schneider, R., Sturm, C. and Reuss, M., Optimal experimental design for parameter estimation in unstructured growth models. Biotechnol. Prog., 10 (1994), 480 - 488.
2. Holmberg, A., On the practical identifiability of microbial growth models incorporating Michaelis-Menten type nonlinearities. Math. Biosci., 62 (1982), 23 - 43.
3. Munack, A. and Posten, C., Design of optimal dynamical experiments for parameter estimation. In: Proc. American Control Conference ACC89, Pittsburgh (USA), 1989, 2011 - 2016.
4. Munack, A., Optimal feeding strategy for identification of Monod-type models by fed-batch experiments. In: Computer Applications in Fermentation Technology Modelling and Control of Biotechnological Processes, Elsevier, Amsterdam, 1989, 195 - 204.
5. Nihtilä, M. and Virkkunen, J., Practical identifiability of growth and substrate consumption models. Biotechnol. and Bioeng., 19 (1977), 1831 - 1850.
6. Van Impe, J., Modeling and optimal adaptive control of biotechnological processes. PhD Thesis, Department of Electrical Engineering, Katholieke Universiteit Leuven (Belgium), 1993.
7. Van Impe, J., Claes, J. and Bastin, G., Optimal feed rate profiles for combined bioprocess modeling and optimization. In: Proc. European Control Conference ECC95, Roma (Italy), 1995, 3510 - 3515.
8. Versyck, K., Claes, J. and Van Impe, J., Optimal feed rate profiles for parameter estimation of Haldane growth kinetics. In: Proc. Multiconference on Computational Engineering in Systems Applications CESA'96, Volume 1: Modelling, Analysis and Simulation, (Eds.: Borne, P., Soenen, R., Sallez, Y. and El Khattabi, S.) Lille (France), 1996, 136 - 141.
9. Walter, E. and Pronzato, L., Qualitative and quantitative experiment design for phenomenological models - A survey. Automatica, 26(2) (1990), 195 - 213.

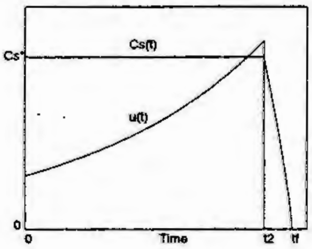
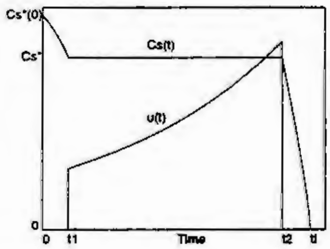
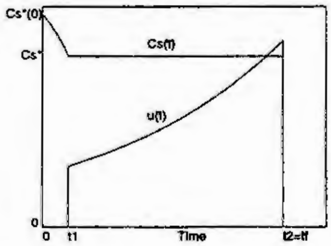
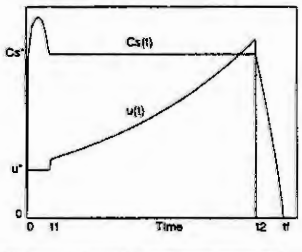
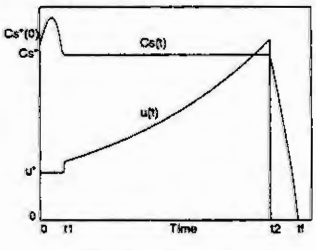
<table>
<tr><th></th><th></th><th colspan="5">Haldane (μ_m,K_i)</th></tr>
<tr><th>#</th><th>profile</th><th colspan="2">DOF</th><th>$\Lambda(\mathcal{F})$</th><th>$\Phi(\mu)$ [1/h^3]</th><th>t_f [h]</th></tr>
<tr><td>1</td><td><i>batch</i></td><td>-</td><td>-</td><td>1.87 10^8</td><td>2.4186 10^{-2}</td><td>3.2082 10^2</td></tr>
<tr><td>2</td><td>$u^*(t)$ <i>constant</i></td><td>u^*_{opt}</td><td>3.1253 10^{-2}</td><td>2.47 10^6</td><td>1.9805 10^{-1}</td><td>2.0789 10^2</td></tr>
<tr><td>3</td><td>
</td><td>$C^*_{S,opt}$</td><td>3.0874 10^1</td><td>4.55 10^7</td><td>3.3473 10^{-2}</td><td>2.0802 10^2</td></tr>
<tr><td>4</td><td>
</td><td>$C^*_{S,opt}(0)$
$C^*_{S,opt}$</td><td>3.7816
7.5594 10^{-1}</td><td>$\boxed{1.00}$</td><td>1.6683 10^{-1}</td><td>3.5808 10^1</td></tr>
<tr><td>5</td><td>
</td><td>$C^*_{S,opt}(0)$
$C^*_{S,opt}$</td><td>3.7749
7.5553 10^{-1}</td><td>$\boxed{1.00}$</td><td>8.6110 10^{-4}</td><td>3.5745 10^1</td></tr>
<tr><td>6</td><td>
</td><td>u^*_{opt}
$C^*_{S,opt}$</td><td>9.0821 10^{-3}
2.3004 10^{-1}</td><td>$\boxed{1.00}$</td><td>6.4188 10^{-2}</td><td>6.4590 10^1</td></tr>
<tr><td>7</td><td>
</td><td>$C^*_{S,opt}(0)$
u^*_{opt}
$C^*_{S,opt}$</td><td>4.0597 10^{-1}
1.1765 10^{-2}
5.0598 10^{-1}</td><td>$\boxed{1.00}$</td><td>1.4449 10^{-1}</td><td>4.0035 10^1</td></tr>
</table>

Table 1: $\Lambda(\mathcal{F})$, $\Phi(\mu)$, and t_f for different feeding strategies for identification of Haldane parameters (μ_m,K_i) (DOF: degrees of freedom, $C^*_{S,opt}$ [g/L], $C^*_{S,opt}(0)$ [g/L], u^*_{opt} [L/h])

DETERMINATION OF RENAL RESERVE CAPACITY BY IDENTIFICATION OF KINETIC SYSTEMS

W. Estelberger, S. Zitta, K. Stoschitzky, R. Zweiker, T. Lang, F. Mayer, and G. Reibnegger
Karl-Franzens-University Graz,
Harrachgasse 21, A-8010 Graz

Abstract. A computer-based method of system identification and parameter variance estimation for two-compartment models matched to dynamic marker concentration profiles for the determination of renal clearance is used for the investigation of changes in renal function due to dietary agents. For this purpose two single-shot experiments using sinistrin together with p-aminohippuric acid as (p-AH) clearance markers, and proteins as dietary factor at the beginning of the second experiment were performed. The procedure was applied to normal controls and moderately hypertensive patients.

The glomerular filtration rates (GFR) rise in normal test subjects under the influence of protein loads. In the patients, however, one group consisting of moderately hypertensive persons shows higher or at least the same GFR values, whereas another group comprising persons having suffered from severe hypertension for some time exhibits 'paradoxical' decreases of the glomerular filtration rates following the protein test meals.

The different types of clearance responses to the dietary loads are obviously correlated with long-term effects of increased blood pressures. The study demonstrates that accurate minimal modelling and system identification of kinetic experiments allows one to detect and understand such long-term pathophysiological causal relationships.

Introduction

The study of changes of glomerular filtration rates (GFR) and effective renal plasma flows (ERPF) in hypertensive and diabetic renal patients under the influence of protein-rich diets has revealed results which appear as 'paradoxical' insofar as, quite in contrast to normal organ behaviour, the GFR values frequently decrease, whereas the ERPF values in the same patients either increase or remain at least constant under protein dietary loads [1].

For such dynamic kidney function tests GFR estimates derived from creatinine measurements, by traditional steady-state methods or by kinetic slope-intercept methods are definitely inappropriate [2]. Either stationary creatinine levels do not show short-term changes of the clearances involved or marker amounts remaining in the extracellular space from the first kinetic experiment are not taken into account for the evaluation of the following experiment. Therefore a computer-based method for the evaluation of single-shot experiments has been employed which is based on a two-compartment model incorporating the initial conditions of the marker distribution and elimination processes involved [3,4]. This kinetic technique which utilizes the information in the temporal marker concentration profiles allows the determination of the estimates of the clearances and distribution volumes together with their accuracies for consecutive single-shot experiments with dietary protein loads in between.

Model Formulation

Figure 1 schematically depicts the two-compartment model assumed as underlying the organismic marker distribution and elimination processes involved in kinetic experiments studied. Therein the extracellular space in which the markers applied distribute is considered to be composed of two functionally separated spaces, a well perfused central volume and a less perfused peripheral compartment. The marker kinetics as represented by the temporal courses of the marker amounts in the two compartments is the result of the infusion strategy, the exchange transports between the two compartments, and finally the renal elimination process [5-8].

The model can be formulated by a set of two simultaneous differential equations describing the rates of change of the marker amounts in the two respective compartments:

$$dx_1/dt = f(t) - (k_{01} + k_{21})x_1 + k_{12}x_2 \qquad \text{(Eq. 1)}$$

$$dx_2/dt = k_{21}x_1 - k_{12}x_2 \qquad \text{(Eq. 2)}$$

Equations 1 and 2 can be stated verbally in the following way: Firstly, the rate of change of the marker amount in the central compartment, dx_1/dt, is determined by the input strategy chosen, the loss of marker from the central to the peripheral compartment, its gain by the central from the peripheral volume, and its elimination through the renal excretion mechanism. Secondly, the rate of change of the marker amount in the peripheral space, dx_2/dt, is due to gain from and loss to the central pool. These transport processes are assumed to be proportional to the marker amounts momentarily contained in the respective distribution volumes.

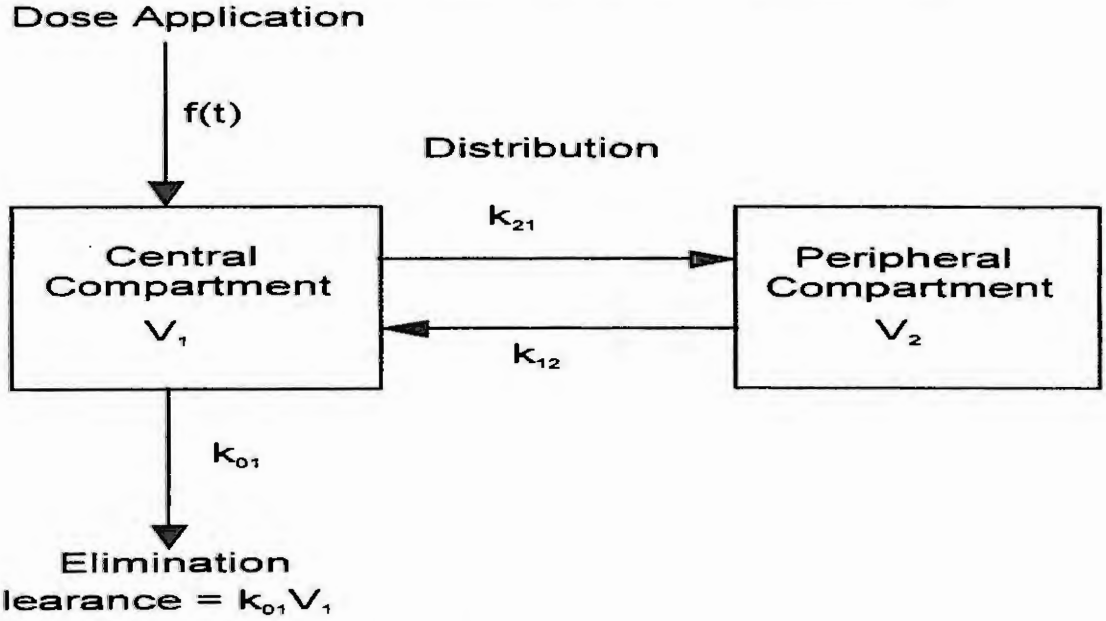


Fig. 1 Compartment Model Describing Marker Distribution and Elimination.

The input function of an experiment consisting of a bolus injection followed by constant infusion is given by Equations 3 and 4:

$f(t)=D/\tau$, if $0\leq t<\tau$ (Eq. 3)

$f(t)=\rho$, if $\tau\leq t<T_c$ (Eq. 4)

The initial marker amounts are given by

$x_1(0)=c_1(0)V_1=x_{10}$ (Eq. 5)

$x_2(0)=c_2(0)V_2=c_2(0)V_1(k_{21}/k_{12})=x_{20}$ (Eq. 6)

The fitting of the solution of the model defined by Equations 1 to 6 to the experimental plasma concentration data measured over a sufficiently long time horizon can be done by a method for the search of the minimum of a criterion of the sort:

$E=\Sigma(c_1(t_i)-c_{exp}(t_i))^2, (i=1\ldots n)$ (Eq. 7)

The identification of the model is most efficiently done with the Levenberg-Marquardt algorithm [9] allowing one to estimate the optimal values of the independent system parameters k_{01}, k_{21}, k_{12}, and V_1 as well as of dependent parameters such as V_2, the clearance $C_{INU}=k_{01}V_1$, the permeability time constant $t_{21}=\ln(2)/k_{21}$ etc.

Since there is always 'noise' in the experimental data consisting of random and systematic fluctuations around the ideal behavior of the system, the accuracy of the parameters has to be ascertained. This can be done by means of a Monte-Carlo technique for the generation of artificial protocols by superposition of Gaussian random numbers on the optimal trajectory. The random numbers are taken from a distribution with mean zero and a standard deviation given by the following expression [9]:

$s=(E/(n-4))^{½}$ (Eq. 8)

About 100 artificial protocols created in this way and themselves subjected to the identification procedure suffice for the parameter variance estimations. The resulting parameter constellations are evaluated statistically for the determination of the means of the parameters and their standard deviations. These standard deviations are equivalent to the standard errors of the parameters derived by means of the so-called Fisher's information matrix method [10]. But since this classical technique has as a necessary condition a Gaussian distribution of the residuals superposed to the solutions of strictly linear models, the computer-oriented procedure outlined is more universally applicable [11]. The exact solution of the model formally described by Equations 1 to 6 generalized to both single-injection and constant-infusion inputs is given by a superposition of the solution of the eigenvalue problem posed by the corresponding homogeneous system and the particular solution of the inhomogeneous problem which can be found by the method of undetermined coefficients [12].

The temporal profiles of the concentrations $c_1(t)$ and $c_2(t)$ in their respective compartments are defined by Eqs. 9 and 10:

$c_1(t) = x_1(t)/V_1$ (Eq. 9)

$c_2(t) = x_2(t)/V_2$ (Eq. 10)

The symbols in the expressions have the following meanings:

f(t) the input strategy as a function of time t,
x_1 the amount of the marker in the central compartment,
x_2 the amount of the marker in the peripheral compartment,
k_{21} the relative rate of transport from compartment 1 to 2,
k_{12} the relative rate of transport from compartment 2 to 1,
k_{01} the relative rate of elimination,
D the priming dose,
τ the injection duration,
ρ the infusion rate,
T_c the duration of the constant-infusion experiment,
V_1 the volume of the central compartment,
V_2 the volume of the peripheral compartment.

GFR is defined as clearance of sinistrin and ERPF is defined as clearance of p-AH.

Results

Figure 2 shows the temporal concentration profiles of sinistrin in a normal test person. It serves to exemplify the kind of kinetic experiments done by double bolus application of both sinistrin and p-AH in each of the subjects contained in Table 1 and in 3 healthy controls. Figures 3 and 4 show the values of GFR and ERPF in the healthy controls and in hypertensive patients. The respective adjacent bars constitute the magnitudes of the clearance values together with their respective error measures obtained in the consecutive single-shot experiments.

Table 1. Hypertensive Patients with Mean Arterial Pressures.

	Age [years]	**Sex**	**Duration of Hypertension [years]**	**MAP [mm Hg]**
DP	52	m	5	105
PO	56	m	25	103
LH	69	m	20	110
SM	64	f	5	108
KA	58	m	10	122
GF	52	f	10	120
WC	48	f	10	115
HM	64	f	35	123
HF	54	f	35	120
SE	54	f	5	115

$$MAP = \frac{SYST + 2xDIAST}{3}$$

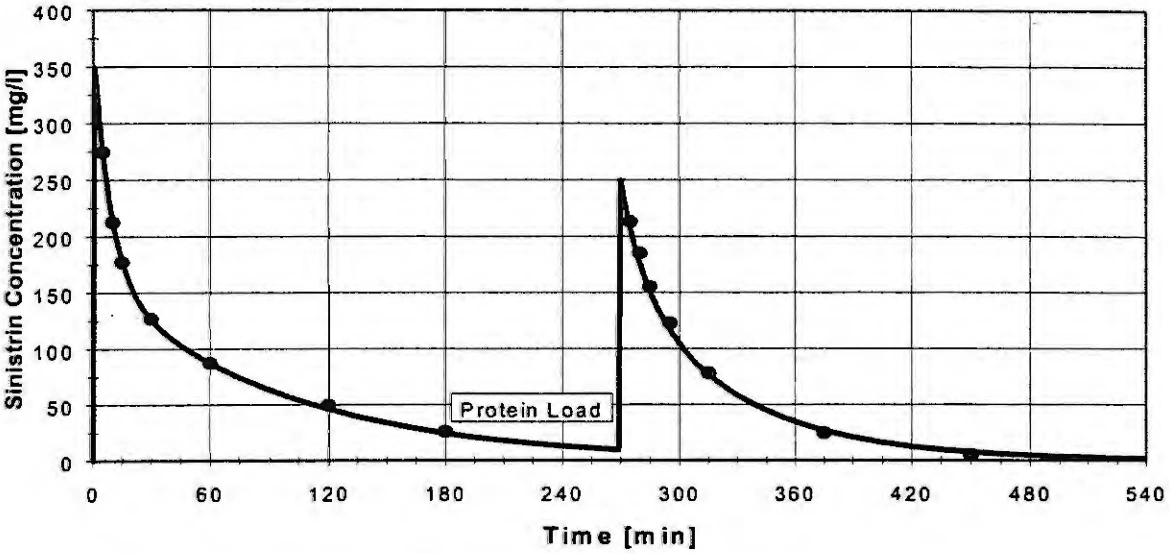


Fig. 2: Temporal concentration profiles of sinistrin in a normal test person.

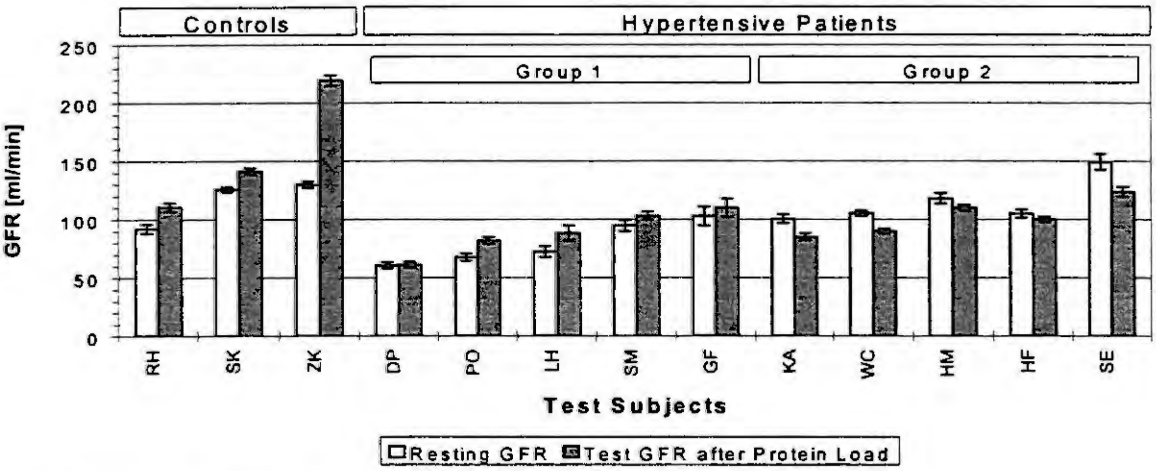


Fig. 3: GFR values in healthy controls and in hypertensive patients.

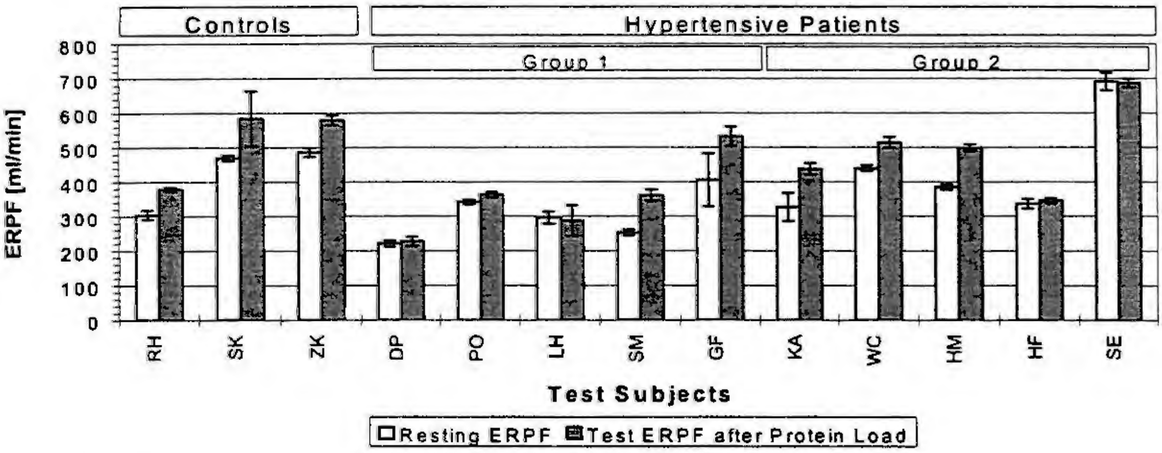


Fig. 4: ERPF values in healthy controls and in hypertensive patients.

Summary

As can be seen the glomerular filtration rates and ERPF values rise in normal test subjects under the influence of protein loads. As to be expected normal test subjects have renal reserve capacity. In the patients studied, however, there are two categories. In the hypertensive patients there is a group showing constancy of GFR and another one revealing decreases in GFR (Fig. 2). The respective ERPF values exhibit constancy or even increases, but no decreases (Fig. 3).

The patients of the two groups differ by their mean blood pressures and additionally by the durations of their hypertensive states. The moderately hypertensive patients with normal serum creatinine levels were investigated at the end of the second week of the washout phases of their standard antihypertensive medication (ACE inhibitors, beta blockers, and calcium antagonists). Thus, the observed decreases in GFR cannot be attributed to the medication applied, but rather reflect the stages of renal impairment as indicated by the GFR responses to the dietary loads correlated with the long-term effects of increased blood pressures.

Especially differences in the vascular resistances of the vasa afferentia and the vasa efferentia have previously been made responsible for the 'paradoxical' decreases of the glomerular filtration rates as observed in diabetic patients. Long-term developments towards hypertensive states might account for differential deteriorations of the glomerular vessels, since the vasa afferentia are exposed to higher pressure stresses than the vasa efferentia. This higher incidence of vascular lesions might lead to the decreases of the glomerular filtration rates observed in the hypertensive patients in hyperfiltration experiments.

Dynamic renal function tests of the kind presented obviously offer the possibility to judge the true state of the renal vasculature. In contrast, single resting state assessments are inadequate, since both increases and decreases of GFR estimates are found within the same range of basic glomerular filtration rates in both nearly normotensive and hypertensive patients. Therefore 'normal' resting GFR values alone are obviously not conclusively indicative of intact renal functional reserve, but may be caused by renal hyperfiltration. As demonstrated the identification of dynamic models describing the kinetics of suitable markers plays an essential role for the assessment of altered patterns of physiological responses to exogenous stimuli.

References

1. Bosch JP, Lew S, Glabman S, Lauer A. Renal Hemodynamic Changes in Humans. American Journal of Medicine, 81 (1986), 809-15.
2. Maschio G, Oldrizzi L, Rugiu C, De Biase V. Dynamic evaluation of renal function: a chimera for nephrologists? Journal of Nephrology, 3 (1989), 157-64.
3. Estelberger W, Petek W, Zitta S, Mauric A, Horn S, Holzer H, Pogglitsch H. Determination of the glomerular filtration rate by identification of sinistrin kinetics. European Journal of Clinical Chemistry and Clinical Biochemistry, 33 (1995), 201-9.
4. Estelberger W, Zitta S, Lang T, Mayer F, Mauric A, Horn S, Holzer H, Petek W, Reibnegger G. System identification of the low-dose kinetics of p-aminohippuric acid. European Journal of Clinical Chemistry and Clinical Biochemistry, 33 (1995), 847 - 53.
5. Estelberger W, Paletta B, Aktuna D, Petek W, Horn S, Pogglitsch H. Modelling and Identification of Tracer Kinetics in Kidney Function Diagnostics. In: Gál K, editor. MIE'91 Satellite Conference on Computer Modelling; 1991 August 24-24; Budapest. Budapest: John von Neuman Society for Computing Sciences 1991:119-127.
6. Estelberger W, Petek W, Pogglitsch H. Model-based determination of renal clearance from temporal venous plasma profiles of markers. In: Trappl R, editor. Cybern. Syst. Res. Vol. 2; 1992 April 21-24; Vienna. Singapore: World Scientific 1992:893-900.
7. Estelberger W, Petek W, Pogglitsch H. Simulation der sättigbaren und hemmbaren Kinetik renal-tubulär eliminierter Pharmaka. In: Boenick U., Schaldach M, editors. Biomedizinische Technik; 1992 September 17-19; Graz. Berlin: Schiele & Schön, 1992:37 Erg. 1:73-75.
8. Valkó P, Vajda S. Advanced scientific computing in BASIC with applications in chemistry, biology and pharmacology. Amsterdam: Elsevier, 1989:161-173.
9. McIntosh JEA, McIntosh RP. Mathematical Modelling and Computers in Endocrinology. Berlin: Springer, 1980:74-102.
10. Carson ER, Cobelli C, Finkelstein L. The Mathematical Modeling of Metabolic and Endocrine Systems. Model Formulation, Identification, and Validation. New York: Wiley, 1982:204-216.
11. Metzler CM. Statistical Properties of Estimates of Kinetic Parameters. In: Bozler G, van Rossum JM, editors. Pharmacokinetics during Drug Development: Data Analysis and Evaluation Techniques.Stuttgart: Fischer, 1982:138-143.
12. Kreyszig E. Adavanced Engineering Mathematics. New York: Wiley, 1993:186-188.

A NEW MODEL OF THE MICROPLASMA PULSATION IN P-N JUNCTION

B.Datsko[1], A.Demchuk[1], V.Gafiychuk[1], G.Ilchuk[2], Yo.Hromyak[2], V.Pavlysh[2]

[1] Institute of Applied Problems in Mechanics and Mathematics
NAS of Ukraine, 3 b Naukova str., Lviv 290603, Ukraine, e-mail: gaf@viva.lviv.ua
[2] Technical University Lviv,Bandera str. 12, Lviv 290000, Ukraine

Abstract. The main characteristics of the current flow in the form of microplasma impulses were confirmed by a numerical solution based on the proposed model, which is described by the nonlinear system of partial differential equations: the balance of the average (over the space charge region) electron density, continuity of the total current in the quasineutral n- or p-type region of structure and selfheating of lattice of $p-n$-structure. It is shown that microplasma can appear spontaneously when a local inhomogeneity is present in a space charge layer of $p-n$-structure, and increasing in the lattice temperature as a result of the Joule heating leads to the disappearing of microplasma. Numerical studies are reported of the kinetics of microplasma impulses, of their shape, length on the voltage drop across a structure and of parameters $p-n$-structure.

Introduction

When avalanche breakdown takes place in $p-n$-junction formation of microplasmas in the form of region of $\sim 1\mu m$ size with a very high density of the local avalanche current is usually observed even at low overall avalanche current density j. In the article [2] the authors have developed a theory of microplasmas that appear in $p-i-n$ structures. The appearance of microplasmas is attributed in [2] to the spreading of the current to quasineutral parts of $p-n$ junction and to rising dependence of the avalanche multiplication coefficient M on the free-carrier density n in the i-type region of $p-i-n$ structure. The dependence $M(n)$ is related to a redistribution of the electric field in the i-type region on increase in the density of carriers formed by impact ionization (fig.1a). It is shown in that microplasma can be excited by an external brief local perturbation of $p-i-n$ structure, which is perfectly homogeneous over its area. Inhomogeneities, always present in real $p-n$ junction, then act as nuclei of spontaneous formation of microplasmas.

Nevertheless, this work did not explain a phenomenon of spontaneous oscillations of microplasmas. In the present manuscript we propose a mathematical model, which can explain a phenomenon of current flow in the form of microplasmas impulses. A strong avalanche current in a channel of microplasma leads to selfheating of $p-n$-structure and disappearing of microplasma. Such a dependence modifies the character of the current flow and leads to relaxation oscillations of current density in $p-n$ structure. In general, microplasma can be regarded as spike oscillation autosolitons, which are regions far from equilibrium that appear in systems which a close to equilibrium [5] (fig.1b).

Mathematical model

We consider the model scheme of $p-n$ junction. It consists of the thin nonlinear region N with large specific resistance and nonlinear properties and of the quasineutral region L with linear properties. The current flows through this structure as is shown on fig.1c. Thus, for the description of current flow we must take into account processes in N and L regions.

The distribution of current density in the nonlinear thin film is described by the well-known equation [8,9]

$$\tau_n \frac{\partial n}{\partial t} = l^2 \Delta n - q(n, V_i),$$

where n is the concentration of current carrier, τ_n and l are the characteristic values of time and space relaxation respectively, V_i is the voltage drop on the thin film and q is some nonlinear function.

The spreading of the current in ohmic region L is described by the equation of continuity

$$C \frac{\partial V_i}{\partial t} = div \vec{j},$$

where C is the capacitance per unit area of the junction.

Equation describing the voltage distribution V_i can be obtained by the presentation of quasineutral region by an equivalent circuit containing series- and parallel-connected passive resistances [3,7]. This equation can be obtained more strictly in mathematical sense.

For this purpose we shall find the potential distribution in quasineutral domain at the condition that it is set the value of voltage $\tilde{V}(x) = V - V_i$ at the boundary of quasineutral domain, etc. at $z = h$ (fig.1c).

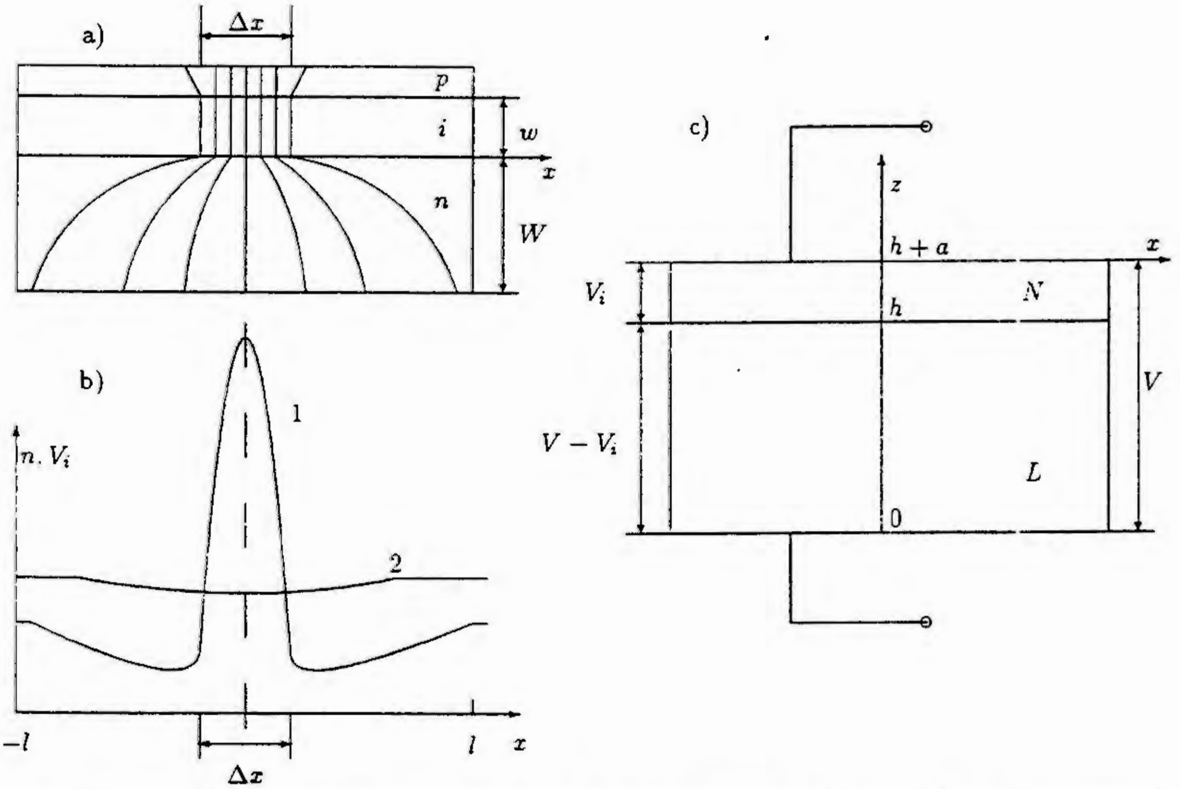


Fig.1 Physics of the appearence of microplasmas in p-i-n structures: a) lines of flow of the current in the space charge layer and in the quasineutral n-type regions of a $p - i - n$ structure in the presence of a microplasma of size Δx at a point $x = 0$; b) qualitative distribution of the current density j(x), i.e., of the carrier density $n(x)$ (curve 1), and the voltage drop V_i across the space charge layer of the $p - i - n$ structure (curve 2) in a microplasma; c) a scheme of a drop voltage on $p - n$ structure.

Presenting $\tilde{V}(x)$ in the form of Fourier integral, we get

$$\tilde{V}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \tilde{V}(k) e^{ikx} dk = V - V_i,$$

where

$$\tilde{V}(k) = \int_{-\infty}^{\infty} \tilde{V}(x) e^{-ikx} dx,$$

We shall find the potential distribution φ in the form of harmonic function in quasineutral domains. Expanding $\varphi(x, z)$ into Fourier integral

$$\varphi(x, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \varphi(k, z) e^{ikx} dk$$

we obtain the following boundary problem for $\varphi(k, z)$

$$\varphi''_{zz} - k^2 \varphi = 0$$

$$\varphi |_{z=0}= 0, \qquad \varphi |_{z=h}= \tilde{V}(k).$$

As a result it is easy to get the solution

$$\varphi(k,z) = \frac{\tilde{V}(k)}{\mathrm{sh}(kh)}\mathrm{sh}(kx)$$

Separating the homogeneous $\tilde{V}_-$ and inhomogeneous $\tilde{V}_\sim$ parts we get the following expression for the potential $\varphi(x,z)$ in quasineutral domains

$$\varphi(x,z) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{\tilde{V}(k)dk}{\mathrm{sh}kh}e^{ikx}\mathrm{sh}kz + \frac{\tilde{V}_-}{h}z$$

At the boundary between the space charge region and quasineutral region of $p-n$ structure the boundary conditions of the form

$$C\frac{\partial V_i}{\partial t} + j = \frac{h}{\rho}\frac{\partial\varphi}{\partial z} |_{z=h}$$

must be satisfied.

Substituting the value of $\varphi(x,z)$ into boundary conditions, we obtain

$$c\frac{\partial V_i}{\partial t} + j = \frac{\tilde{V}_-}{\rho} + \frac{h}{2\pi}\int_{-\infty}^{\infty}(\mathrm{ctg}kh)k\tilde{V}(k)e^{ikx}dk$$

The equation for V_i is the equation of current continuity at the boundary "nonlinear - linear domain". The system of equations for the average carrier concentration (current density) and the voltage distribution on $p-n$ junction is closed and it allows to find inhomogeneous distribution of charge carrier in N as well as in L regions of $p-n$ structure. At that substantial the nonlocal dependence of current on the voltage in nonlinear N region is. In the case of not very large scale of inhomogeneous along axis x space distributions the equation for current flow through quasineutral domains can be substantially simplified. Instead, taking into account that the expansion cthkh decreases quickly enough at $kh < \pi$, we transform underintegral expression of the equation for V_i. Taking into account the first two terms of the expansion, we get

$$c\frac{\partial V_i}{\partial t} + j(x) = \frac{1}{\rho}\left[\tilde{V}_- + \frac{1}{2\pi}\int_{-\infty}^{\infty}(1-\frac{1}{3}k^2h^2)\tilde{V}(k)e^{ikx}dk\right]$$

Replacing the value ik under the integral by $\partial/\partial x$ we obtain the equation for spreading of the current in the form

$$c\frac{\partial V_i}{\partial t} + j = \frac{1}{\rho}(V-V_i) + \frac{1}{3}h^2\Delta V_i.$$

The equation for the voltage drop on the nonlinear region generalize the same equation obtained in [2,6].

Therefore, the model proposed in [2] leads to a system of two equations: the balance of the average (over the thickness of the space charge region) electron density

$$\frac{\partial n}{\partial t} = D\Delta_\perp n + n\nu_i(n,V_i) - \frac{n}{\tau_n} + G_T \tag{1}$$

and the continuity of the total current in quasineutral n- or p- type regions of the structure

$$C\frac{\partial V_i}{\partial t} = \sigma\tilde{W}\Delta_\perp V_i - J + (V-V_i)\rho^{-1}, \tag{2}$$

where ν_i is the average (over the thickness of the space charge region) rate of carrier ionization, related to the multiplication coefficient M with the obvious expression $M = (1-\nu_i\tau_n)^{-1}$; V_i is the voltage drop across the space charge region of the $p-n$ junction; $J = env_n$ is the avalanche current density; D and v_n are, respectively, the diffusion coefficient and drift velocity of electrons in space charge region; $\tau_n = w/v_n$ - is the transit time of carrier across the space charge region of $p-n$ junction; W is the thickness of the space charge region, C is the specific capacitance of $p-n$ structure; $\Delta_\perp = \partial^2/\partial x^2 + \partial^2/\partial y^2$, the z axis is selected to be along the normal to the $p-n$ junction plane; $\tilde{W}$ is the effective thickness of the region where the current spreads in the base of the p- or n-quasineutral regions; V is the total voltage drop across the $p-n$ structure; G_T is the rate of thermal and tunnel generation of carriers in the space charge region. We are following here for the fact that in the case of the investigated $p-n$ junctions we have $\rho_n = (W_n/\sigma_n) >> \rho_p = (W_p/\sigma_p)$, where σ_n, σ_p and W_n, W_p are the conductances and the thicknesses of the n- and p-quasineutral regions. so that in Eq.(2) we have $W = W_n$, $\sigma = \sigma_n$, and $\rho = \rho_n$. It is clear from Eqs. (1) and (2) that the characteristic spatial length of changes in the carrier density in the space charge region is $l = (D\tau_n)^{1/2}$, whereas the characteristic length of changes in the voltage drop V_i across the space charge region is $L = (W\tilde{W})^{1/2} \approx W$.

The investigation of physics and properties of microplasmas, conducted on the basis of model (1),(2), did not take into account the Joule heating of the structure. In that time as very high current density in a channel of microplasmas leads to the substantial heating of $p-n$ - structure in the localization region of microplasma. At very high values of current density J the Joule heating of the lattice

$$\Delta T = T - T_i = JVR_T \tag{3}$$

(T_i - is the temperature of thermostat, R_T - specific heating resistance of the structure) can be of order of $\Delta T = 20 - 100$ [4]. The changes of the parameters of avalanche breakdown with increasing of a temperature of semiconductor lead to the temperature dependence of the microplasma parameters, because the impact ionization coefficient of electrons and holes in a channel of microplasma is decreasing [4].

Because of that the ionization velocity ν_i in equation (1) is really the function not only from n and V_i, but also from the lattice temperature T. So, it is necessary to complete the system (1),(2) by the equation which describes the temperature distribution along the plane of $p-n$ junction, and to take into account in equation (1) decreasing dependence of the impact ionization velocity ν_i on a temperature. The temperature distribution along the plane of $p-n$ junction can be described by using the simplified (averaged over the thickness of the space charge region) heat transfer equation

$$\tau_T \frac{\partial T}{\partial t} = \lambda^2 \Delta_\perp T - (T - T_i) + \tilde{C} V_i n, \tag{4}$$

where τ_T, λ - are time and length of heat relaxation, $\tilde{C} = e v_n R_T$, R_T - specific heating resistance of the structure, T_i - the temperature of effective thermostat [1].

A linear analysis of Eqs. (1) and (2) shows that a homogeneous distribution of the avalanche current in the range $J > J_c$ becomes unstable in the presence of aperiodic growth of fluctuations n and V_i with the wave number $k_0 \approx (Ll)^{-1/2}$. Filamentation of the avalanche current in a region of size $k \sim k_0^{-1} (l \ll d \ll L)$ increases the impact ionization rate in this region and this in turn increases the avalanche current density.

Since the characteristic spatial lengths and times of temperature changes are much grater than characteristic ones of changes of carrier charge concentration and temperature feedback exists, so in the system the relaxation oscillations of current density must take place.

Numerical simulation

The system of equations (1),(2),(4) was solved numerically for parameters typical of a silicon $p-n$ structure $D = 1cm^2/s$, $v = 10^6 cm/s$,$w = 2 \cdot 10^{-5} cm$. $\tilde{W} = 4.5 \cdot 10^{-3} cm$, $C = 2 \cdot 10^{-8} F/cm^2$, $\rho = 1.5 \cdot 10^{-3} Om \cdot cm^2$, $J_0 = 10 A/cm^2$, $V_0 = 50V$, $R_t = 0.03 K \cdot m^2/Wt$, $j_T = G_T e v_r = 10^{-4} A/cm^2$, $\tau_T = 2.5 \cdot 10^{-5} s$, $\lambda^2 = 1.25 \cdot 10^{-5} cm^2$, $T_i = 300K$. In numerical investigation for $\nu_i(n, V_i, T)$ we considered the specific case described by the expresion

$$\nu_i(V_i, T, j) = \nu(j) \exp(V_0(1 - (\exp(-\beta(1-T))/V_i^2))), \tag{5}$$

$$\nu_i(j) = a + b \exp(1 - \frac{c}{j}), \tag{6}$$

where the depedence of ν_i on V_i and T is of standard form, and the dependence $\nu_i(j)$ is typical for $p-n$ structures, in which stratification of the current has be observed [3,4,9]. Inhomogeneities in $p-n$ junctions were simulated by the local change of ionization function $\nu_i(n, V_i, T)$ and the rate of thermal and tunnel generation of carriers in the space charge region G_T [6].

When the external voltage reaches the value $V = V_c$, in the inhomogeneous region $\approx \Delta x$ in a time of order $\tau_V = C\rho$ spontaneously arise inhomogeneous current flow (fig. 2a). The local heating of the lattice in the localization region of microplasma leads in a time of order of τ_T to its decreasing in amplitude and in a time of order of τ_V to its disappearing and transformation to the state of homogeneous current flow.

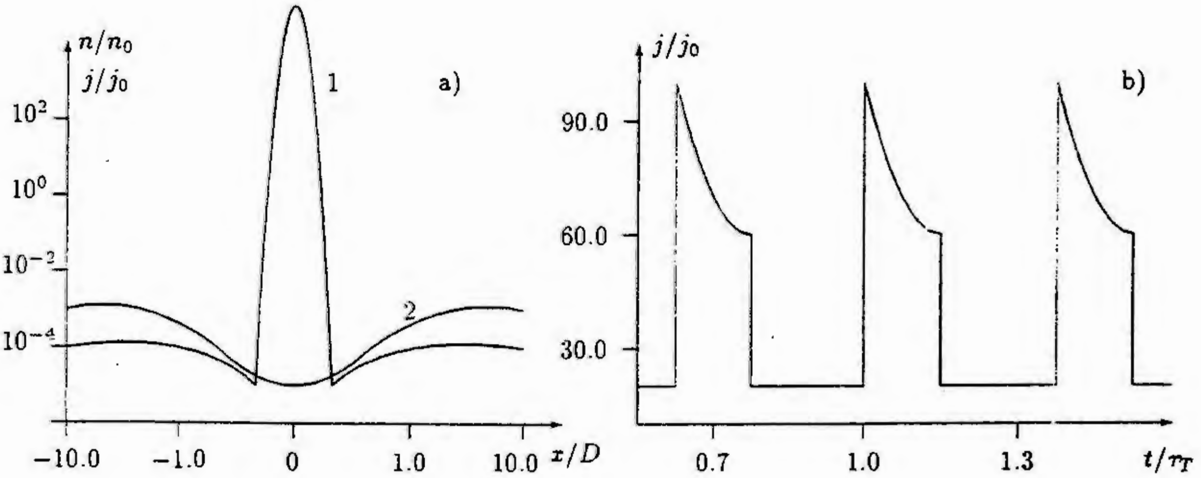


Fig.2 Dynamics of the current flow in a form of microplasma pulsations: a) a spatial distribution of current density in the excited (curve 1) and unexited (curve 2) states; b) dynamics of change of the current density over the $p-n$ sctructure.

In the result the lattice is cooling and system comes back to the state, when microplasma can spontaneously arise again. The arising of new microplasma leads to a new heating of structure and further (in a time of order of τ_T) to its disappearing. So, in the structure the local relaxation oscillations of current density which lead to the jumps of the current on the current-voltage characteristic of $p-n$ junctions take place. The Fig.1b presents the oscillation dynamics of the current density. At the decreasing of resistance of quasineutral region the form of impulses tends to reach the rectangular one, and at the increasing - the triangular one (fig 3).

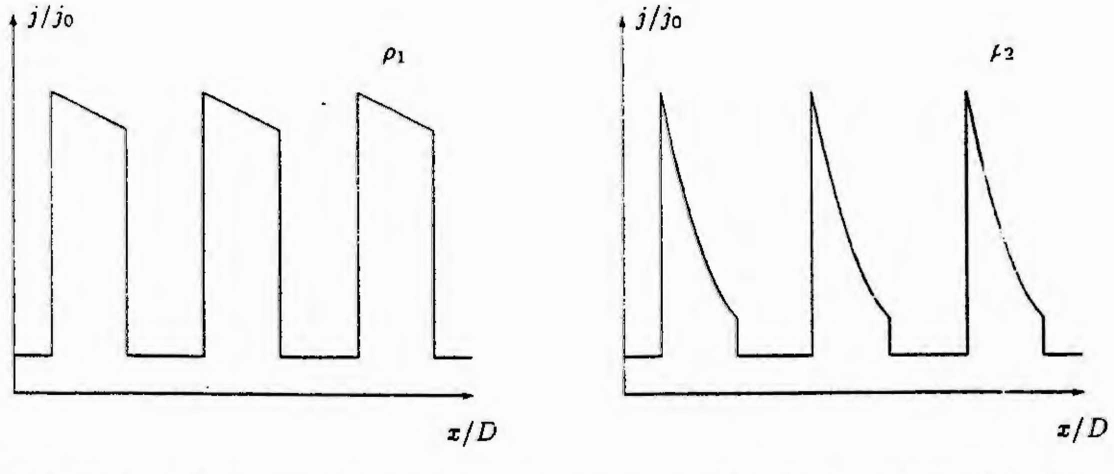


Fig.3 Dynamics of the current flow in a form of microplasma pulsations at the decreasind of resistance of quasineutral regions ($\rho_1 < \rho_2$)

At the increasing of the total voltage drop V across a $p-n$ junction structure an amplitude of impulses is practically invariable, but the impulse width increases substantially and pauses between them decrease (fig.4).

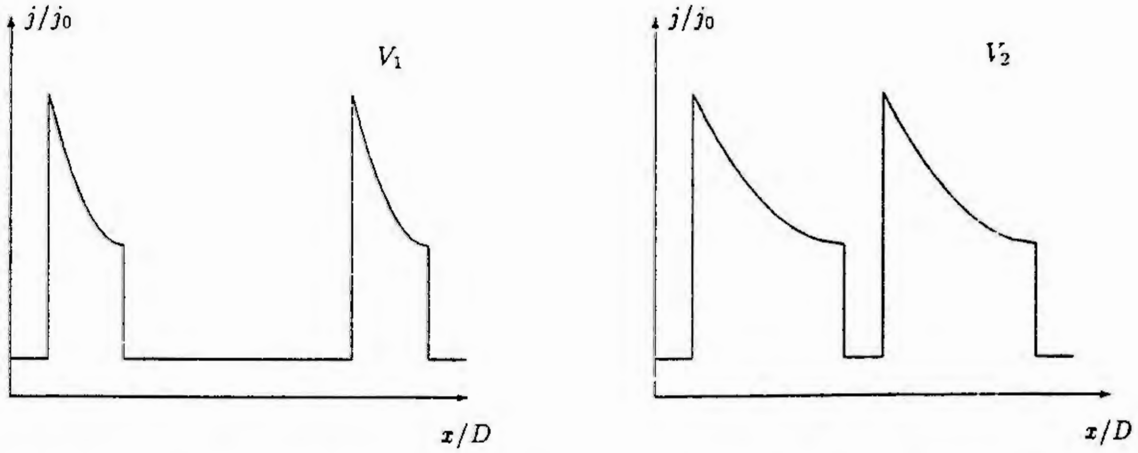


Fig.4 Dynamics of the current flow in a form of microplasma pulsations at the increasind of the total voltage V across the $p-n$ structure ($V_1 < V_2$).

These results are in very good agreement with the theory of autosolitons [5] as well as with experimental investigations of microplasmas [4].

References

1. Bonch-Bruevich, V.L., Kalashnikov, S.G., Physics of Semicjductors. Nauka, Moscov (in Russian), 1977.

2. Gafiychuk, V.V., Datsko, B.I., Kerner, B.S., Osipov, V.V., Microplasmas in perfectly homogeneous $p-i-n$ structures. Sov. Phys. Semicond. v.24, 4 (1990), 455-459.

3. Gafiychuk, V.V., Datsko, B.I., Kerner, B.S., Osipov, V.V., Spontaneous formation and evolution of local impact ionization regions in perfectly homogeneous $p-n$ structures. Sov. Phys. Semicond. v.24, 7 (1990), 806-811.

4. Greckhov, I. and Sereshkin Yu., Avalanche breakdown of $p-n$ junction in semiconductors. Energiya, Leningrad (in Russian), 1980

5. Kerner, B.S., Osipov, V.V., Autosolitons: A New Approach to Problem of Selforganization and Turbulence, Kluver, Dortrecht, 1994.

6. Muller, M.W. and Guckel, H., Avalanche Injection and Second Breakdown in Transistors. IEEE Trans.Electron Devices, ED-15, (1968), 320-335.

7. Purwins, H.G., Radehaus, C. and Berkemeier, J., Experimental Investigation of Spatial Pattern Formation in Physical Systems of Activator Inhibitor Type. Z.Naturforsch. T.A43, 10 (1988), 17-29.

8. Scott, A., Active and Nonlinear Wave Propagation in Electronics, Wiley, New York. 1970.

9. Sze, S.M., Physics of Semiconductor Devices. 2nd ed.,Wiley Interscience, New York. 1981.

MATHEMATICAL MODELLING AND COCHLEAR IMPLANTS

Petra Lutter
Universitätsspital, ORL-Klinik
Frauenklinkstr. 24, CH-8091 Zürich[1]
email: plutter@ws1.atv.tuwien.ac.at

Abstract. For the purpose of improving signal processing strategies for cochlear implant patients a two-step approach is taken: Results of a macromechanical model of the cochlea that describes the movement of the basilar membrane in the inner ear provide hints for the design of new strategies. In a second step a nerve model that predicts nerve fiber reactions to electrode stimulation allows already existing strategies to be tested.

Introduction

About 16,000 hearing-impaired people have been provided with cochlear implants to date. These are auditory prostheses for hearing-impaired people with a damaged cochlea (inner ear), but an intact 8th nerve (hearing nerve). Cochlear malfunction is compensated by a set of electrodes that are inserted into the inner ear to electrically stimulate the 30,000 fibers of the primary auditory nerve. Together with a speech processor these implanted electrodes form the basic components of a cochlear implant.

Although many modelling attempts have been made to explain cochlear function [6], the theory often lags behind the technical development. Unfortunately, models can only reproduce *certain* limited aspects of experimental data. Due to this lack of a compound cochlear model, this article concentrates on two aspects, a macromechanical model to explore speech coding mechanisms, and a nerve model to predict nerve fiber reaction to electrode activation.

A Macromechanical Model of the Cochlea

The main task of the cochlea is to transform mechanical information in the form of sound pressure into neural signals. The heart of this transduction process is situated in the scala media, one of the 3 fluid-filled compartments of the cochlea. A row of 3500 inner hair cells registers the motion of the basilar membrane (BM) with the help of their cilia (hairs). When these cilia are bent as a consequence of BM motion, action potentials are evoked in the connected nerve fibers.

Acoustical signals are represented in the auditory nerve both by place-rate information - every place along the BM has its characteristic frequency - and by the temporal fine structure of the time differences in the firing pattern.

The biophysical model chosen for this investigation of speech coding concentrates on the fluid motion of the cochlea (macromechanical) and ignores mechanical details such as hair cell structure or interaction between hair cells (micromechanical). The reason for this classical one-dimensional approach is the still incomplete knowledge about nonlinear and active cochlear phenomena [1]. Based on the Peterson and Bogert model [8] and modified according to the numerical solution method of Diependaal et al. [3], the basic equations read

$$p(x,t) = m(x)\ddot{u}(x,t) + r(x,t)\dot{u}(x,t) + s(x,t)u(x,t)$$

$$p''(x,t) - [2\rho\beta(x)/a]\ddot{u}(x,t) = 0,$$

where $\dot{u}$ means differentiation with respect to time and p' denotes differentiation with respect to place x (the BM length coordinate); $p(x,t)$ denotes transmembrane pressure, $m(x) = 0.5$ mg/mm^2 denotes BM mass, $u(x,t)$ denotes BM displacement, $r(x,t) = 1.12$ mg/mm^{2}ms denotes BM resistance, $s(x,t) = 20000e^{-0.3x}$ mg/(mm^{2}ms^2) denotes BM stiffness, $\rho = 1$ mg/mm^3 denotes fluid density, $\beta(x) = 0.1$ mm denotes BM width, $a = 1$ mm^2 denotes cross section area of one fluid channel (both channels are assumed to be equal in size), and $l = 35$ mm denotes BM length.

The model allows computation of BM reactions with regard to time or place. Fig. 1 shows computed BM reactions to a short segment of the natural speech sound /e/ displayed on top of the figure. From

[1] On leave of absence from Inst. f. Analysis, Techn. Math. u. Vers.math., TU Wien, Wiedner Hauptstr. 8-10/114, A-1040 Wien, Austria

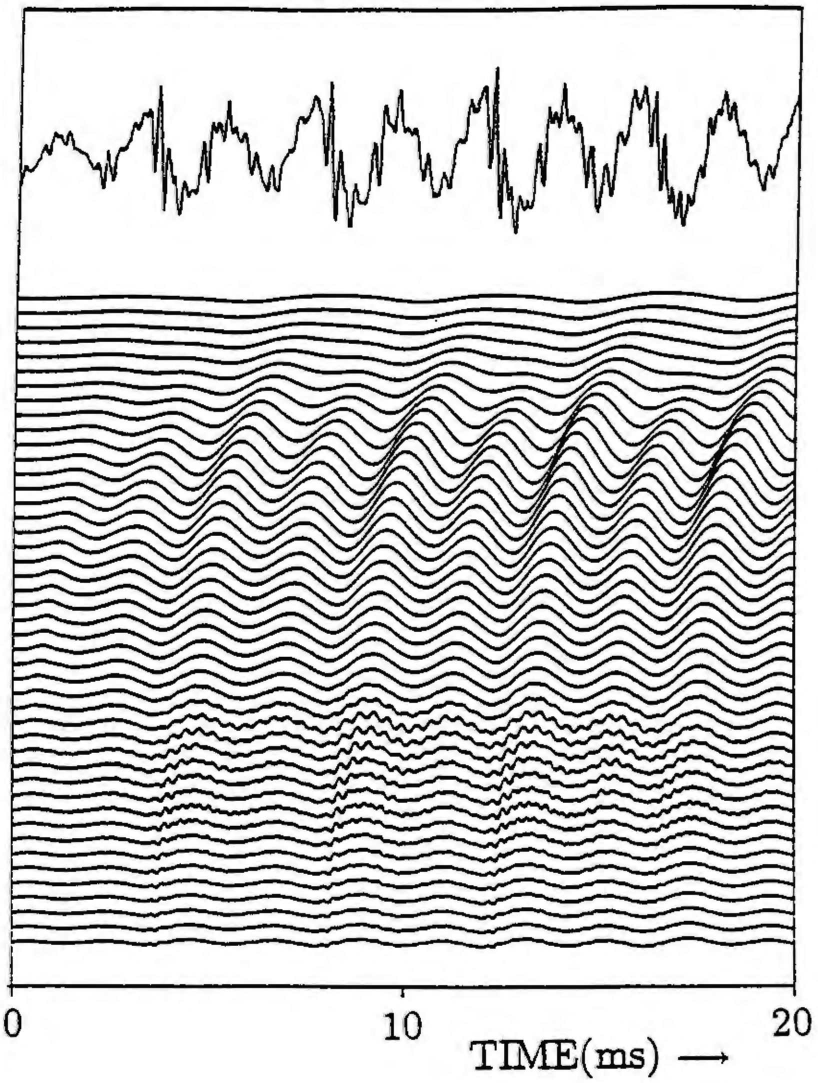


Fig. 1: Computed BM vibrations generated by signal /e/ from a female speaker (upper trace). Lines in the lower frequency region (upper part) are compressed to fit them in the figure. Simulation was done with SIMUL_R, see text for model data.

bottom to top the lines cover a frequency range from 10 kHz to 100 Hz. From what is known about place-rate coding strategies one would expect that each frequency place had its specific representative line in the figure. What can be observed, however, is not a continuous variation of response patterns, but a discrete change of whole bands of similar frequency. This capture effect has also been experimentally observed in recordings of auditory nerve activity [11]. Additional evidence for the importance of temporal speech coding mechanisms, which in turn is to be integrated into advanced signal processing strategies for cochlear implant patients, comes from comparing model data with tuning curves and click experiments [7,10].

Signal Processing Strategies

Speech signals for cochlear implant patients are typically processed in the following way: After filtering and analog-to-digital-conversion, the power spectrum is calculated via fast Fourier transform and different frequency areas are assigned to different electrodes. Alternatively, this electrode mapping could be achieved through analog filtering. After extraction of specified speech features the coding strategy is then applied and the stimulus parameters are transferred to the electrodes.

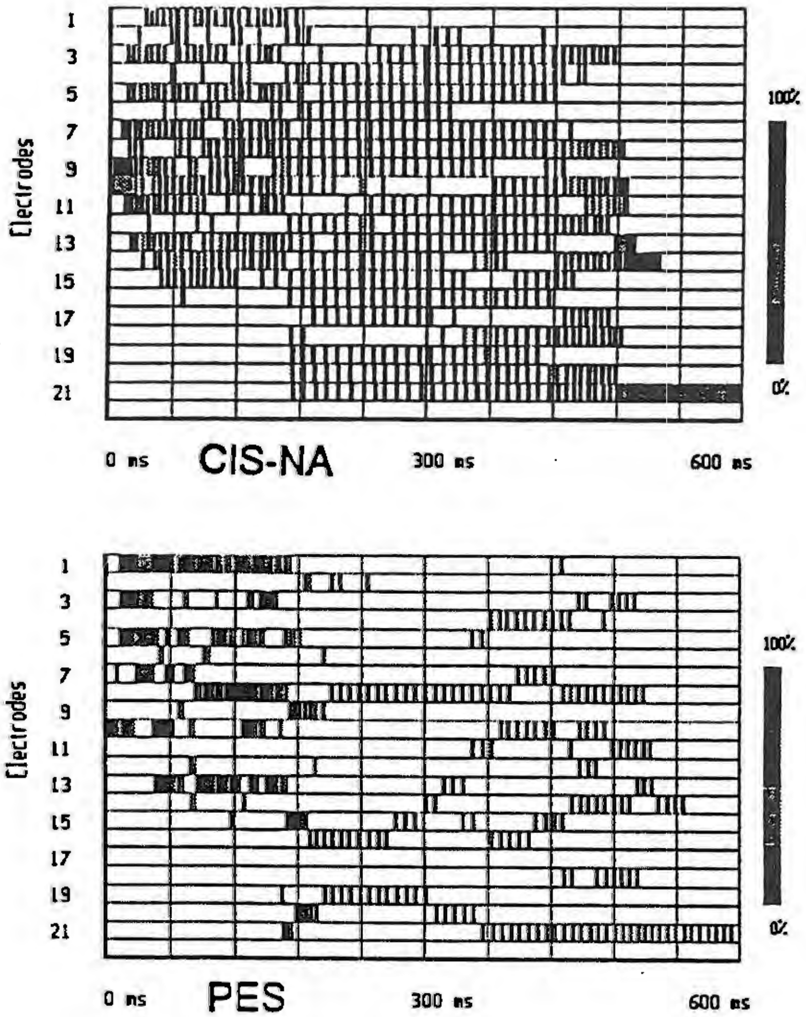


Fig. 2: Electrodograms of the German word "Schein" for the processing strategies PES and CIS-NA; after [4].

Fig. 2 shows electrode activation (electrodograms) with two different strategies applied to the German word "Schein". While the strategy PES (Pitch Excited Sampler) uses voice pitch to control the pulse rate of any given electrode, CIS (Continuous Interleaved Sampler) uses a stimulation pulse rate which is independent of the fundamental frequency of the input signal. CIS-NA, a variation of CIS, uses all narrow band analysis channels whose values exceed a preset noise cut level. While PES leads to good voice pitch discrimination, CIS allows for better consonant identification due to the higher stimulation rate. Promising studies with hybrid strategies [4] have shown that speech discrimination can be improved considerably. The results indicate that consonant identification may be enhanced by more detailed temporal information and specific speech feature transformations.

Modelling Nerve Behavior

To predict nerve fiber behavior in reaction to different strategies a mathematical model based on the Hodgkin-Huxley [5] equations has been designed. The Hodgkin-Huxley model is the only one known to reflect important experimental phenomena such as multiple spiking, maximum firing rate or chronaxy [9]. Fig. 3(a) shows the geometric situation in one single nerve fiber. The myelinated nerve membrane is cut into tiny segments according to the nodes of Ranvier, the only place for ionic currents to enter the nerve. At each node the membrane is simulated by an electric circuit consisting of capacitance, voltage source, and nonlinear resistance. $V_{e,n}$ and $V_{i,n}$ are the external and the internal potential at the nth

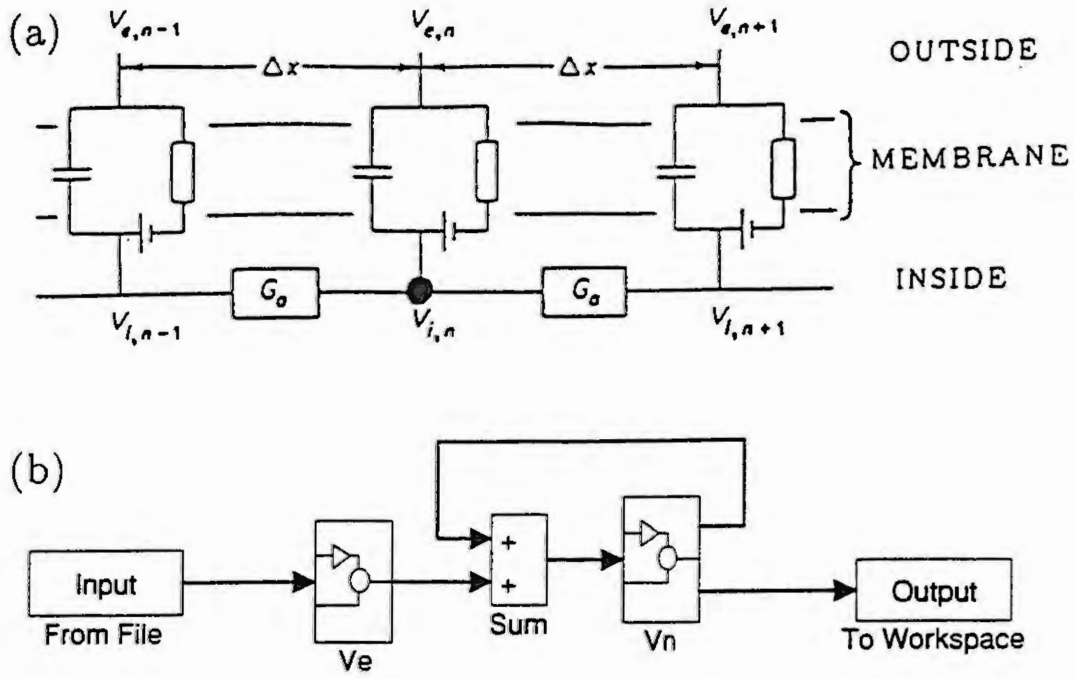


Fig. 3: (a) Electrical network to simulate the currents in a nerve fiber. The membrane of every cylinder of length Δx is simulated by an electric circuit. Segments are connected by G_a, the conductance of axoplasm. (b) Block diagram of Simulink-implementation of the nerve model, see text for details.

segment, G_a denotes the conductance of axoplasm between two segments.

By introducing the reduced voltage

$$V_n = V_{i,n} - V_{e,n} - V_{rest},$$

where V_e is the external potential and V_{rest} is the inside resting potential, the main equation to simulate the uniform nerve fiber reads

$$\frac{dV_n}{dt} = \{-i_{ionic} + \frac{d}{4\rho_i \Delta x \cdot L} \cdot [(V_{n-1} - 2V_n + V_{n+1}) + (V_{e,n-1} - 2V_{e,n} + V_{e,n+1})]\}/c_m,$$

with $d = 0.00015$ cm being the axon diameter, $\rho_i = 100$ Ω.cm being the intracellular resistance, $\Delta x = 0.023$ cm being the internodal length, $L = 0.0001$ cm being the nodal gap width, and $c_m = 1$ μF/cm^2 being the membrane capacitance. V_e is approximated by ohmic resistance:

$$V_e = \frac{\rho_e I_{el}}{4\pi r},$$

where I_{el} is the electrode current, r gives the distance to the electrode, and $\rho_e = 300$ Ω.cm is the specific resistance of the extracellular medium.

Fig. 3(b) shows the implementation of this system in Simulink (Matlab): The output of the block V_e enters the block V_n, where the ionic currents are determined according to the Hodgkin-Huxley equations. An arbitrary number of nerve fibers can easily be combined to form a larger network of neural arrays.

Fig. 4 shows the natural speech stimulus /bob/ (upper panel) and the simulated neural response after being processed through the Simulink model as follows: At different distances from one electrode the reactions (action potentials) of 16 nerve fibers each consisting of 9 segments have been calculated separately. The total information of these 16 fibers has been summed up to form a resynthesized hypothetical whole nerve potential which is called the "neurophonic signal" (lower panel). Listening to these reconstructed signals demonstrates the rather poor speech intelligibility achieved through single channel stimulation [12].

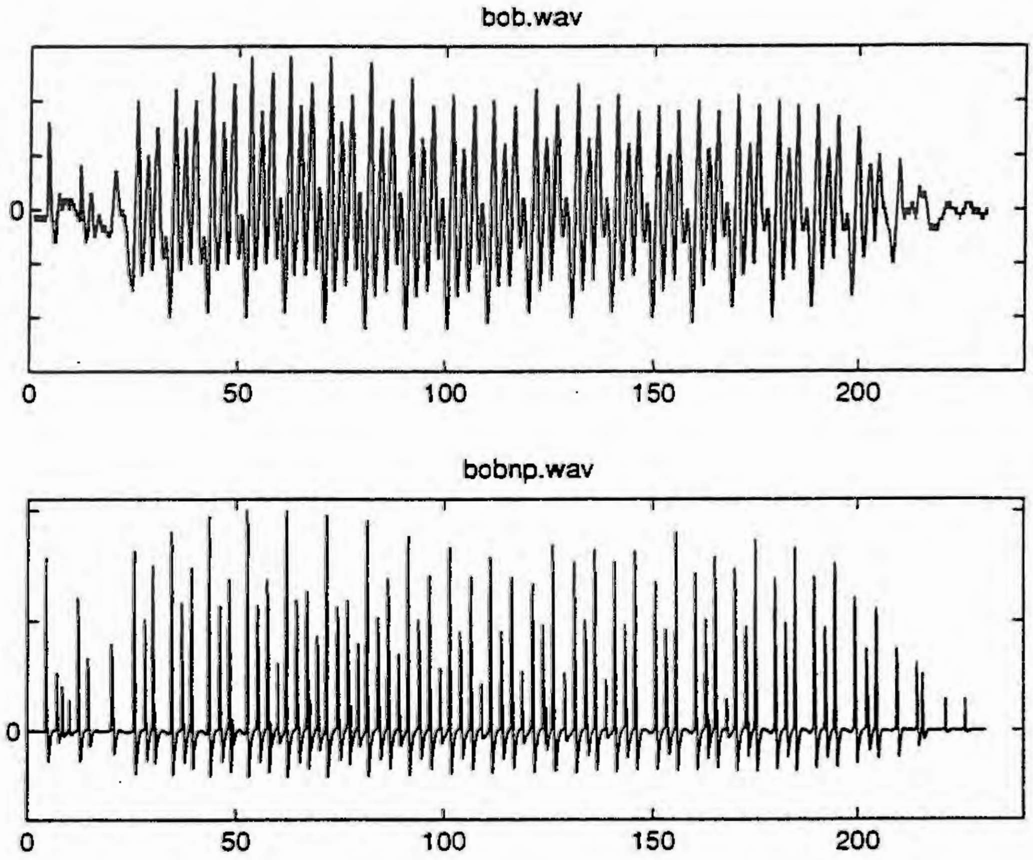


Fig. 4: Natural speech stimulus /bob/ (upper panel) spoken by a male speaker. Horizontal axis is time in ms. Lower panel: Neurophonic signal /bob/ as output of Simulink-model, see text for model data. Vertical scales (level normalizing) in arbitrary units.

Conclusions

Two modelling approaches have been presented to improve cochlear signal processing strategies: A macromechanical model of the cochlea indicates the necessity for increased emphasis on temporal speech coding aspects when designing signal processing strategies. Initial results of modelling the nerve behavior help predict compound nerve fiber responses to single channel stimulation. As different electrodes are usually not simultaneously active in modern cochlear implants, modification of the model for the multichannel case is a logical next step. Experimental verification of the models can be expected from new techniques of directly measuring nerve responses in cochlear implant patients (see e.g. [2]), which are about to become a clinical tool in the near future.

Acknowledgement. This research was supported by research grant J01229-TEC from the Austrian "Fonds zur Förderung der wissenschaftlichen Forschung" (FWF).

References

1. de Boer, E., Some like it active. In: Biophysics of Hair Cell Sensory Systems (Eds.: Duifhuis, H., Horst, J.W., van Dijk, P., and van Netten, S.M.), World Scientific, Singapore, 1993, 1-22.

2. Brown, C.J., Abbas, P.J., Borland, J., and Bertschy, M.R., Electrically Evoked Whole Nerve Action Potentials in Ineraid Cochlear Implant Users: Responses to Different Stimulating Electrode Configurations and Comparison to Psychophysical Responses. Journal of Speech and Hearing Research, 39 (1996), 453-467.

3. Diependaal, R.J., Duifhuis, H., Hoogstraten, H.W., and Viergever, M.A., Numerical methods for solving one-dimensional cochlear models in the time domain. J. Acoust. Soc. Am., 82 (1987), 1655-1666.

4. Dillier, N., Lai, W.K., and Bögli, H., A High Spectral Transmission Coding Strategy for a Multi-Electrode Cochlear Implant. In: Advances in Cochlear Implants (Eds.: Hochmair-Desoyer, I.J. and Hochmair, E.S.), Manz, Vienna, 1994, 152-157.

5. Hodgkin, A.L. and Huxley, A.F., A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol., 117 (1952), 500-544.

6. Hubbard, A.E. and Mountain, D.C., Analysis and Synthesis of Cochlear Mechanical Function Using Models. In: Auditory Computation (Eds.: Hawkings, H.L., McMullen, T.A., Popper, A.N., and Fay, R.R.), Springer, N.Y., 1996 (ISBN 0-387-97843-7), 62-120.

7. Lutter, P., Rattay, F. and Mark, H.E., Computer simulations enlighten the old controversy in speech perception: Tonotopic versus temporal coding. In: Proc. EUROSIM '95, Vienna (Eds.: Breitenecker, F. and Husinsky, I.), Elsevier, North Holland, 1995 (ISBN 0-444-82241-0), 957-962.

8. Peterson, L.C. and Bogert, B.P., A dynamical theory of the cochlea. J. Acoust. Soc. Am., 22 (1950), 369-381.

9. Rattay, F., Electrical nerve stimulation. Springer, Wien-New York, 1990.

10. Rattay, F. and Lutter, P., Speech Sound Representation in the Auditory Nerve: Computer Simulation Studies on Inner Ear Mechanisms. ZAMM, 1996 (in press).

11. Shamma, S., Speech processing in the auditory system I: The representation of speech sounds in the responses of the auditory nerve. J. Acoust. Soc. Am., 78 (1985), 1612-1621.

12. Stüger, V., Hearing into the auditory nerve of cochlear implant persons by computer simulation. Diploma thesis, Techn. Univ. Vienna, 1996.

SIMULATION ANALYSIS OF THE EFFECTS OF THE JUNCTIONAL FOLDS ON SPONTANEOUS GENERATION OF THE MINIATURE ENDPLATE CURRENT AT NEUROMUSCULAR JUNCTION

Takashi Naka
Saitama Junior College
Hanasaki-Ebashi 519-5, Kazo-shi, Saitama 347, Japan

Abstract. The dynamic behavior of acetylcholine in the synaptic cleft with a junctional fold is modelled as a reaction-diffusion system in an axis-symmetrical two-dimensional space. The effects of the junctional fold are analyzed by computer simulation on spontaneous generation of the miniature endplate current at the neuromuscular junction. The amplitude of the current may be maximized at an optimal value of the width of the junctional fold, while the depth causes less effects.

Introduction

In processing of the neuronal signals, the synaptic chemical transmission is an important process, and investigation of the molecular events in the process has led to the neurotransmitter theory [5]. The dynamic behavior of neurotransmitter in diffusion through the synaptic cleft and action at the synaptic membranes is engaged in a fundamental function for the transmission process. Analysis of such behavior can be performed most appropriately with representation of the transmission process as a reaction-diffusion system (RD system) for neurotransmitter because the experimental analysis still is practically difficult for the molecular processes in the cleft.

Some mathematical models for the dynamic behavior of acetylcholine (ACh), a typical neurotransmitter, in spontaneous generation of the miniature endplate current (MEPC) at the neuromuscular junction have been proposed to analyze the transient process of the synaptic chemical transmission. The one-dimensional compartment models of Rosenberry [9] and Thomas [2] is extended to a two-dimensional compartment model in our previous study [7]. It is revealed from analysis of our model that the radial diffusion process of ACh has more distinctive effects on spontaneous generation of the MEPC than the transverse diffusion process. In this study the two-dimensional compartment model is applied for examination of the effects of junctional folds of the postsynaptic membrane on the response characteristics of the MEPC.

Modelling of the synaptic chemical transmission

In the RD system as illustrated in Fig. 1, the ACh concentration is assumed to vary with time t and point (x, r) in a two-dimensional space of axis-symmetrical disc of the synaptic cleft (the range of space variables: $0 \leq x \leq d_c$, $0 \leq r \leq w_c$) and the concentric cylinder representing the junctional fold (the range of space variables: $d_c \leq x \leq d_c + d_f$, $0 \leq r \leq w_f$), due to influx of ACh through a circular area on the presynaptic membrane ($r \leq a$ [$< w_c$] on the top surface boundary at $x = 0$), transverse and radial diffusion in the synaptic cleft and the junctional fold, and interactions with acetylcholinesterase (AChE) in the light grayed area and ACh receptor (AChR) in the dark grayed area. The junctional fold is simplified as a concentric cylinder with its top surface attached to the bottom of the disc in order to avoid the additional dimmension for the RD system. The fold thus opens a hole to the synaptic cleft at the postsynaptic membrane.

The interaction of ACh with functionally dimeric AChR follows the minimal mechanism [2, 4] as given in:

$$\mathrm{ACh} + \mathrm{R} \underset{k_{-r}}{\overset{2k_r}{\rightleftharpoons}} \mathrm{R}_1 \qquad \mathrm{ACh} + \mathrm{R}_1 \underset{2k_{-r}}{\overset{k_r}{\rightleftharpoons}} \mathrm{R}_2 \underset{k_c}{\overset{k_o}{\rightleftharpoons}} \mathrm{R}_o \tag{1}$$

where R and R_1 indicate the AChR species free and singly bound with ACh, respectively, and the AChR species doubly bound with ACh, R_2 and R_o, are associated with ion channel function of AChR so that the closed channel form R_2 interconverts to the open channel form R_o. The k_i's ($i = r, -r, o, c$) are the

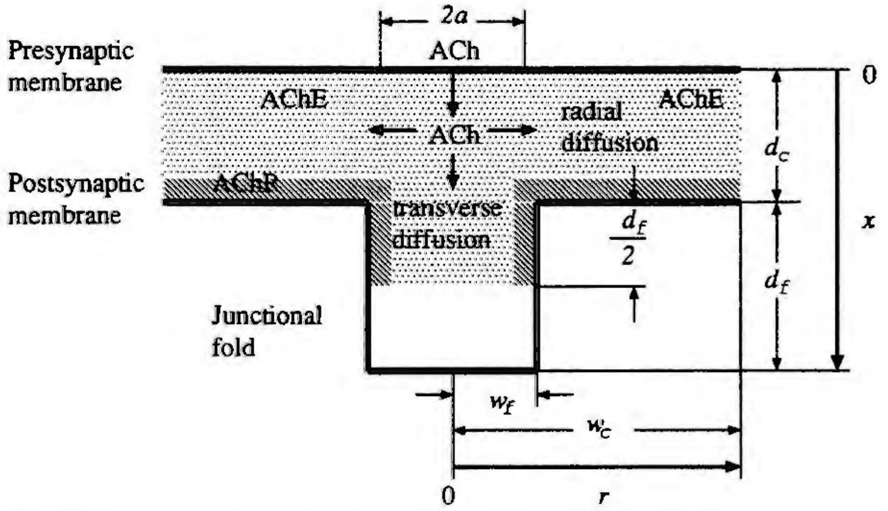


Fig.1. Rection-diffusion system for ACh in a two-dimensional space of axis-symmetrical disc of the synaptic cleft with a junctional fold. The radius of the release area in the presynaptic membrane is denoted by a. The light and dark grayed areas indicate the AChE- and AChR-distributed regions, respectively.

rate constants for the respective steps in the mechanism. The reaction of AChE proceeds in the following mechanism originally proposed by Rosenberry [8]:

$$\mathrm{ACh} + \mathrm{E} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{X}_1 \overset{k_2}{\rightarrow} \mathrm{X}_2 + \mathrm{Ch} \qquad \mathrm{X}_2 \overset{k_3}{\rightarrow} \mathrm{E} + \text{acetate} \tag{2}$$

where E, X_1 and X_2 denote AChE species free, complexed with ACh and acetyl group, respectively. The k_i's ($i = 1, -1, 2, 3$) are the rate constants for the respective steps in the mechanism.

The RD system for ACh in the synaptic cleft and the junctional fold is thus represented by a two-dimensional diffusion equation with nonlinear reaction terms, accompanied by the rate equations for AChE and AChR as follows:

$$\begin{aligned}
\frac{\partial A}{\partial t} &= D(\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial r^2} + \frac{1}{r}\frac{\partial A}{\partial r}) - k_1 AE + k_{-1}X_1 - 2k_r AR + (k_{-r} - k_r A)R_1 + 2k_{-r}R_2 \\
\frac{dE}{dt} &= -k_1 AE + k_{-1}X_1 + k_3 X_2 \\
\frac{dX_1}{dt} &= k_1 AE - (k_{-1} + k_2)X_1 \\
\frac{dX_2}{dt} &= k_2 X_1 - k_3 X_2 \\
\frac{dR}{dt} &= -2k_r AR + k_{-r}R_1 \\
\frac{dR_1}{dt} &= 2k_r AR - (k_{-r} + k_r A)R_1 + 2k_{-r}R_2 \\
\frac{dR_2}{dt} &= k_r AR_1 - (2k_{-r} + k_o)R_2 + k_c R_o \\
\frac{dR_o}{dt} &= k_o R_2 - k_c R_o
\end{aligned} \tag{3}$$

where the italic capital letter denotes the concentration of the respective chemical species at point (x, r) and time t and $A(x, r, t)$ expresses the ACh concentration. D is the diffusion coefficient of ACh.

The boundary conditions for ACh are expressed by

$$\begin{aligned}
\frac{\partial A(x,r,t)}{\partial x} &= 0 \quad \text{at } x=0,\ x=d_c \text{ for } r \geq w_f,\ x=d_c+d_f \text{ for } r \leq w_f; \\
\frac{\partial A(x,r,t)}{\partial r} &= 0 \quad \text{at } r=0,\ r=w_f \text{ for } d_c \leq x \leq d_c+d_f; \\
A(x,r,t) &= 0 \quad \text{at } r=w_c \text{ for } x \leq d_c
\end{aligned} \tag{4}$$

so that ACh cannot leak out at the boundaries of the disc and the cylinder except at the side surface of the disc where ACh is removed by radial diffusion. The initial condition at $x = 0$ corresponds to the release of a single quantal packet of ACh, and is expressed by an impulse-wise increase in $A(0, r, 0)$ $(r \leq a)$ from 0 to an appropriate concentration for a quantal packet of ACh. There initially exist no ACh inside the disc.

For simulation under the specified boundary and initial conditions, the method of lines [10] is applied to discretize the partial differential equation with respect to the space variables for the transverse and radial coordinates, resulting in the two-dimmensional compartment model [7]. The rate equations (ordinary differential equations) thus derived for ACh, AChE and AChR are numerically integrated with respect to time by the Gear method [6] to yield the spatial and temporal changes in concentrations of ACh in the disc and the cylindrical hole, and of the open channel form of ACh at the bottom of the disc and at the side surface of the cylindrical hole. The total number of the open channel form of ACh is given by

$$C(t) = \int_{w_f}^{w_c} R_o(d_c, r, t)dr + \int_{d_c}^{d_c+d_f} R_o(x, w_f, t)dx \tag{5}$$

which is assumed to be linearly correlated to generation of the MEPC.

The following values of the kinetic parameters [2] are used for the simulation in this study:

$$\begin{aligned}
k_r &= 30\text{mM}^{-1}\text{msec}^{-1}, k_{-r} = 10\text{msec}^{-1}, k_o = 20\text{msec}^{-1}, k_c = 5.0\text{msec}^{-1}; \\
k_1 &= 200\text{mM}^{-1}\text{msec}^{-1}, k_{-1} = 1.0\text{msec}^{-1}, k_2 = 110\text{msec}^{-1}, k_3 = 20\text{msec}^{-1}
\end{aligned} \tag{6}$$

The value for R_T (total concentration of AChR at a point; $= R + R_1 + R_2 + R_o$) used in this study is derived from the surface density of AChR ($2 \times 10^4 \mu\text{m}^{-2}$), and dependent on the volume of the discretized space in which AChR is uniformly distributed. The R_T values for the minimal compartment model with the critical radius determined below correspond to $R_T = 2.0$mM in the cleft, and $R_T = 1.33$mM in the fold of 500nm depth, or 0.89mM in the fold of 1000nm depth. Consideration of the size of the synaptic vesicles leads us to set $a = 50$nm and $d_c = 50$nm is known. The value for E_T (total concentration of AChE at a point; $= E + X_1 + X_2$) is set to 74μM.

Though $D = 1.0 \times 10^{-6}\text{cm}^2\text{sec}^{-1}$ is prospoed as the approriate value for the homogeneous diffusion in the previous study [7], the accurate value is still unknown and it is possible that the attaching the fold to the cleft affects the procedure to determine the appropriate value of D. Hence the whole analysis in this study is performed in the range of D between $(0.5 \sim 2.0) \times 10^{-6}\text{cm}^2\text{sec}^{-1}$.

Effects of junctional folds on the response characteristics

To discretize the partial defferental equation governing the RD system, the subdivision numbers of N_t on the transverse coordinate and N_r on the radial coordinate of the disc as well as the radius w_c of the disc are the parameters to be chosen for optimal representation of the behavior of the model. The radius w_c of the disc is defined as the extent of a quantal packet of ACh to generate the MEPC, and referred to critical radius [2]. The minimal compartment model with $w_c = 500$nm, $N_t = 3$ and $N_r = 10$ chosen [7] sufficiently represents the dynamic behavior of the RD system for the chemical transmission process in the synaptic cleft without the junctional fold, which then is called the foldless case in this study.

These parameters are determined as the same manner in the previous study [7] for the RD system of the synaptic cleft with the junctional fold. The cylinder representing the junctional fold is divided into the compartments by the same scale as in the disc, that is, $N_t \times d_f/d_c$ and $N_r \times w_f/w_c$ on the transverse and the radial coordinates, respectively. The applicability of the model is evaluated with reference to the relative variation in the total number of R_o due to parameter change from P_1 to P_2, i.e., by a quantity,

$$V_r = \int \frac{|C(t; P_1) - C(t; P_2)|}{C(t; P_1)} dt \tag{7}$$

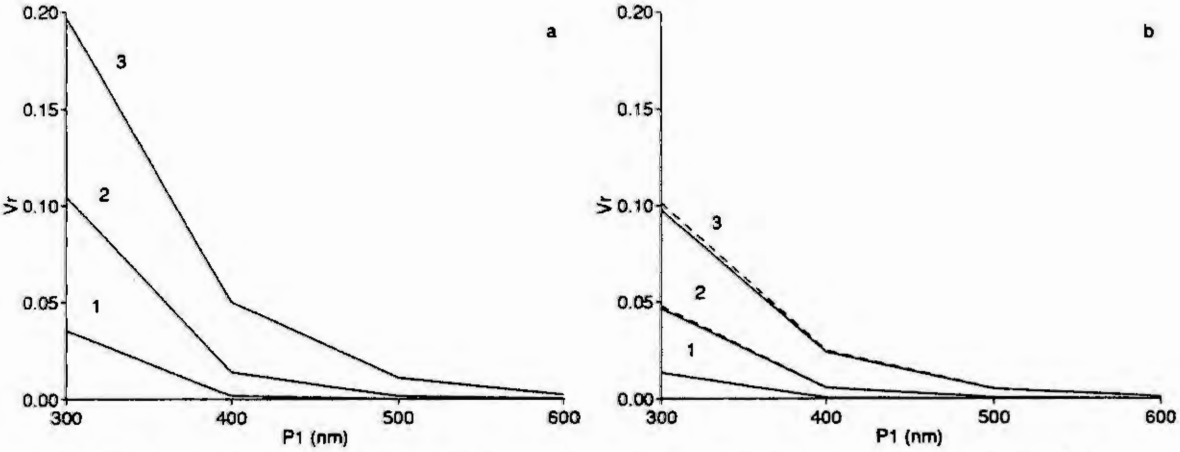


Fig.2. Effect of the critical radius of the disc on the relative variation in the total number of R_o. a: foldless ($w_f = 0$); b: junctional fold with the radius of $w_f = 50$nm and the depth of $d_f = 500$nm (solid lines) or $d_f = 1000$nm (broken lines). The variation V_r is evaluated for increase by 100nm in each of the radius w_c (in nm) of 300, 400, 500 and 600 indicated on the abscissa. The number on a curve indicates the case for the following value of D (in 10^{-6}cm^{2}sec^{-1}): 1: 0.5, 2: 1.0, 3: 2.0.

where $C(t; P)$ represents $C(t)$ for the parameter P. The value of the parameter is determined to be P_1 when the value of V_r becomes less than a certain tolerance against the various diffusion coefficients and the different junctional folds (cylinders) with the radius ($w_f = 0$ (foldless), 50nm or 100nm) and the depth ($d_f = 500$nm or 1000nm).

Figure 2 shows the behavior of the variation V_r with increase by 100nm in w_c (i.e., $P_2 = P_1 + 100$) at every 100nm for w_c (as P_1) between 300nm and 600nm. The attachment of the cyliner to the disc decreases the value of V_r almost by half regardless of the depth of the cylinder. In all of the junctional folds, the values of V_r are lower than 0.03 with the parameters of the minimal compartment model for the foldless, concluding that the model is applicable to the RD system for the synaptic cleft with the junctional fold.

The simulation analysis with the kinetic parameters in eq.(6) and the various diffusion coefficients is performed to examine the effects of the junctional folds on the behavior of $C(t)$ (i.e., equivalent of the

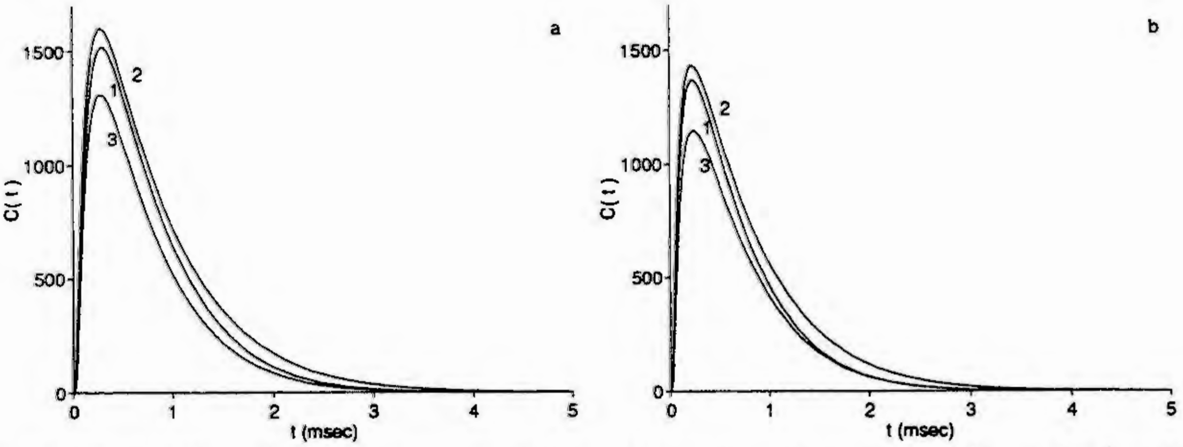


Fig.3. Effect of the radius of the junctional fold on $C(t)$. a: $D = 1.0$; b: $D = 2.0$ (in 10^{-6}cm^{2}sec^{-1}). The number on a curve indicates the case for the following value of the radius (in nm): 1: $w_f = 0$ (foldless); 2: $w_f = 50$; 3: $w_f = 100$.

Table 1 Effects of the junctional fold on the characteristic parameters of the MEPC.

w_f (nm)	d_f (nm)	D	C_{max}		t_m (μsec)		c (msec^{-1})	
0	(foldless)	0.50	1560	(1.00)	144	(1.00)	1.00	(1.00)
		1.00	1520	(1.00)	105	(1.00)	1.14	(1.00)
		2.00	1370	(1.00)	81	(1.00)	1.27	(1.00)
50	500	0.50	1660	(1.07)	124	(0.86)	0.95	(0.95)
		1.00	1600	(1.05)	94	(0.90)	1.05	(0.92)
		2.00	1440	(1.05)	76	(0.94)	1.14	(0.89)
	1000	0.50	1660	(1.07)	124	(0.86)	0.96	(0.95)
		1.00	1560	(1.05)	94	(0.90)	1.08	(0.95)
		2.00	1430	(1.04)	75	(0.93)	1.19	(0.93)
100	500	0.50	1390	(0.90)	113	(0.79)	1.12	(1.11)
		1.00	1320	(0.87)	91	(0.87)	1.19	(1.04)
		2.00	1150	(0.84)	78	(0.96)	1.22	(0.96)
	1000	0.50	1390	(0.90)	113	(0.78)	1.13	(1.12)
		1.00	1310	(0.86)	90	(0.86)	1.22	(1.07)
		2.00	1140	(0.83)	76	(0.94)	1.30	(1.02)

The value in parenthesis indicates the ratio to the value in the foldless case with the corresponding diffusion coefficient D (in 10^{-6}cm^{2}sec^{-1}).

MEPC) by variation in the values of the radius (w_f = 0, 50nm and 100nm) and the depth (d_f = 500nm and 1000nm) of the cylinder. The responses of $C(t)$ to a quantal release of ACh are demonstrated in Fig. 3. The maximum concentration (peak) is attained around 0.5msec after the release of ACh, and all the channels close within about 4msec. The curve of $C(t)$ with w_f = 50nm is always above the curve for the foldless case, which in turn is always above the curve with w_f = 100nm, regardless of the values of the diffusion coefficient D.

The response may be characterized quantitatively with the amplitude C_{max} (maximum value of $C(t)$), growth time t_m (time for $C(t)$ to increase from 20% to 80% of C_{max}) and decay constant c (reciprocal time constant for exponential decay of $C(t)$). The effects of the junctional fold and the diffusion coefficient on the characteristic parameters are demonstrated in Table 1.

Attachment of the cylinder with w_f = 50nm to the disc results in higher C_{max}, steeper concentration gradient and slower fall of the peak for whole the range of the diffusion coefficient examined, while the larger radius of the cylinder (w_f = 100nm) makes the amplitude C_{max} lower and the fall of the peak quicker for the diffusion coefficient D less than 1.0 × 10^{-6}cm^{2}sec^{-1}. The higher value of C_{max} with w_f = 50nm than with w_f = 100nm and 0 (foldless) indicates that the maximal amplitude of the MEPC may be attained with an optimal width of the junctional fold. The depth of the fold has less distinctive effects on $C(t)$ than the radius.

The effects of the diffusion coefficients on the response of $C(t)$ that increase in D reduces all of C_{max}, t_m, and 1/c (decay time) are not modified with the junctional folds.

Discussion

The minimal compartment model proposed previously [7] is applied to the analysis of the effects of the junctional fold on sponteneous generation of the MEPC. The simulation analysis reveals that the junctional fold causes the substantial effects, which might be relevant to the phenomenon of the notable spread of the growth time t_m [11]. It is also found that the width of the junctional fold has more distinctive effects than the depth, implying that the radial diffusion process of ACh has more distinctive effects on the MEPC than the transverse process as observed in the previous study [7]. The Thomas' model [2] in which the depth of the junctional fold is represented with the thickness of the disc (d_c in this study) is not satisfactory because the diffusion process in the radial direction is simplified as the simple efflux of ACh due to concentration gradient.

In the two-dimmensional compartment model the amplitude of the MEPC may be maximized at an optimal value of the width of the junctional fold, suggesting that the folds enlarge the reacting area of the postsynaptic membrane. Monte Calro procedure by the three-dimensional model with a junctional fold in actually folded shape indicates against the results in this study that the amplitude C_{max} and growth time t_m decrease but the decay constat c does not change with addition of more junctional folds [1]. The difference of the results might originate from the difference in the patemeter value set, especially the value for the diffusion coefficient $D = 6.5 \times 10^{-6}\text{cm}^2\text{sec}^{-1}$. It is also possible that the folded shape of the junctional fold, not cylinder as for the model in this study, may yield the different results. Extension of the two-dimensional minimal model to a three-dimensional model is required for full elucidation of the chemical transmission process, and is now under way at our laboratories.

Conclusions

The dynamic behavior of ACh in the synaptic cleft with a junctional fold is modelled with the two-dimensional compartment model, and analyzed by computer simulation to demonstrate that the junctional fold causes the substantial effects on spontaneous generation of the MEPC at the neuromuscular junction. Attachment of the junctional fold (cylinder of 50nm radius) to the disc results in higher C_{max}, steeper concentration gradient and slower fall of the peak. The larger radius of the cylinder (100nm) makes the amplitude C_{max} lower than the foldless case regardless of the values of the diffusion coefficient. The higher value of C_{max} with w_f = 50nm than with w_f = 100nm and 0 (foldless) indicates that the maximal amplitude of the MEPC may be attained with an optimal width of the junctional fold. The depth of the fold has less distinctive effects on $C(t)$ than the radius.

References

1. Bartol, T.M.J., Land, B.R., Salpeter, E.E. and Salpeter, M.M., Monte Carlo simulation of miniature endplate current generation in the vertebrate neuromuscular junction. Biophysical Journal, 59 (1991), 1290-1307.

2. Friboulet, A. and Thomas, D., Reaction-diffusion coupling in a structured system: application to the quantitative simulation of endplate currents. Journal of Theoretical Biology, 160 (1993), 441-455.

3. Hayashi, K. and Sakamoto, N., Dynamic Analysis of Enzyme Systems. JSSP/Springer-Verlag, Tokyo/Berlin, 1986.

4. Land, B.R., Harris, W.V., Salpeter, E.E. and Salpeter, M.M., Diffusion and binding constants for acetylcholine derived from the falling phase of miniature endplate currents. Proceedings of the National Academy of Science of the United State of America, 81 (1984), 1594-1598.

5. Mathews, G.G., Cellular Physiology of Nerve and Muscle. Blackwell Scientific Publications, Palo Alto, 1986.

6. MathWorks, Simulink User's Guide. The MathWorks, Inc., Natick, MA, 1992.

7. Naka, T and Sakamoto, N., A two-dimensional compartment model for reaction-diffusion system of acetylcholine in the synaptic cleft at neuromuscular junction. BioSystems, 40 (1996), 111-121.

8. Rosenberry, T.L., Acetylcholinesterase. Advances in Enzymology, 43(1975), 103-218.

9. Rosenberry, T.L., Quantitative simulation of endplate currents at neuromuscular junctions based on the reaction of acetylcholine with acetylchoine receptor and acetylcholinesterase. Biophysical Journal, 26 (1979), 263-290.

10. Schiesser, W.E., The Numerical Method of Lines. Academic Press, San Diego, 1991.

11. Van der Kloot, W., The rise times of miniature endplate currents suggest that acetylcholine may be released over a priod of time. BioPhysical Journal, 69 (1995), 148-154.

BIOHEAT TRANSFER PROBLEM AND APPLICATIONS

†I.Lubashevsky and ‡V. Gafiychuk
† Moscow State University,
Leninsky hills, Moscow, 117234, Russia;
‡ Institute for Applied Problems of Mechanics and Mathematics, National Academy of Sciences,
3b Naukova str., Lviv, 290603, Ukraine.

Abstract. We develop a theory of heat transfer in living tissue when heating or cooling are strong enough so that the vascular network response to variations in the tissue temperature is essential and gives rise to substantial dependence of the blood flow rate on the nonuniform temperature distribution. A macroscopic model for heat transfer and temperature self-regulation in a living tissue domain, containing a single microcirculatory bed, is proposed. On the basis of the obtained macroscopic description mathematical models of hyperthermia, cryosurgery and formation and growth of a necrosis domain in living tissue due to local laser irradiation are developed. Heat transfer in living tissue contained a tumor is also analyzed. Characteristics of temperature distribution in living tissue under local strong heating are investigated numerically.

Bioheat transfer model

Mathematical modelling of bioheat transfer is a fundamental problem of modern biophysics on the one hand and on the other is very useful in the study and optimization of hyperthermia treatment ,cryosurgery processes, formation and growth of a necrosis domain in living tissue due to local laser irradiation and so on. For the theory of heat and mass transfer in living tissue one of the central issues is how to create good models that could describe these transport phenomena, at least on the mesoscopic level, in terms of certain physical fields.

For the last years a number of different approaches to describing heat transfer in living tissue have been proposed [1, 2, 3, 4, 5]. For a review, analysis, and criticism of these models see, for example, [6, 7, 8, 9]. All these models allow for various features of the bioheat transfer and each of them may be valid, at least at the qualitative level, under certain conditions. So taking into account the present state of the bioheat transfer theory it has been suggested to use for application the following generalized bioheat equation which combines the main models mentioned above :

$$c_t\rho_t\frac{\partial T}{\partial t} = \nabla(\kappa_{eff}\nabla T) - fc_b\rho_b j(T - T_a) + q_h. \tag{1}$$

Here T is the tissue temperature, T_a is the temperature of blood in large arteries of a systemic circulation, c_t, ρ_t are the density and heat capacity of the tissue, c_b, ρ_b are the same values for blood, κ_{eff} is the effective thermal conductivity, the cofactor f ranges from 0 till 1, q_h is the heat generation rate caused by metabolic processes and external power sources, and j is the blood flow rate, i.e. the volume of blood flowing through unit tissue volume per unit time. In this model the ratio κ_{eff}/κ of the effective and true thermal conductivities of the tissue and the cofactor f are phenomenological parameters. Concerning the form of equation (1) we note that it is also phenomenological one rather than a reliable result of averaging the microscopic equations governing the heat propagation in the tissue.

In order to find a more rigorous equation governing evolution of the tissue temperature one, first, should develope an averaging technique which can reduce accurately the corresponding microscopic equations of heat transfer in living tissue to a macroscopic equation (or equations). In this way it is necessary to keep in mind that living tissue is an active heterogeneous medium organized hierarchically. In fact, due to the vessel system being hierarchically organized blood flow distribution over the vascular network as well as over the tissue domain has to be characterized by strong correlations between different hierarchy levels and also by spatial correlations. Therefore, to describe the blood flow effect on heat transfer one should take into account the vascular network as a whole rather than consider vessels of different levels individually. So, the vascular network models dealing with living tissue phantoms containing infinitely long vessels or models where the effect of blood flow through different vessels on heat transfer is treated in the same terms cannot form the basis of the successive procedure of averaging the microscopic equations. The aforementioned averaging technique has been developed in part in monograph [11] (see also [12, 13]). In particular, it turns out that the obtained macroscopic equation for the tissue temperature contains an averaged blood flow rate $j_v(\mathbf{r})$ rather than true one $j(\mathbf{r})$, which becomes essential when the tissue temperature is nonuniform on spatial scales of order 1 cm. The averaged and true blood flow rates, $j_v(\mathbf{r},t)$ and $j(\mathbf{r},t)$ are related by the equation

$$j_v - \frac{\kappa}{c_t\rho_t L}\nabla^2 \ln j_v = j \tag{2}$$

where L is also a certain constant of order unity.

The next characteristic property of living tissue is its active response to temperature variations. Living tissue tries to remain its temperature within a certain vital interval $[T_-, T_+]$. Therefore, if a certain tissue domain is, for example, heated, the vessels supplying this domain with blood will expand and the blood flow rate will increase. In order to find specific relationship between the blood flow rate $j(\mathbf{r})$ and the tissue temperature field $T(\mathbf{r})$ one should, in principle, account for the temperature response of the vascular network as a whole. It should be noted that blood flow rate can increase locally by tenfold [10].Inside the normal tissue Q_n the thermoregulation leads to the time variations in the blood flow rate $j(\mathbf{r},t)$ governed by the equation

$$\tau\frac{\partial j}{\partial t} + j\Phi(T) = j_0 \quad \text{if} \quad \mathbf{r} \in Q_n. \tag{3}$$

Here τ is the delay time of the vessel response, $j_0 \sim j_{tm}$, is the blood flow rate in the normal tissue when $T = T_a$, and the function $\Phi(T)$ describing the vessel response is of the form

$$\Phi(T) = 1 - \frac{T - T_a}{\Delta} \tag{4}$$

Here the velue $\Delta$is the half width of the survival temperature interval.

This is, for example,typically the case during hyperthermia treatment of small tumors. Another result is the fact that the tissue response to temperature variations may be described, at least as a first approximation, in terms of local relation of the blood flow rate $j(\mathbf{r})$ and the value $T(\mathbf{r})$ of the tissue temperature at the same point $\mathbf{r}$ (3) . When there is a certain small domain in living tissue, for example, tumor response of its vessels to temperature variations as well as variations in concentration of O_2, CO_2, etc. is depressed. So, under a strong heating in normal tissue the blood flow rate can increase by tenfold, whereas in tumors it remains practically at the same level. Under ordinary conditions the blood flow rates in normal tissue and in a tumor can differ little in magnitude.

Mathematical model for temperature distribution in tissue domain containing a tumour during hyperthermia treatment

We suppose that the temperature response of the normal vessels is ideal whereas the vessels contained in the tumor domain Q_t do not respond to blood temperature at all. In addition, the resistances of the latter vessels are assumed to have such values that the distribution of the blood flow rate $j_t(\vec{r})$ over the tumor domain Q_t be of a given form when the tissue temperature coincides with the arterial blood temperature T_a. Besides, in what follows for simplicity we study only the case $j_t(\vec{r}) \geq j_0$.

Within the framework of the proposed model for the ideal self - regulation process equation (3) should be added and in the tumor domain $\vec{r} \in Q_t$ by expression

$$j = j_t(\vec{r}). \tag{5}$$

The system of equations (1), (2), (3) or (5) forms the desired description of heat transfer in living tissue with tumor.

1 Mathematical model of necrosis domain due to laser irradiation

Let us consider a mathematical model for the formation and growth of a small necrosis domain in living tissue due to local laser irradiation[14]. The model assumes that a laser beam is delivered to a small internal tissue region where due to laser energy absorption the temperature attains such high values (above 70 ^{0}C) that lead to immediate tissue coagulation, including blood coagulation. The heat diffusion into the surrounding tissue causes its further thermal coagulation, giving rise to growth of the necrosis domain. The coagulation is treated in terms of phase transition, i.e. it is assumed to occur when the tissue temperature exceeds a certain threshold value T_{cg}. which is justified, for example, concerning the collagen coagulation [15]. Inside the necrosis domain the heat diffusion is controlled only by thermal conduction, whereas the heat propagation into the surrounding tissue is governed by both its thermal conduction and blood flow causing effective heat sink. It should be noted that the blood flow affects considerably the temperature distribution in the tissue during laser-induced interstitial thermotherapy [16]. In addition we take into account that due to vessel response to local variations in the tissue temperature the blood flow rate can increase substantially, leading to nonlinearity of heat transfer.

In order to describe the growth of the necrosis domain the proposed model considers tissue as involving two regions: the necrosis domain where the blood flow rate is equal to zero and the living tissue responding to temperature variations by increasing locally the blood flow rate. Beside the model has to take into account that the blood flow rate in tissue becomes substantially nonuniform and the vessel response can be delayed.

Following the theory developed in [11], we consider heat transfer in tissue containing regions, where it is anomalous in properties, in terms of heat transfer in normal tissue, where the vascular network responds to an effective (seeming) tissue temperature rather than the real one. Keeping in mind the black spot model [11] we describe the tissue coagulation as follows.

The necrosis region is regarded as a certain domain Q_{cg} where hydrodynamic resistances of the vessels are infinitely great, thereby, the blood flow rate in this region is equal to zero:

$$j(\mathbf{r},t) = 0 \quad \text{if} \quad \mathbf{r} \in Q_{cg}. \tag{6}$$

The necrosis domain Q_{cg} is assumed to be small in comparison with the microcirculatory bed domain Q_0. For real vascular network vessels will exhaust their potentials of expanding as temperature becomes high enough. This causes the blood flow rate to attain certain large but finite values $j_{\max}$ in the region where the tissue temperature exceeds a certain value $T_{vr} < T_{cg}$. Taking the latter into account and using the black spot model we now specify the proposed model for the thermal coagulation.

In the necrosis domain Q_{cg} the tissue temperature evolves according to the conventional heat conduction equation for solids:

$$c_t \rho_t \frac{\partial T}{\partial t} = \kappa \nabla^2 T + q_h \tag{7}$$

where κ is the intrinsic tissue conductivity and q_h is the rate of heat generation due to the laser beam absorption. Inside the living tissue its temperature is governed by the equation (1)

At the interface $\Gamma = \partial Q_f$ between the necrosis domain and living tissue the temperature is assumed to be equal to T_{cg} and the heat flux has no jumps. In mathematical terms this means that the tissue temperature at the interface Γ meets the following boundary conditions

$$(\kappa_{eff} \nabla_n T)|_{\Gamma_+} = (\kappa \nabla_n T)|_{\Gamma_-} \tag{8}$$

and

$$T|_{\Gamma_+} = T|_{\Gamma_-} = T_{cg} \tag{9}$$

Inside the living tissue there are also two regions that are different in properties. Inside the first one, Q_{tm}, matching the tumor, the blood flow rate is assumed to be constant j_{tm}, which reflects the fact that in real tumors blood vessels cease responding to temperature variations and the blood flow rate does not considerably increase during tumor heating [10]. In other words

$$j(\mathbf{r},t) = j_{tm} \quad \text{if} \quad \mathbf{r} \in Q_{tm}. \tag{10}$$

Inside the normal tissue Q_n the thermoregulation leads to the time variations in the blood flow rate $j(\mathbf{r},t)$ governed by the equation (3)

The function $\Phi(T)$ describing the vessel response is of the form

$$\Phi(T) = \begin{cases} \epsilon + (1-\epsilon)\frac{T_{vr}-T}{T_{vr}-T_a}. & \text{if} \quad T < T_{vr} \\ \epsilon & \text{if} \quad T > T_{vr} \end{cases} \tag{11}$$

where the ratio $\epsilon = j_0/j_{\max}$ is a small parameter, $\epsilon \ll 1$.

At the interface Γ the normal gradient of the averaged blood flow rate the blood flow rate is set equal to zero

$$\nabla_n j_v|_{\Gamma+} = 0 \tag{12}$$

This condition reflects the fact that there is no blood flow through the necrosis domain.

Two boundary model for freezing of living tissue during cryosurgery treatment

Mathematical analysis of temperature distribution in living tissue during freezing is useful in the study and optimization of cryosurgical treatment. Propagation of the freezing front Γ is conventionally described in terms of the free boundary problem of the Stefan-type:

$$v_n \rho_t \mathcal{L} = -(\kappa_{eff} \nabla_n T)\ |_{\Gamma_+} + (\kappa \nabla_n T)\ |_{\Gamma_-} \tag{13}$$

$$T\ |_{\Gamma_+} = T\ |_{\Gamma_-} = T_f \tag{14}$$

where $\mathcal{L}$ is the latent heat of fusion, Γ_+ and Γ_- denote the boundaries of the freezing front on the living and frozen sides of the tissue, respectively, and T_f is the freezing temperature.

A more accurate description of heat transfer process in living tissue is obtained if one takes into account the fact that living tissue form an active, highly heterogeneous medium . Also, when the size of the frozen region of the tissue is small in comparison with the characteristic length of the blood vessels that directly control the heat exchange between the cellular tissue and blood, the heterogeneity of living tissue has a substantial effect on the heat transfer process. Therefore, equation (3) which models the blood flow rate in terms of a continuous field $j(\vec{r})$ has to be modified [2]. In the frozen region Q_f the tissue temperature evolves according to the conventional heat conduction equation for solids(7).

Inside the unfrozen living tissue there are two regions that are different in thermoregulation properties. In the first one, Q_{vr}, adjacent to the frozen domain Q_f the tissue temperature varies from T_f to T_{vr}, and the blood flow rate is constant and equal,for example, to

$$j = j_0 \frac{T_a - T_{vr}}{T_{vr} - T_f} \tag{15}$$

In the second region, where $T < T_{vr}$ the blood flow rate is related to the local value of the tissue temperature by the equation (3).

At the interface Γ_{vr} of these two domains conditions8,9 fulfilled due to the interface Γ_{vr} containing no heat sink. The averaged and true blood flow rates are related, as before, by the equation(1).

At the interface Γ_{vr} the averaged blood flow rate, as well as its spatial derivatives, is continuous and at the freezing front Γ,conditions (12)should fulfilled.

This model allows for not only phenomena caused by phase transition during freezing living tissue, but also characteristics of living tissue response to substantial cooling as well as nonlocality in heat exchange between the cellular tissue and blood. It should be noted that in spite of this two boundary model containing the collection of the above –listed equations within the framework of this model the temperature distribution can be analyzed not only numerically but also by analytical methods.

Concluding remarks

In the present paper we propose new mathematical model for heat and mass transfer in living tissue. Although this model contains the collection of equations within its framework the temperature distribution can be successfully analyzed numerically. In particular, we have numerically analyzed the growth rate of the necrosis domain due to thermal coagulation, depending on nonlinear properties of heat transfer, the forms of frozen domains due to cryosurgical treatment and peculiarities of hyperthermia treatment .

References

References

[1] Pennes,H.H. Analysis of tissue and arterial blood temperatures in the resting human forearm. J. Appl. Phys., **1**, 93, 1948, pp.93–122.

[2] Chen,M.M., and Holmes,K.R. Microvascular contributions in tissue heat transfer. Ann. N.Y. Acad. Sci., **335**, 1980, pp.137–154.

[3] Klinger,H.G. The description of Heat Transfer in Biological Tissue Annals of N.Y.Academy of Science **335** , 1980, pp.133–136.

[4] Lagendijk,J.I.W. A new theory to calculate temperature distributions in tissues or why the "bioheat transfer" equation does work. in Hyperthermia Oncology ed. by I.Overgrad (London Taylor & Francis 1984) pp.507–510.

[5] Weinbaum,S., and Jiji,L.M. A new simplified bioheat equation for the effect of blood flow on local average tissue temperature. Trans. ASME J. Biom. Eng. **107**, 1985, pp.131–139.

[6] Heat transfer in medicine and biology. Analysis and applications, edited by A.Shitzer and R.C.Eberhart (Plenum, New York, 1985), **1,2**.q

[7] Wissler,E.H. Comments on the new bioheat transfer equation proposed by Weinbaum and Jiji. Trans.ASME J.Biom.Eng. **109**, 1987, pp.226–233.

[8] Baish,l.W., Ayyaswamy,P.S., Foster,K.R. Heat transport Mechanisms in Vascular Tissues: A Model Comparison, Trans. ASME J. Biom. Eng., **108**, N11, 1986, pp.324–331.254–256.

[9] Shitzer,A., and Eberhart,R.C. in: Heat Transfer in Medicine and Biology, Anal. and Appl., edited by A.Shitzer and R.C.Eberhart (Plenumm, New York, 1985), **1**, p.137

[10] Song,C.W., Lokshina,A., Rhee,I.G., Patten,M., and Levitt,S.H. Implication of Blood Flow in Hyperthermia Treatment of Tumors. IEEE Trans. Biom. Eng., **BME-31**, 1984, N1, pp.9–15 and the references in it.

[11] Lubashevsky I.A., Gafiychuk,V.V., and Cadjan,A.G. Bioheat Transfer Problem (to be published).

[12] Lubashevskii,I.A., Gafiychuk,V.V. A simple model of self-regulation in large natural hierarchical systems. J.Env.Syst. - **23(3)**, p.281-289 (1995).

[13] Lubashevskii,I.A., Gafiychuk,V.V. Mathematical Modelling of Heat and Mass Transfer in Living tissue. in Proceedings of the IMACS Symposium on Mathematical Modeling, February **2–4**, 1994. Vienna, pp.356–359.

[14] Lubashevsky, I.A., Priezzev, A.V., Gafiychuk, V.V. Free boundary model for local thermal coagulation,Proceedings of BoOs 96 Conferences, v.2681, San Jose, California., February 1996.

[15] Jacques S.L. Laser–Tissue Interactions: Photochemical, Photothermal, and Photomechanical. Surgical Clinics of North America **72** (1992) n. 3, p.p.531–558.

[16] Roggan A and Müller G. Dosimetry and Computer Based Irradiation Planning for Laser–induced Interstitial Thermotherapy (LITT). in: Laser–induced Interstitial Thermotherapy, ed. by G. Müller and A. Roggan, SPIE Institute Series Vol. **IS13** (1995).

A DYNAMICAL MODEL FOR ALCOHOLIC FERMENTATION IN WINEMAKING WITH TEMPERATURE CONTROL

A. Paulo G. M. Moreira and J. L. Martins de Carvalho
FEUP - DEEC / ISR, Rua dos Bragas
4099 Porto Codex, Portugal, Fax: +351-2-2000808
email: amoreira@garfield.fe.up.pt, jmartins@garfield.fe.up.pt

Abstract. Models for the dynamics of wine fermentation are presented. Their parameters and structure are adjusted to a collection of experimental data. Particular attention is given to the evolution of the concentrations of sugars, alcohol and yeast population along the fermentation process. We analyse the variation of the model parameters for several initial sugar concentrations. The influence of the temperature and the thermal model of the tank are also analysed.

1. Introduction.

The control of the fermentation is of great importance in order to obtain wines of high quality and to minimize yearly quality fluctuations. Predicting the behavior of the fermentation is only possible with a sufficiently accurate mathematical model. Besides the work of Boulton [2], there are very few studies dealing with the mathematical modelling of wine fermentation, particularly batch fermentation.

Although a satisfactory model for all types of wine is very difficult to find, our experience has shown that reasonably accurate models, with a particular structure, can be found for wines with similar characteristics. A model even with restrictions is always helpful in understanding the dynamics of fermentation and the role of each parameter. The actuator normally used in the control of wine fermentation are control valves with a maximum of three states: cooling, warming or absence of control when the valve is closed. The mathematical model for the fermentation in these control conditions will be analyzed with the help of experimental data collected in a pilot scale plant.

2. Methods

2.1. Yeast cell mass growth, sugar consumption and alcohol production models

Following Boulton [2] the model for the growth of yeast population X is given by:

$$\frac{dX}{dt} = \mu X_v \qquad ,X_v = (1 - \frac{E\,t}{K_t})\,X \tag{1}$$

were X_v is the viable yeast population which decreases with time t, and with the alcohol concentration E. The substract (sugar) consumption and product (alcohol) production is given by:

$$\frac{dS}{dt} = -\frac{1}{Y_m}\cdot\frac{dX}{dt} - m\,X_v \qquad \text{and} \qquad \frac{dE}{dt} = -Y_{E/So}\frac{dS}{dt} \tag{2}$$

i.e. sugars are consumed at a rate proportional to the amount of viable yeast X_v and their growth, μ being the specific growth yeast cell coefficient, Y_m the maximum growth yield, m the specific maintenance rate and t the time elapsed since the beginning of the fermentation. The alcohol production is proportional to the sugar consumption.

2.2. Models for the specific yeast cells mass growth coefficient

The specific growth yeast cell coefficient depends on several factors: Substract concentration *S*, biomass concentration *X*, product concentration *E*, in our case, sugars, yeast and alcohol (ethanol), respectively, besides other factors such as temperature, PH, oxygen concentration, light intensity, etc. As such, the yeast cells mass growth coefficient will be given by the multiplication of a constant by a function of each of the above mention factors:

$$\mu = K\mu(S)\,\mu(X)\,\mu(E)\,\mu(T) \tag{3}$$

the remaining factors were considered non-relevant.

2.2.1. Influence of sugar and yeast cell mass concentration

The influence of sugars concentration is usually described by Monod's law as mentioned, amongst others, by Strehaiano [11]. High yeast cells concentrations have also an inhibiting effect on their growth. A model which translates simultaneously the sugars and yeast cells inhibiting effect was suggested by Verhust and is cited by Bastin [1].

$$\mu(S,X) = \frac{\mu^* S}{K_C X + S} \tag{4}$$

2.2.2. Influence of alcohol concentration

Alcohol concentration has also an inhibiting effect on yeast cells growth. We will use the model suggested by Boulton [2] and also referenced by Bastin [1].

$$\mu(E) = \mu^* e^{-KpE} \tag{5}$$

2.2.3. The influence of temperature

Temperature is a determining factor in wine fermentation. Very low temperatures may halt the fermentation and therefore must be avoided. Too high temperatures must also be avoided since wine contains volatile products such as flavors and wine compounds whose loss degrades wine quality. The rise of temperature begins by having a positive effect on the specific growth factor. However after a certain value it induces less favorable yeast growth conditions. The following expression is normally used:

$$\begin{aligned} &\mu(T) = a_1\exp(-E_1/RT) - a_2\exp(E_2/RT) - b && , T_1 \le T \le T_2 \\ &\mu(T) = 0 && , T < T_1 \text{ or } T > T_2 \end{aligned} \tag{6}$$

were R is the gas constant. In Fig. 1 we see the evolution of this function for E_1 = 16000 J/g.mole, E_2= 34500 J/g.mole, b = 0.037, $a_1 = 2.4 \times 10^{11}$, $a_2 = 1.02 \times 10^{24}$, T_1 = 273 K e T_2 = 320 K. The values were used by Bastin [1].

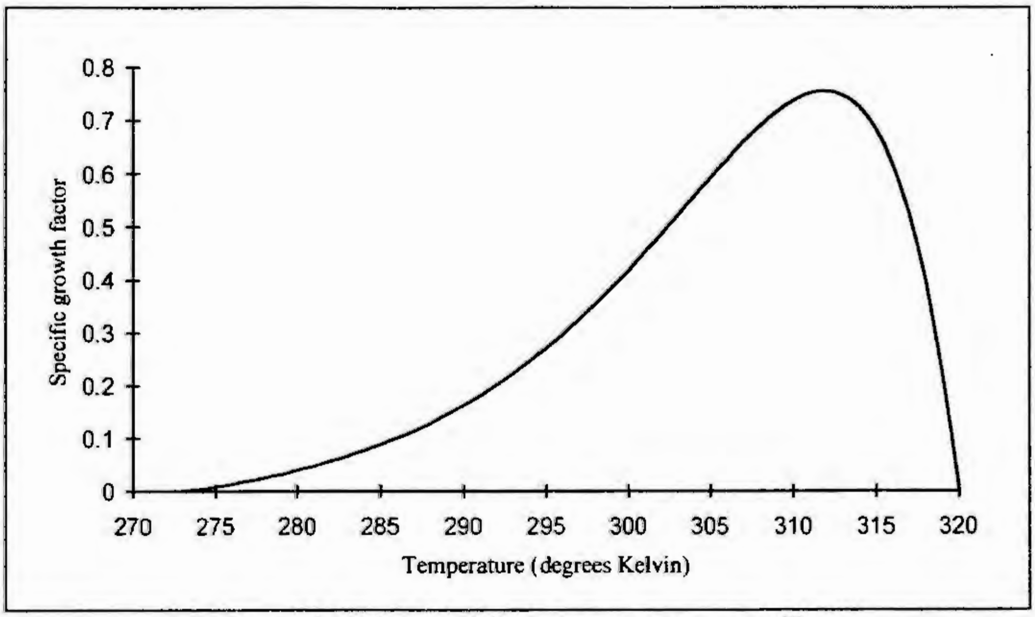


Fig. 1 - Specific growth factor versus temperature (T).

Besides its influence upon the yeast specific growth factor the temperature also influences the *m* parameter in equation (2)

$$m = m_0 \, EXP\left(\frac{-9000(T - 293.3)}{293.3 \, R.T} \right) \tag{7}$$

were T denotes the temperature in degrees Kelvin and R the gas constant.

The variation of parameter *m* with temperature has an effect opposite to the variation of the specific growth factor with temperature on the rate of sugar consumption. However, for typical values, the effect of the specific growth factor clearly dominates and the rate of sugar consumption grows with temperature between 279 °K and 312 °K. This range encompasses the range of recommended temperatures for wine fermentation, which is normally located between 10 °C (283 °K) and 35 °C (308 °K).

2.3 Thermal model for the tank

During the wine fermentation process the amount of generated heat (P_c) is related with the rate of sugar consumption. The must temperature depends also of the room temperature (T_{amb}), the temperature of the cooling/heating water, must density and type of fermenter. Boulton [2] presents a very simplified model. More detailed models are used by Sablayrolles and Barre [10] but both are first order models. During data collections performed at several winemaking industries we found that the models above did not explain some observed phenomena such as the further decrease in must temperature after closing the cooling water valve, cf. Moreira and Carvalho [5], [6]. There will be then a higher order dynamics since the jacket doesn't reach instantaneously the cooling/heating water temperature after the valve is open. Convection currents in the must also contribute this phenomena. In order to model the evolution of the must temperature (*T*) and water temperature in the jacket (T_j) we will use the follow equations when the valve is open:

$$T(k+1)=T(k)+P_c(k)+K_a[T(k)-T_{amb}(k)]+K_m[T(k)-T_j(k)] \tag{8}$$

where,

$$T_j(k) = P_f \, T_j(k-1) + (1-P_f) \, T_f^*(k) \tag{9}$$

and $T_f^*(k)$ is the cooling/heating water temperature $T_f(k)$ when the valve is open or the fermenter temperature T(k) when the valve is closed.

And the following model is used when the valve is closed:

$$T(k+1)=T(k)+P_c(k)+K_t[T(k)-T_{amb}(k)] \tag{10}$$

Equation (9) is a first order system with unit gain at low frequencies. Consequently $U_f(k)$ is always changing between $T_f(k)$ and $T(k)$. If we commute from equation (10) to equations (8) and (9) when the valve opens $U_f(k)$ will evolve gradually from $T(k)$ to the cooling/heating water temperature $T_f(k)$. In this way we model the temperature variation of the jacket. But in order to model this same phenomena when the valve closes we need to continue using equations (8) and (9) by a few more instants after the vale closes. After doing some tests we conclude that the best result is achieved when we use equations (8) and (9) for one more sampling period (10 minutes) after valve closure.

2.4. Simulation methods and adjustment of the parameters to the experimental data

The model presented in the previous sections was employed in the simulation of fermentations at constant temperature. The computer simulation was carried out by means of a 5th-order Runge-Kutta method with adaptive step size, as shown by Press *et. al.* [9] with some adaptions performed by the authors. The experimental data was obtained from the work of Haloui *et all.* [4] which describes several fermentations at a laboratory scale (15 litres fermenters) with temperature control (30 °C ± 0.1 °C) and initial sugars concentration between 160 g/l and 260 g/l. The initial yeast inoculation was 0.2 g/l.

The adjustment of the parameters to the experimental data was achieved by the minimization of a cost function with the help of the "Downhill Simplex" method, which can also be found in the above mentioned work. In a previous work [7] three different cost functions have been analyzed. However we found they had the tendency to yield, in the beginning of the fermentation, a large error in sugar concentrations. This can be explained by the fact that the yeast and alcohol concentrations are small in the early stages of the fermentation, therefore giving rise to large relative errors more visible in the sugar concentration that has higher absolute values in this stage. On the other hand it is sensible to give more weight to measurements with larger values since these are more accurate. Consequently an additional factor *P* is introduced which acts as an increasing weighting factor. Round(X) denotes the integer nearest to X. In this work the following cost function as used

$$F=\frac{1}{N}\sum_{i=1}^{N}[(\frac{S^*(t_i)-S(t_i)}{S(t_i)}(P.\mathrm{Round}(S(t_i)/100)+1))^2+$$
$$+(\frac{X^*(t_i)-X(t_i)}{X(t_i)}(P.\mathrm{Round}(S(t_i)/10)+1))^2+(\frac{E^*(t_i)-E(t_i)}{E(t_i)}(P.\mathrm{Round}(S(t_i)/100)+1))^2 \tag{11}$$

were the asterisk denotes values obtained by simulation. As *P* increases greater weight is given to the measurements with higher values.

For the thermal model the simulation is discrete. The sampling period was 10 minutes. We estimate $P_c(k)$ by a simple recursive least squares algorithm. A forgetting factor of 0.992 has performed well with the dynamics of the process in study. The other parameters are assumed constant and are obtained by minimizing the following cost function:

$$F(K_a, K_m, K_t, P_{c0}, P_f)=\sum_{i=1}^{N}(\Delta T(k)-\Delta\hat{T}(k))^2 \tag{12}$$

where $\Delta T(k) = T(k+1) - T(k)$ and $\Delta\hat{T}(k)$ is the value given by the model. The parameter P_{c0} is the initial value of $P_c(k)$.

3. Results

The obtained values can be seen in table 1 for a value of $P = 9$. The influence of P begins to decrease as P increases. The efficiency of alcohol production with respect to the initial sugars concentration ($Y_{E/So}$) was previously computed by dividing the final alcohol concentration by the initial sugars concentration.

So	YE/So	K	KM	Kp	Kt	m	Ym	F
160	0.431	0.219	10.2	≈0	9146	0.0472	0.0573	1.103
184	0.420	0.502	≈0	0.0398	4506	0.782	0.0779	0.672
205	0.463	0.508	59.6	≈0	5625	1.137	0.114	0.663
230	0.437	0.423	45.4	0.0050	7734	0.708	0.0740	1.364
260	0.419	0.386	16.2	0.0216	10046	0.720	0.0652	0.968

Table 1 - Values obtained for the parameters of the model in the minimization of cost function (11) with $P = 9$.

As shown in the F column the obtained approximations were rather satisfactory. As an illustration we show in Fig. 2 the results for an initial sugar concentration of 205 g/l. The differences between $P = 3$ and $P = 6$ are much more significant than for $P = 6$ and $P = 9$.

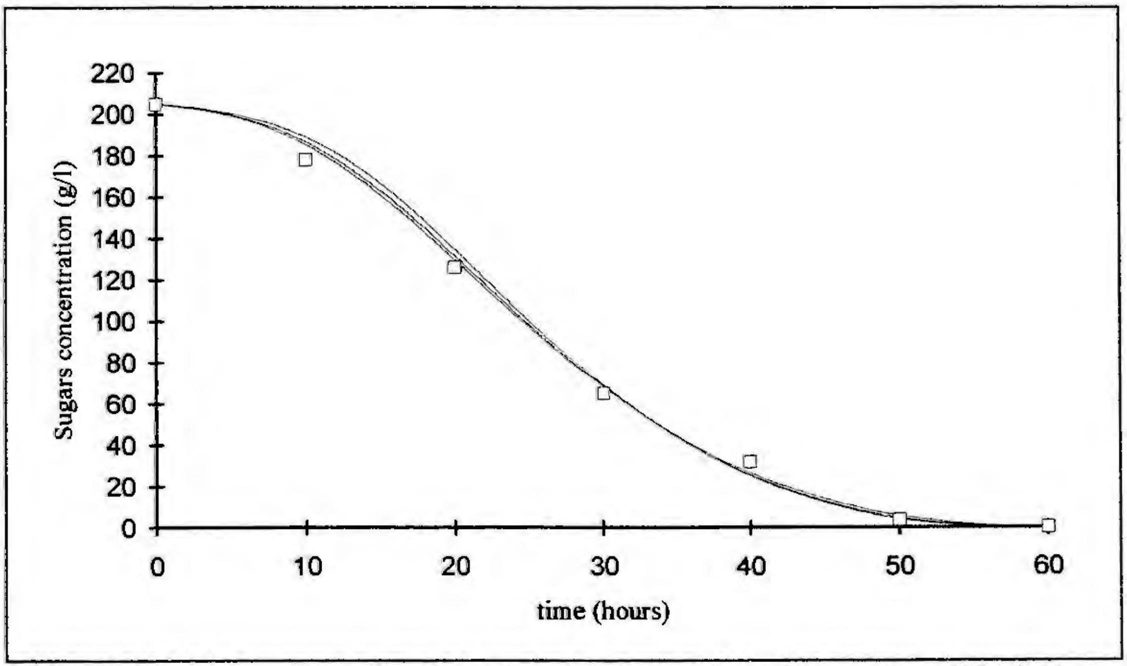


Fig. 2 -Sugars concentration versus time obtained via (11) for P=3, P=6, P=9 and S_0 = 205 g/l. The squares represent the experimental values.

For the thermal model the results obtained are illustrated in table 2 and table 3. The model revealed to be well adapted to the must temperature evolution collected in industrial winemaking plants.

K_a	-0.002319
K_m	-0.01446
K_t	-0.002191
P_{c0}	0.01046
P_f	0.3888

Table 2 - results for the model given by equations (8), (9) and (10)

average of $(\Delta T(k) - \Delta \hat{T}(k))^2$	0.000140
average of $\Delta T(k) - \Delta \hat{T}(k)$	-0.00192
variance of $\Delta T(k) - \Delta \hat{T}(k)$	0.000142
max. abs. value of $\Delta T(k) - \Delta \hat{T}(k)$	0.0716

Table 3 -characterization of the approximation with the model given by equations (8), (9) and (10)

4. Conclusions and future developments

A simulation for alcoholic fermentation in enological conditions was presented, together with a set of values for its parameters. The values obtained for the parameters still lack validation in terms of physical significance. In any case, the good adjustment between simulated and experimental data allow the use of the proposed model in the development of controllers for batch fermentation processes.

The consideration of other models, namely empirical models with a smaller number of parameters may be easier to identify, despite the fact that their parameters have no physical significance. This approach was followed by Bove *et al.* [3}.

References

1. Bastin, G. and Dochain, D., On-line Estimation and Adaptive Control of Bioreactors, Elsevier Science Publishers, Amsterdam, 1990.

2. Boulton, R., The Prediction of Fermentation Behavior by a Kinetic Model. Am. J. Enol. Vitic., Vol. 31, N° 1 (19080), 40-45.

3. Bovee, J.P., Strehaiano, P., Goma, G. and Sevely, Y., Alcoholic Fermentation: Modelling Based on Sole Substrat and Product Measurement. Biotechnology and Bioengineering, 26 (1984), 323-334.

4. Haloui, N. El, Picque, D. and Corrieu, G., Mesures Physiques Permettant le Suivi Biologuique de la Fermentation Alcoolique en Oenologie. Sciences des Aliments, 7 (1987), 241-465

5. Moreira, A.P.G.M. and Carvalho, J.L.M., On the Estimation of the Rate of Heat Generated in Fermenters. In: Proc. of the IASTED International Conference Applied Modelling, Simulation and Optimization, Cancun, Mexico, 1995, 1-4.

6. Moreira, A.P.G.M. and Carvalho, J.L.M., Continuous density measurement in fermenters: a model based approach. In: Proc. of the 13th World Congress International Federation of Automatic Control, 1996, San Francisco, California, USA

7. Moreira, A.P.G.M. and Carvalho, J.L.M., A Dynamical Model for Alcoholic Fermentation in Winemaking, In: Proc. of the 2nd Portuguese Conference on Automatic Control, 1996, Porto, Portugal

8. Navarro, J.M. and Goma, G., Fermentation continue: théorie et applications à la préparation des boissons fermentées, Centre de Documentation Internationale des Industries Utilisatrices de Produits Agricoles, Serie Syntheses Bibliographiques, 11 (1976), pag. 27.

9. Press, William H., Flannery, Brian P., Tenkolsky, Saul A. and Vetterling, William T., Numerical Recipes - The Art of Scientific Computing (Fortran Version). Cambridge University Press, 1989.

10. Sablayrolles, J.M. and Barre, P., Pilotage Automatique de la Température de Fermentattion en Conditions Oenologiques. Sciences de Aliments, 9 (1989), 239

11. Strehaiano, P., Uribelarrea, J.L., Mota, M., and Goma, G., Observations sur les Mecanismes d'Inhibition en fermentation Alcoolique. Rev. Franc. Oenol, 97 (1985), 55-60.

OPTIMAL MACROECONOMIC POLICIES FOR AUSTRIA: A SIMULATION ANALYSIS

R. Neck[1] and S. Karbuz[2]
[1]Department of Economics, University of Osnabrueck
D-49069 Osnabrueck, Germany
[2]UTESAV, Mecidiye Cad. 7/50 Mecidiyeköy, Istanbul, Turkey

Abstract. In this paper, we determine optimal monetary and fiscal policies for Austria using a small macroeconometric model. We use a Keynesian model of the Austrian economy, called FINPOL1, estmated by ordinary least squares, which relates the main objective variables of Austrian economic policies to fiscal and monetary policy instruments. Optimal macroeconomic policies are calculated for the model under a quadratic objective function using the algorithm OPTCON. Several control experiments are performed in order to assess the influence of different assumptions about the weights in the objective function and about stochastic parameters on optimal budgetary and monetary policies.

Introduction

One aim of macroeconomic modelling is to provide an interpretation and explanation of aggregate economic behavior by using a set of macroeconomic variables that are linked by mathematical relationships in the form of a numerical linear or nonlinear model. Then quantitative statements concerning the analysis of alternative policies over some time period can be made. Since the 1930s, the use of economic models for analyzing problems of macroeconomic policy formation has increased and even become international. Attempts have been made at using the interrelationships between macroeconomic variables with a view to forecasting the effects of policy decisions.

In Austria, the size of the federal budget deficit has been of much concern to policy-makers since the mid-eighties. In general, they now agree upon the necessity of consolidating the federal budget to prevent a loss of credibility of fiscal policies. Nevertheless, there are trade-offs and side-effects associated with a policy of gradually or even suddenly diminishing the budget deficit. In addition, the interactions between fiscal and monetary policies have to be considered in order to obtain valid conclusions about macroeconomic policy effects on the Austrian economy. So far, there is not sufficient quantitative information available about the effects of budgetary measures on the main objectives of Austrian economic policy, such as growth, full employment, price stability, and balance-of-payments equilibrium. Moreover, neither the intertemporal trade-offs nor the issue of policy-makers' limited information about future events has received serious attention in the political debate in Austria so far. In order to determine whether fiscal and monetary policies may have countercyclical effects, we build a model, called FINPOL1, which analyzes the influence of variables of the federal budget and of money supply on other macroeconomic variables.

Due to the stochastic nature of aggregate economic variables and their relations, we might say that our model is only an imperfect representation of the reality. Given this stochastic nature, the question of how to design optimal fiscal and monetary policies can be seen as a typical problem of quantitative economic policy. An optimal policy is one that minimizes an objective function without violating given constraints. Stochastic optimum control theory provides a consistent framework for processing and integrating the policy maker's objectives and constraints into a policy strategy over some time period while taking into account the dynamics of the economic system. If we postulate an objective function to be optimized by policy-makers, we can apply stochastic control theory to derive and analyze optimal macroeconomic policies for past or future periods. This is the aim of the present paper. In the next section, we describe the macroeconometric model FINPOL1. Afterwards, we discuss stochastic optimum contol theory and a stochastic control algorihm, OPTCON. We then report about the results of some control experiments under certainty and under uncertainty. Finally, some policy interpretations conclude the study.

FINPOL1: A small-scale econometric model of the Austrian economy

The model FINPOL1 is based on traditional Keynesian macroeconomic theory in the sense of conventional IS-LM/aggregate demand-aggregate supply models. Stochastic behavioral equations for the demand side include a consumption function, an investment function, an import function and an interest-rate equation as a reduced-form money market model. Prices are largely determined by aggregate demand variables. Disequilibrium in the labor market, as measured by the excess of unemployed persons over vacancies, is modelled to depend on the real GDP growth rate and the rate of inflation, embodying both an Okun's law-type relation and a rudimentary Phillips curve. The main objective variables of Austrian economic policies, such as real GDP, the labor market disequilibrium variable (related to the rate of unemployment), the rate of inflation, the balance of payments and the ratio of the federal net budget deficit to GDP, are related directly or indirectly to those fiscal and monetary policy instruments which are used as control variables, in particular to federal budget expenditures and revenues.

The model, which is dynamic and nonlinear, was estimated by OLS using annual data over the period 1965 to 1988. The estimates and test statistics together with simulation results suggest that the model is not far off the mark as a framework for policy analysis for Austria. For lack of space, details of the model cannot be discussed here; see [3] and [4] for the model equations and additional simulation and optimization experiments.

The optimization methodology

The theory of economic policy mostly assumes that the effects of policy instruments on endogenous variables are known with certainty, e.g. [5]. But things are different when considerations of risk are added. The essential point is to recognize both the errors in the regression coefficients in the macroeconomic model and the additive errors in the equations. "With coefficient uncertainty, big doses of policy medicine enlarge the variance of outcomes ... the standard error of the regression (or reduced form) forecast of a policy-objective variable is positively related to the distances of the regressors from their mean values in the sample data on which the model was estimated. Policy makers do not like to move in big steps" [6]. Macroeconomic policy problems can be viewed as involving the optimization of an intertemporal objective function by a decision-maker who is constrained by a dynamic system subject to various kinds of uncertainties. What we need, therefore, is stochastic optimum control theory, an approach to optimal policy design that can handle these different kinds of uncertainties [1].

Stochastic optimum control problems with nonlinear dynamic systems are usually complex, hence only numerical solutions under some simplifying assumptions can be obtained for particular values of the parameters. Even then, in most cases only approximations to the true optimum solution can be found. Here we use the algorithm OPTCON developed by Matulka and Neck [2]. OPTCON is an algorithm which determines approximate solutions of stochastic optimum control problems with a quadratic objective function and a nonlinear multivariable dynamic model under additive and parameter uncertainties. The objective function is quadratic in the deviations of the state and control variables from their respective desired values, but can easily be transformed to a general quadratic form. Symmetry of the possibly time-varying weight matrices of the objective function is assumed without loss of generality. The dynamic system is required to be given in a state space representation. Apart from the additive error term, a constant vector of unknown (and hence stochastic) parameters enters the system equations. Parameter and additive disturbance vectors are assumed to be independent with known expectations and covariance matrices. Although the dynamic system is formulated as a first-order system of difference equations, longer lags can be easily handled by augmentation of the state vector. As input for the algorithm, the user has to supply the system function, the initial value of the state vector, a tentative path for the control variables, the expected value and the covariance matrix of the stochastic parameter vector, the covariance matrix of the additive system noise, the weight matrices of the objective function, and the desired paths for the state and control variables.

Optimal macroeconomic policies under certainty

We can now apply the stochastic control algorithm OPTCON to our macroeconometric model FINPOL1. The purpose is to obtain tentative knowledge about the conduct of optimal budgetary and monetary policies in the eighties, during which the problem of rising federal budget deficits was much debated among politicians and commentators in Austria. Although the model is relatively simple, optimum control experiments may at least yield some informations about the policy trade-offs inherent in the model. Moreover, comparisons between fiscal and monetary policies delivered as "optimal" by using the model and actual ones and their effects on other objective variables may provide a first step towards evaluating historical policies within the framework of the theory of quantitative economic policy. Finally, introducing random parameters in the calculations shall shed some light on the influence of uncertainty on optimal policies within our model.

In all the experiments described below, we choose the planning horizon as 1981 to 1988, hence we are performing counterfactual experiments. Among the variables whose deviations from desired values are to be penalized, we distinguish two categories: First, there are five "main" objective variables which are of direct political relevance in assessing the performance of the Austrian economy. These are the rate of inflation ($PV\%_t$), the labor market excess supply variable (UN_t) as a measure for involuntary unemployment, the rate of growth of real GDP ($YR\%_t$), the current account (LBR_t), and the federal net budget deficit as percentage of GDP ($DEF\%_t$). In all experiments, 2% p.a. is considered as the desired rate of inflation ($PV\%_t$), 3.5% p.a. as the desired real growth rate ($YR\%_t$), and the desired levels for labor market excess supply (UN_t) and the balance of current account (LBR_t) are set equal to zero. For the deficit variable, we assume that the hypothetical policy-maker wants to consolidate the federal budget deficit gradually such that the desired value of $DEF\%_t$ is reduced by 0.3 percentage points each year, from 2.6% in 1981 to 0.5% in 1988.

Next, we introduce a category of "minor" objective variables. These include real private consumption (CR_t), real private investment (IR_t), real imports of goods and services (MR_t), the nominal rate of interest (R_t), real GDP (YR_t), real total aggregate demand (VR_t), real disposable income (YDR_t), the real stock of money supply ($M1R_t$), the domestic price level (PY_t), real public consumption (GR_t), real public-sector net tax revenues (TR_t), the price level of public consumption (PG_t), nominal total aggregate demand (V_t), nominal public consumption (G_t), nominal public-sector net tax revenues (T_t), and nominal GDP (Y_t), as well as the policy instrument (control) variables nominal stock of money supply ($M1_t$), federal budget net expenditures (NEX_t), and federal budget tax receipts (BIN_t). We assume 1980 historical values of these "minor" objective variables (except for R_t) to be given and postulate desired growth rates of 3.5% p.a. for the planning horizon for all real variables, desired growth rates of 2% p.a. for the price level variables, and desired growth rates of 5.5% p.a. for the nominal variables. The rate of interest R_t has a desired constant value of 7 for all periods.

In the weight matrix of the objective function, all off-diagonal elements are set equal to zero, and the main diagonal elements are given weights of 10 for the "main" objective variables and of 1 for the "minor" objective variables in most of the experiments. The state variables that are not mentioned above get weights of zero, thus being regarded as irrelevant to the hypothetical policy-maker. The above setting characterizes the input for the deterministic Experiment 1 below as well as for the stochastic experiments. Several alternative experiments will also be described briefly. We assume the weight matrix of the objective function to be constant over time.

Now we are in a position to perform some control experiments. To do this, we first assume all parameters of the model to be known with certainty. The only stochastic influences considered are the additive error terms in the behavioral equations, whose variances contribute to the optimal value of the objective function but do not affect the optimal policies as compared to a purely deterministic set-up. As these stochastics have no influence upon the optimal policies calculated, these experiments can be considered as "deterministic" ones. Both in the deterministic and in the stochastic experiments, we assume the values of the exogenous non-controlled variables to be known for all time periods in advance. Obviously, this attributes more information to the hypothetical policy-maker than any actual policy-maker could have had in 1980; this would have to be taken into account if comparisons of the results of the optimization runs to those of actual policies were attempted.

As a benchmark for comparisons, we report about the results of Experiment I in Table 2. For comparisons, Table 1 shows the historical values of the instrument variables and the values for the "main" objective variables which are obtained in the control solution, i.e. the historical dynamic simulation or the ex-post forecast with the instruments set at their actual values.

Table 1: Historical values of instruments and simulated values of "main" objectives

year	$M1_t$	NEX_t	BIN_t	$PV\%_t$	UN_t	$YR\%_t$	LBR_t	$DEF\%_t$
1981	165.635	315.292	287.791	6.680	1.642	-0.125	9.386	2.628
1982	170.964	347.560	300.954	3.906	2.917	-0.899	17.246	4.293
1983	192.941	382.243	316.673	2.817	3.697	0.462	22.800	5.775
1984	200.395	402.306	344.901	4.386	1.848	9.413	11.816	4.419
1985	201.421	433.014	372.895	4.240	1.543	4.776	7.663	4.220
1986	209.693	464.765	391.675	0.530	4.405	-5.949	-0.222	5.263
1987	227.636	479.356	409.556	1.334	5.274	-0.379	2.931	4.892
1988	247.586	517.824	451.343	3.267	2.679	11.541	-9.172	4.026

Table 2: Results of Experiment 1

year	$M1_t$	NEX_t	BIN_t	$PV\%_t$	UN_t	$YR\%_t$	LBR_t	$DEF\%_t$
1981	246.023	300.709	255.798	6.100	1.293	1.310	5.367	4.263
1982	246.837	337.237	249.769	3.160	1.942	1.801	3.917	7.869
1983	246.992	381.980	269.518	2.067	2.591	1.685	3.303	9.655
1984	246.264	353.769	333.305	3.324	2.291	4.546	7.061	1.630
1985	242.975	352.137	326.782	3.176	2.329	3.494	13.240	1.892
1986	242.565	439.445	276.722	0.000	3.511	-0.034	-13.664	11.801
1987	250.696	485.249	322.748	1.047	3.922	1.905	-24.742	11.244
1988	259.027	445.620	414.749	2.780	3.261	5.245	-19.876	1.967

Comparing the values in Tables 1 and 2, one can see that optimal fiscal policies are considerably more countercyclical than historical ones. Both federal budget expenditures (NEX_t) (except for 1987) and revenues (BIN_t) are lower than their historical values and react in a countercyclical way to changes in real GDP and its growth rate. In particular, NEX_t shows a strongly restrictive behavior during the boom years 1984, 1985, and 1988, and an expansionary behavior during the recession of 1986 and 1987; BIN_t acts in an expansionary way (has relatively low values) in 1981 and 1982, in addition. These fiscal policy instrument variables therefore fluctuate much more in this experiment than they did actually. In contrast, the monetary policy instrument ($M1_t$) jumps from its historical value of 160.662 Bill. AS in 1980 to about 246 Bill. AS in 1981 and stays at that level for most of the planning horizon. One reason for this (rather unrealistic) behavior of money supply is the attempt to raise investment expenditures by lowering the rate of interest during the earlier years of the planning horizon. The other endogenous variables of the model are affected primarily by the activities of budgetary policies. Labor market excess supply (UN_t) and real growth ($YR\%_t$) show smoother time paths than in the control solution. Inflation ($PV\%_t$) and the current account (LBR_t) have lower values than in the control solution, with the latter exhibiting deficits from 1986 to 1988. The deficit-to-GDP ratio ($DEF\%_t$) is higher than it was historically in most years (except for the boom years 1984, 1985, and 1988), especially during the recession 1986 - 1987. A similar countercyclical behavior is also exhibited by the public-sector variables G_t and T_t, which are directly influenced by budgetary policy instruments. The smoothing effect of budgetary policy on real GDP is also transmitted to most of the other demand-side variables of the model. By and large, the results of Experiment 1 are consistent with an optimistic Keynesian view of fiscal policy effectiveness to dampen recessions and booms, with monetary policy providing incentives for investment through lower interest rates.

Next, we investigate the influence of the weights given to the "minor" as compared to the "main" objective variables. Optimization experiments were conducted with the weights given to the "minor" objectives being altered between 1 and 10. As an example, consider the results of Experiment 2, where the "minor" objective variables get the weight 10 instead of 1 as in Experiment 1. This puts more emphasis on the goal of achieving "balanced" growth for all the variables in the model at the expense of equilibrating the "main" objective variables. As can be seen from Table 3, this modification calls for even stronger countercyclical actions of budgetary policies. Federal budget expenditures (NEX_t) are mostly (especially in the recession 1986 - 1987) higher, federal budget revenues are lower during the recession than in Experiment 1. Also money supply ($M1_t$) is still higher than in Experiment 1. This policy-mix results in higher values of real GDP and its components and also of the growth rate ($YR\%_t$); the effects on inflation ($PV\%_t$) and unemployment (UN_t) are small. The deficit variable $DEF\%_t$ is even higher in the recession, and the current account (LBR_t) exhibits larger deficits than in Experiment 1. Raising the weights given to the "minor" objective variables thus reinforces the Keynesian policy prescriptions; fluctuations of endogenous variables due to exogenous shocks (from the exogenous variables, which are mostly determined by developments in world markets) or to the dynamics of the model are kept down at the expense of higher fluctuations of the fiscal instrument variables.

Table 3: Results of Experiment 2

year	$M1_t$	NEX_t	BIN_t	$PV\%_t$	UN_t	$YR\%_t$	LBR_t	$DEF\%_t$
1981	248.338	300.214	258.568	6.059	1.356	1.085	6.165	3.964
1982	249.661	339.501	250.886	3.112	1.969	1.910	4.667	7.991
1983	250.158	383.612	269.537	2.011	2.602	1.742	3.918	9.818
1984	249.737	352.510	333.538	3.249	2.332	4.444	8.097	1.518
1985	246.757	351.749	325.056	3.105	2.332	3.629	14.027	2.000
1986	246.004	448.831	267.703	-0.010	3.295	0.779	-16.030	13.086
1987	252.524	494.084	314.564	1.057	3.683	2.112	-28.938	12.347
1988	259.400	450.318	410.843	2.785	3.158	4.879	-23.416	2.509

Optimal macroeconomic policies under uncertainty

In addition to the deterministic optimization runs, we also introduce different assumptions about parameter uncertainties in order to assess the influence of various kinds of uncertainty on the design of optimal budgetary policies during the eighties. In particular, we want to investigate the effects of making several key parameters determining fiscal and monetary policy multipliers uncertain. All these stochastic experiments start from the specification of the objective function assumed in Experiment 1. Thus the results of the following stochastic experiments can be compared directly to those given in Table 2 for the deterministic Experiment 1 and to those obtained from the control solution given in Table 1.

One difficulty arises from the fact that our model FINPOL1 was estimated by OLS instead of simultaneously. This means that we do not have an estimate of the full covariance matrix of the parameters, which is needed as input for stochastic control experiments with OPTCON. First, we assume this covariance matrix to be diagonal and select some of the diagonal elements of this matrix to be non-zero. The estimated values of the regression coefficients regarded as stochastic and their estimated standard deviations are used as expected values and standard deviations, respectively, of these parameters. This procedure amounts to neglecting correlations between stochastic parameters.

In our first stochastic experiment (Experiment 3), some parameters which are crucial for the monetary multiplier are regarded as stochastic in order to explore the consequences of the effectiveness of monetary policy becoming more uncertain. These are the coefficient of the real rate of interest (RR_t) in the investment equation and the coefficients of the real money supply ($M1R_t$) and of real GDP (YR_t) in the interest rate equation. Incidentally, these three coefficients happen to be insignificant, i.e. to have low t-values, thus providing an additional argument for emphasizing uncertainty about their numerical values. The results of Experiment 3 are given in Table 4. They show that optimal money supply now has a smoother path than in the corresponding deterministic Experiment 1. This can be interpreted to mean that as monetary policy has uncertain effects on real economic variables, it becomes optimal to use it less for stabilization purposes but to keep it more closely to its own desired path. Fiscal policies are affected by this kind of uncertainty, too: federal budget expenditures are lower and federal budget receipts are higher than in Experiment 1, resulting in a more restrictive, but still countercyclical course of budgetary policy. This reduces the values of the deficit variable $DEF\%_t$, but reduces also the growth rate ($YR\%_t$) and raises unemployment. The current account (LBR_t) is higher than in Experiment 1. Optimal policies in this case can be characterized as more cautious and less expansionary than in the deterministic case. The optimal value of the objective function, which was 145,949.3 in Experiment 1, becomes 241,182.5 in Experiment 3, reflecting additional costs due to the introduction of uncertainty.

Table 4: Results of Experiment 3

year	$M1_t$	NEX_t	BIN_t	$PV\%_t$	UN_t	$YR\%_t$	LBR_t	$DEF\%_t$
1981	217.323	298.876	260.156	6.321	1.374	0.955	6.260	3.678
1982	222.253	335.489	255.382	3.487	2.040	1.617	5.296	7.193
1983	225.659	379.585	277.992	2.434	2.705	1.485	5.086	8.678
1984	229.168	349.300	344.926	3.671	2.412	4.368	9.247	0.346
1985	232.485	346.006	340.481	3.453	2.448	3.368	15.766	0.408
1986	237.994	434.015	291.449	0.218	3.620	-0.117	-10.855	10.210
1987	248.870	478.936	338.950	1.186	4.018	1.857	-21.660	9.553
1988	258.578	433.080	429.170	2.844	3.308	5.357	-16.961	0.245

Experiment 4 takes the same coefficients as random as Experiment 3, but adds the marginal propensity to consume as a fourth stochastic parameter. This is the coefficient of real disposable income (YDR_t) in the consumption equation, which is important for the transmission of fiscal policy effects to demand-side variables, as is well known from Keynesian macroeconomic theory. In this case both monetary and fiscal multipliers are made uncertain, and we explore the effects of this uncertainty of demand-side policies on their optimal design. The results of this experiment, which are shown in Table 5, may be compared either to those of the deterministic Experiment 1 or to those of the previous Experiment 3. They show that there are considerable deviations from the previous experiment. Money supply ($M1_t$) now has lower values, especially in the earlier years of the planning period, and grows rather smoothly. Federal expenditures (NEX_t) and in particular federal revenues (BIN_t) are much higher than in previous experiments, and they are higher than they were historically. The federal budget deficit is lower than in the deterministic case, with surpluses created in 1981, 1984, 1985, and 1988. The results of this kind of macroeconomic policy are higher unemployment (UN_t), lower growth rates ($YR\%_t$), and high surpluses of the current account (LBR_t). One remarkable effect of the restrictive fiscal policy stance in 1981 is a policy-induced recession in that year (real GDP falls by more than 3%). This does not improve the performance of the economy in the following years, however, and in spite of quite large fluctuations of budgetary policy variables the cumulative loss of real output seems considerable, with real GDP being lower than in the deterministic case by more than 8% in 1988. One interesting point to note is the larger size of the public sector implied by this experiment, which nevertheless does not fulfill the task of stabilizing the economy very well. There is an enormous increase in the optimal value of the objective function to 1,992,725.2, which is more then eight times the value obtained in Experiment 3, showing the extremely high costs of uncertainty of fiscal policy effectiveness in this model where stabilization rests mainly on fiscal instead of monetary policy.

Table 5: Results of Experiment 4

year	$M1_t$	NEX_t	BIN_t	$PV\%_t$	UN_t	$YR\%_t$	LBR_t	$DEF\%_t$
1981	140.047	336.856	348.956	6.622	2.517	-3.362	20.220	-1.196
1982	156.591	395.957	318.072	4.025	3.231	0.655	26.123	7.293
1983	173.158	443.014	347.457	3.065	3.787	1.054	28.884	8.478
1984	189.196	411.563	426.269	4.198	3.399	3.978	35.402	-1.203
1985	204.433	408.016	432.068	3.809	3.385	2.902	44.506	-1.837
1986	221.134	498.686	388.372	0.479	4.490	-0.469	19.778	8.183
1987	241.010	545.707	447.197	1.231	4.855	1.470	11.449	6.990
1988	256.910	484.847	538.493	2.649	4.101	5.084	19.539	-3.518

This impression is confirmed by the results of Experiment 5, where we assume the coefficients relating to monetary policy to be known for certain and investigate the effects of making only fiscal policy multipliers uncertain. To do so, we take as stochastic parameters the marginal propensity to consume (as in Experiment 4) and the coefficients of federal budget expenditures (NEX_t) in the equation determining public consumption and of federal budget receipts (BIN_t) in the public-sector tax equation. The results of this experiment are given in Table 6. Although the variances of the parameters made stochastic now are low as compared to their estimated values, again there are considerable differences to the deterministic optimization results. Budgetary policy variables (NEX_t and BIN_t) show a similar pattern as in Experiment 4, but at a lower level of activity, resulting in a smaller size of the federal and the public sector. Money supply, on the other hand, has higher values than before, thus taking over part of the stabilization task of fiscal policy. The overall effect of this policy-mix on the "main" objective variables is similar as in Experiment 4. Again, in 1981 a recession is created by a budget surplus. Otherwise, the growth performance is slightly better than in the previous experiment, although the deficit-to-GDP ratio ($DEF\%_t$) is still lower. The optimal value of the objective function is still higher (2,168,342.1), showing again the deterioration of the optimally controlled system in the presence of parameter uncertainty affecting fiscal policy multipliers.

Table 6: Results of Experiment 5

year	$M1_t$	NEX_t	BIN_t	$PV\%_t$	UN_t	$YR\%_t$	LBR_t	$DEF\%_t$
1981	174.877	290.699	312.273	6.341	2.461	-3.085	19.663	-2.134
1982	187.965	342.739	294.385	3.621	3.105	1.040	24.668	4.538
1983	201.324	388.128	320.362	2.607	3.651	1.270	26.983	6.050
1984	212.572	356.102	390.257	3.776	3.197	4.400	32.387	-2.816
1985	220.089	352.842	393.383	3.477	3.154	3.202	40.401	-3.127
1986	229.520	442.883	352.195	0.197	4.324	-0.511	15.682	6.819
1987	245.239	485.794	402.905	1.039	4.713	1.521	7.197	5.972
1988	258.132	423.501	483.466	2.587	3.887	5.447	13.701	-3.982

Next, we want to learn how optimal policies look like when there is a high amount of uncertainty about the parameters of the model. To do so, in Experiment 6 we assume all the estimated parameters of the model (the coefficients and the constants) to be stochastic. Altogether, we now have 35 uncertain parameters in the model, whose variances (but not covariances) enter the determination of optimal policies. The results of this experiment are given in Table 7. They show that monetary and fiscal policies are now completely different from those in the deterministic Experiment 1 and also from those in previous stochastic experiments. Money supply ($M1_t$) starts at an extremely low level in 1981 (much less than half the historical value) and increases afterwards to reach approximately the value of Experiment 1 in 1988. Federal budget expenditures (NEX_t) are in most years below, federal budget receipts (BIN_t) in all years above the values of the deterministic case. Every year, a budgetary surplus is called for by this optimization experiment, which is of considerable amount in some years (more than 130 Bill. AS in 1981 and 1984). This combination of restrictive monetary and fiscal policies creates a severe recession in 1981, with real GDP falling by 10.6%. Also in the following years, real GDP and its components remain well below the values obtained in the deterministic experiment (and below the historical values). Unemployment (UN_t) is higher, and so are current account surpluses (LBR_t). In spite of the overall restrictive stance of budgetary policies, there is still considerable countercyclical variation in both fiscal policy instruments. Fiscal and monetary policies act in the same direction, enforcing a scenario of uneven, but generally low growth. It should be noted that both monetary and fiscal policies have time-paths which are quite different from those obtained in an analogous experiment with a smaller number of "minor" objective variables [4]. As in this study, however, there is no tendency for optimal macroeconomic policies to become less active with the introduction of parameter uncertainty, as one would expect given the results for the linear-quadratic case [7]. The optimal value of the objective function is again much higher in this than in any previous experiment (7,933,103.6), showing the high costs of uncertainty of all the parameters.

Table 7: Results of Experiment 6

year	$M1_t$	NEX_t	BIN_t	$PV\%_t$	UN_t	$YR\%_t$	LBR_t	$DEF\%_t$
1981	61.191	277.178	410.980	6.679	4.464	-10.610	44.236	-14.283
1982	88.782	353.358	368.224	4.348	5.113	-0.302	60.020	-1.511
1983	118.678	398.606	406.355	3.504	5.513	0.451	67.322	-0.747
1984	147.547	354.533	486.580	4.531	4.879	3.817	76.451	-11.705
1985	175.148	341.606	496.721	4.013	4.607	2.961	86.819	-12.799
1986	204.392	426.342	454.729	0.632	5.494	-0.399	62.048	-2.271
1987	236.128	454.473	479.702	1.273	5.213	3.284	48.449	-1.897
1988	260.360	388.021	510.000	2.785	2.918	10.573	38.162	-8.036

The results of Experiment 6 have been affected by the neglect of correlations between stochastic parameters. Although we do not have an estimate of the entire parameter covariance matrix for our model, we do have estimates of the covariances of parameter estimates within each OLS regression equation. In Experiment 7, we use these covariance estimates together with the variance estimates of Experiment 6. Our estimated parameter covariance matrix is hence no longer diagonal and takes into account some (though not all) correlations between different parameters. The results of Experiment 7 are given in Table 8. They show that stochastic optimization with some correlations between parameters being taken into account gives outcomes that are very different from those obtained from stochastic optimization without considering any parameter covariances (Experiment 6). In fact, the results for fiscal policy variables from Experiment 7 are rather close to those obtained from the deterministic optimization (Experiment 1); the path of money supply is close to that obtained in Experiment 3 with stochastic monetary policy multipliers. The value of the objective function is 173,516.1 in Experiment 7, that is not much higher than in the deterministic experiment. Thus, we can conclude that the costs of uncertainty of policy effects result mainly from high risk associated with parameter uncertainty without correlations between these parameters. If this risk is reduced by taking into account at least some covariances between parameters, optimal policies and minimum costs come close to those of the deterministic optimum.

Table 8: Results of Experiment 7

year	$M1_t$	NEX_t	BIN_t	$PV\%_t$	UN_t	$YR\%_t$	LBR_t	$DEF\%_t$
1981	228.636	303.497	259.673	6.243	1.314	1.196	5.558	4.157
1982	232.701	341.017	255.405	3.354	2.002	1.603	4.588	7.691
1983	235.511	387.528	276.266	2.281	2.647	1.615	4.109	9.517
1984	238.732	352.378	338.114	3.510	2.358	4.426	8.139	1.131
1985	239.827	348.660	336.130	3.300	2.425	3.319	14.939	0.930
1986	241.566	443.587	289.401	0.093	3.588	-0.040	-11.785	11.113
1987	249.731	488.441	333.966	1.099	3.997	1.853	-22.585	10.622
1988	258.384	440.160	423.064	2.793	3.325	5.241	-17.480	1.083

Concluding remarks

In this paper, we have applied the stochastic control algorithm OPTCON to the model FINPOL1, which was estimated for the Austrian economy. Optimal fiscal and monetary policies under certainty and under various kinds of uncertainties were numerically determined. It turns out that there may be considerable differences between optimal deterministic and stochastic policies, depending upon the particular parameters which are assumed to be uncertain. The costs due to additional uncertainty may be very high, especially when it affects fiscal policy multipliers. If correlations between parameter estimates are taken into account, on the other hand, the costs of uncertain policy effects are much smaller, and optimal policies are similar to the case where parameters are known for certain. A tendency for optimal policies to be applied in a more cautious way under uncertainty than under certainty, which is sometimes asserted in the economic policy literature, did not emerge.

The purpose of the present study should be seen within the methodology of simulation analysis: Optimal macroeconomic policies are investigated under alternative assumptions about the objective function and about the stochastics of the parameters for a given macroeconometric model. Obviously, modifying some equations of the model or using an alternative model would result in different optimal policy paths than those derived for FINPOL1. For instance, the strong countercyclical conduct of fiscal policies will certainly not be present in experiments with a model whose specification ist less Keynesian than that of the present one. Thus, a systematic analysis of the robustness of our conclusions about the influence of different objective functions and stochastic parameters with respect to different model structures will be the next aim of our research. Nevertheless, the results obtained so far seem to be interesting from the point of view of the theory of macroeconomic policy, because analytical results for models with stochastic parameters can be obtained only for extremely simple theoretical models; hence, a simulation methodology is the only one available to explore the sensitivity of optimal policies with respect to different assumptions about the objective function and about the amount of uncertainty with which the policy-maker is confronted.

Acknowledgement. Financial support from the Austrian "Fonds zur Foerderung der wissenschaftlichen Forschung" (research project no. P7966-SOZ) is gratefully acknowledged.

References

1. Kendrick, D., Stochastic Control for Economic Models. McGraw-Hill, New York, 1981.
2. Matulka, J., and Neck, R., OPTCON: An Algorithm for the Optimal Control of Nonlinear Stochastic Models. Annals of Operations Research, 37 (1992), 375 - 401.
3. Neck, R., Macroeconomic Effects of Austrian Budgetary Policies: Simulation Experiments with a Small Econometric Model. In: Cybernetics and Systems Research '92, (Ed.: Trappl, R.) World Scientific, Singapore, 1992, 973 - 980.
4. Neck, R., and Karbuz, S., Optimal Budgetary and Monetary Policies under Uncertainty: A Stochastic Control Approach. Annals of Operations Research, 58 (1995), 379 - 402.
5. Preston, A.J., and Pagan, A.R., The Theory of Economic Policy. Cambridge University Press, Cambridge, 1982.
6. Tobin, J., On the Theory of Macroeconomic Policy. Cowles Foundation Discussion Paper 931, Yale University, New Haven, 1989.
7. Turnovsky, S.J., Optimal Control of Linear Systems with Stochastic Coefficients and Additive Disturbances. In: Applications of Control Theory to Economic Analysis, (Eds.: Pitchford, J.D., and Turnovsky, S.J.) North-Holland, Amsterdam, 1977, 293 - 335.

The Evolution of Money and Genetic Algorithms

Sylvia Staudinger
Technical University Vienna
Argentinierstr.8/175, A-1040 Wien
email: sstaudin@pop.tuwien.ac.at

Abstract

This paper models a world of repeated Wicksell-triangles, which can be used to explain the medium of exchange function of money. An economy is modeled where money comes up because individuals maximize their expected utility. Individuals choose their trading strategies with respect to maximizing utility which depends on the storage costs of goods and the probabilities of finding a suitable trading partner. First, two types of Markov-Nash-equilibria are deduced, where different goods come up as commodity money, depending on the parameters of the model.

Then genetic algorithms are used to study how artificially intelligent agents learn to coordinate their strategies. The first question I deal with is: do individuals learn the optimal strategies which characterize these equilibria and if so, the second question is: how much time do individuals need to reach the equilibrium strategies.

One obtains that individuals only learn the strategies of the Markov-Nash-equilibrium if the equilibrium is the one with the lowest storage cost good as intermediate good, i.e. money. Otherwise the equilibrium is in general not reached and individuals prefer also the lowest cost good as medium of exchange, although this is not the optimal strategy.

1 Introduction

Money has three major roles in the economy: store of value, medium of exchange and unit of account. Models with an overlapping generations structure are helpful to describe the first item but they do not provide a satisfying explanation of the medium of exchange function. A better approach to explain this function of money is by minimizing the costs of trading or equivalently by maximizing the expected utility of individuals. In this paper following Kiyotaki and Wright [3] the costs are interpreted as storage costs of goods. Optimal trading strategies are deduced by minimizing these costs. Depending on storage costs and probabilities for finding a suitable trading partner, one can obtain two kinds of Markov-Nash-equilibria (in pure strategies) with different goods as commodity money. In one equilibrium called fundamental the good with the lowest storage cost becomes money, in the other equilibrium, the speculative, the good with the highest storage cost plays this role.

In the next step genetic algorithms (GA) are used to analyse the dynamics of the model if individuals with bounded rationality have to coordinate their strategies. It turns out that individuals always prefer fundamental strategies although the strategies leading to the speculative equilibrium would be optimal.

One interpretation of these results is that individuals with bounded rationality are myopic and therefore prefer lower costs in the present to a higher probability of finding a trading partner in the future.

The plan of the paper is as follows. Section 2 presents the model. Section 3 describes the setup for the GA simulation and presents the main results of the simulations. Section 4 recapitulates the main points of the paper.

2 The model

We consider a discrete time model with a continuum of infinitely long lived agents. These agents are divided in three different types, the probability to meet an individual of any type is $q = 1/3$. Each agent of type i (mod 3) produces good $i + 1$ (mod 3) and can consume only good i, which gives her utility u.

Furthermore each agent can store one unit of good j each period at cost c_j. Without loss of generality we assume $0 < c_1 < c_2 < c_3$.

Let us denote the probability that an individual of type i holds good j with $\pi_{ij}(t)$ and $\pi(t) = (\dots, \pi_{ij}(t), \dots)$.

$\tau^i_{jk}(t)$ denotes the strategy of an individual of type i and is defined in the following sense:

$$\tau^i_{jk}(t) = \begin{cases} 1 & : \quad \text{if the agent wants to trade good } j \text{ for good } k \\ 0 & : \quad \text{otherwise} \end{cases} .$$

Each rational agent will maximize her expected utility of good k given the vector $\pi(t)$ and the strategies of the other individuals; this optimal utility is described by the following value functions

$$V_{ik}(t) = \begin{cases} u + V_{i,i+1}(t) = u - c_{i+1} + \max_j \beta E[V_i, j(t+1)|i+1] & : \quad k = i \\ -c_k + \max_j \beta E[V_{ij}(t+1)|k] & : \quad k \neq i \end{cases} .$$

Following [3] and assuming $u > (c_j - c_k)/(1-\beta), \beta$ discount rate, yields $\max_k V_{ik} = V_{ii}$, i.e. an agent always trades a good for his consumption good.

It is obvious that $\tau^i_{jk}(t) = 1$ iff $V_{ik}(t) > V_{ij}(t)$ and $\tau^i_{jk}(t) = 1$ implies $\tau^i_{kj}(t) = 0$, i.e. $\tau^i_{jk}(t) = (1 - \tau^i_{kj}(t))$.

Since the endowment of type i agent at time $(t+1)$ depends only on the endowment in t, on $\pi(t+1)$ and the trading strategies in $(t+1)$ it follows a Markov process. Let us deduce the elements $p^i_{kj}(t)$ of the transition matrix P.

$$p^i_{kj}(t) = \frac{1}{3}\tau^i_{kj}(t) \sum_{l=1}^{3} \pi_{lj}(t-1)\tau^l_{jk}(t), \qquad \text{and } p^i_{kk}(t) = 1 - p^i_{kj}(t).$$

In the remainder of this section we deduce steady state Markov-Nash-equilibria. A steady state Markov-Nash-equilibrium is the set of all strategies $\{\tau^i\}$ which maximize the individual utility given π and the strategies of the other agents. Given $\{\tau^i\}$, π is the resulting steady state vector and π satisfies $\pi = P'\pi$. To work out the equilibria we use an arbitrary set of trading strategies and evaluate necessary and sufficient conditions such that these strategies maximize individual utility given π and the strategies of the other types and then calculate π and P given these strategies.

Fundamental equilibrium

We denote strategies

$$\begin{array}{llllll} \tau^1_{21} = 1, & \tau^1_{22} \in [0,1], & \tau^1_{23} = 0, & \tau^1_{31} = 1, & \tau^1_{32} = 1, & \tau^1_{33} \in [0,1], \\ \tau^2_{32} = 1, & \tau^2_{33} \in [0,1], & \tau^2_{31} = 1, & \tau^2_{12} = 1, & \tau^2_{13} = 0, & \tau^2_{11} \in [0,1], \\ \tau^3_{13} = 1, & \tau^3_{11} \in [0,1], & \tau^3_{12} = 0, & \tau^3_{23} = 1, & \tau^3_{21} = 1, & \tau^3_{22} \in [0,1]. \end{array}$$

fundamental strategies, i.e. each individual prefers to trade for the good with the lowest costs, except her consumption good. Fundamental strategies imply the following ranking of value functions: $V_{11} > V_{12} > V_{13}, V_{22} > V_{21} > V_{23}, V_{33} > V_{31} > V_{32}$. To show that these strategies lead to an equilibrium, which we call fundamental, we have to prove that they maximize utility and then calculate π and P.

Let us look at a representative agent of type 1 and suppose she holds good 2 at storage cost c_2. In the next period she meets an agent of type i with probability $q = 1/3$. If her trading partner is of type 1, no trade occurs and she leaves the market with discounted utility βV_{12}. If she meets type 2 this agent has good 1 with probability π_{21}. Both partners want to trade and type 1 consumes, produces one unit of good 2 hence her utility is given by $\beta(u + V_{12})$. On the other hand, with $\pi_{23} = 1 - \pi_{21}$ type 2 supplies good 3. Since type 2 wants to trade, trade occurs if $\tau^1_{23} = 1$ and utility of type 1 is $\beta[\tau^1_{23}V_{13} + (1 - \tau^1_{23})V_{12}]$. If type 1 meets type 3 no trade takes place due to the strategy of type 3 and type 1 has utility βV_{12}. Hence we have the indirect utility of good 2, given by the value function V_{12}:

$$V_{12} = -c_2 + \beta/3[V_{12} + \pi_{21}(u + V_{12}) + \pi_{23}(\tau^1_{23}V_{13} + (1 - \tau^1_{23})V_{12}) + V_{12}]. \tag{1}$$

In the same manner we obtain for V_{13}:

$$V_{13} = -c_3 + \beta/3[V_{13} + V_{13} + \pi_{31}(u + V_{12}) + \pi_{32}(\tau^1_{23}V_{13} + (1 - \tau^1_{23})V_{12})]. \tag{2}$$

Evaluating the difference $\Delta_1 = V_{12} - V_{13}$, we have

$$\Delta_1[1 - \frac{\beta}{3}(2 - \pi_{23}\tau^1_{23} + \pi_{32}\tau^1_{23})] = (c_3 - c_2) + \frac{\beta}{3}u[\pi_{21} - \pi_{31}].$$

Since the left hand side of this equation is always positive, we get

$$V_{12} - V_{13} > 0 \Rightarrow \tau^1_{23} = 0 \Leftrightarrow (c_3 - c_2) > \beta/3(\pi_{31} - \pi_{21})u. \tag{3}$$

(3) can be interpreted as follows: a fundamental strategy for type 1 is the best answer to fundamental strategies of the other types iff the difference of storage costs is higher then the utility gain due to the higher probability of finding a suitable trading partner, described by $(\pi_{31} - \pi_{21})$.

If we derive utilities for type 2 and 3 and their differences in the same way, we obtain $V_{21} > V_{23}$ and $V_{31} > V_{32}$ for all parameter values. Therefore we have an equilibrium in fundamental strategies with transition matrix:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 2/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{(1+\pi_{13}+\pi_{31})}{3} & \frac{(\pi_{12}+\pi_{32})}{3} & 0 & 0 \\ 0 & 0 & \frac{\pi_{31}}{3} & \frac{(2+\pi_{32})}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1+\pi_{13})}{3} & \frac{(1+\pi_{12})}{3} \end{pmatrix}. \tag{4}$$

Solving the transition equation $\pi = P'\pi$ gives the steady state distribution $\pi' = (1, 0, 0.5, 0.5, 1, 0)$. Substituing in 3 one gets the necessary and sufficient condition for a fundamental equilibrium:$(c_3 - c_2) > \beta/6u$.

In the fundamental equilibrium individuals of type 1 and type 3 store their produced good until they find a partner for direct exchange to get their consumption good. On the other hand agents of type 2 will always trade good 3 for good 1 due to lower costs and use good 1 as intermediate good. Therefore good 1 is the unique commodity money in this equilibrium.

Speculative equilibrium

What happens if inequality (3) is violated? In this case no fundamental equilibrium exists, but there is another equilibrium we call speculative, with speculative strategies of the form:

$$\begin{array}{llllll} \tau^1_{21} = 1, & \tau^1_{22} \in [0,1], & \tau^1_{23} = 1, & \tau^1_{31} = 1, & \tau^1_{32} = 0, & \tau^1_{33} \in [0, I], \\ \tau^2_{32} = 1, & \tau^2_{33} \in [0,1], & \tau^2_{31} = 1, & \tau^2_{12} = 1, & \tau^2_{13} = 0, & \tau^2_{11} \in [0,1], \\ \tau^3_{13} = 1, & \tau^3_{11} \in [0,1], & \tau^3_{12} = 0, & \tau^3_{23} = 1, & \tau^3_{21} = 1, & \tau^3_{22} \in [0,1]. \end{array}$$

The implied ranking of value functions is: $V_{11} > V_{13} > V_{12}, V_{22} > V_{21} > V_{23}, V_{33} > V_{31} > V_{32}$. To check whether these strategies lead to an equilibrium or not we have to proof that fundamental strategies continue to be the best answers of type 2 and 3 agents. We see, that the strategies of type 2 and 3 do not change, the only difference to the former equilibrium is the behavior of type 1 agents.

Setting up the value functions for type 2 we obtain:

$$V_{21} = -c_{21} + \beta/3[\pi_{12}(u + V_{23}) + \pi_{13}(\tau^2_{13}V_{23} + (1 - \tau^2_{13})V_{21}) + V_{21} + \pi_{32}(u + V_{23}) + \pi_{31}V_{21}].$$

$$V_{23} = -c_{23} + \beta/3[\pi_{12}(u + V_{23}) + \pi_{13}V_{23} + V_{23} + \pi_{32}(u + V_{23}) + \pi_{31}(\tau^2_{13}V_{23} + (1 - \tau^2_{13})V_{21})].$$

Deriving the difference $(V_{21} - V_{23})$ we get that $V_{21} > V_{23}$ continues to hold and in the same way we obtain $V_{31} > V_{32}$. Hence we have the following result:

$$V_{12} - V_{13} < 0 \Rightarrow \tau^1_{23} = 1 \Leftrightarrow (c_3 - c_2) < \beta/3(\pi_{31} - \pi_{21})u. \tag{5}$$

Iff for type 1 the additional costs of holding good 3 are lower then the additional utility due to the higher probability of finding a partner who supplies good 1, there exists an equilbrium with speculative strategies for type 1 and fundamental strategies for the other types.

The transition matrix for the speculative equilibrium has following form:

$$P = \begin{pmatrix} \frac{(2+\pi_{21})}{3} & \frac{\pi_{23}}{3} & 0 & 0 & 0 & 0 \\ \frac{\pi_{31}}{3} & \frac{(2+\pi_{32})}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{(1+\pi_{13}+\pi_{31})}{3} & \frac{(\pi_{12}+\pi_{32})}{3} & 0 & 0 \\ 0 & 0 & \frac{\pi_{31}}{3} & \frac{(2+\pi_{32})}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1+\pi_{13})}{3} & \frac{(1+\pi_{12})}{3} \end{pmatrix}. \tag{6}$$

Once again we use the transition equation $\pi = P'\pi$ to get the steady state vector $\pi' = (0.5\sqrt{2}, \frac{\sqrt{2}-1}{\sqrt{2}}, 2-\sqrt{2}, \sqrt{2}-1, 1, 0)$. Substituting in (5) we obtain that $(c_3 - c_2) < \beta/3(\sqrt{2}-1)u$ must hold so that the economy is in a speculative equilibrium.

In contrast to the fundamental case in this equilibrium two goods emerge as commodity money, namely good 1 and 3. Therefore we can say that low storage costs are not the only reason why indviduals use money. The other one is that money makes it easier to find a suitable trading partner.

It can be shown that all other combinations of strategies do not lead to any equilibrium, so in pure strategies this environment does not lead to an equilibrium for all parameter values.

3 Simulations

In this section we analyse the dynamics of the model with GA. Our main questions are: Do the economy converge to a steady state equilibrium, and if so, do individuals with bounded rationality learn the optimal strategies of this equilibrium.

We use a population size of 150, 30 per type, and run for every environment 100 simulations, each of them simulating 5000 periods. The coding of the trading strategies is as follows: We use a string of length 6, each bit represents one trading strategy τ^i_{jk} of indvidual i, the only assumption is, that individuals always consume their consumption good. At period 0 we initialize the string randomly for each individual. In every period individuals are matched pairwise and dependent on their trading strategies a trade occurs.

As raw fitness function we use the discounted utility of an individual (with linear transformation to avoid negative values of fitness). This raw fitness is linearly scaled to avoid premature convergence and slow finishing (with a factor of scale of 1.5). Let us have a closer look to the discounted utility of good 2 for an individual of type 1; the other utilities are derived similiarly. $\bar{\tau}^i_{jk}$ denotes the average strategy of an individual of type i for trading good j for good k.

$$\begin{aligned} V_{12}(t-1) = {} & -c_2 + \beta/3[\pi_{12}(t)V_{12}(t) + \pi_{13}(t)\tau^1_{23}(t)\bar{\tau}^1_{32}(t)V_{13}(t) + \pi_{13}(t)(1 - \tau^1_{23}(t)\bar{\tau}^1_{32}(t))V_{12}(t) \\ & + \pi_{21}(t)\tau^1_{21}(t)\bar{\tau}^2_{12}(t)(u + V_{12}(t)) + \pi_{21}(t)(1 - \tau^1_{21}(t)\bar{\tau}^2_{12}(t))V_{12}(t) \\ & + \pi_{23}(t)\tau^1_{23}(t)\bar{\tau}^2_{32}(t)V_{13}(t) + \pi_{23}(t)(1 - \tau^1_{23}(t)\bar{\tau}^2_{32}(t))V_{12}(t) + \pi_{31}(t)\tau^1_{31}(t)\bar{\tau}^3_{12}(t) \\ & (u + V_{12}(t)) + \pi_{31}(t)(1 - \tau^1_{31}(t)\bar{\tau}^3_{12}(t))V_{12}(t) + \pi_{32}(t)V_{12}(t)]. \end{aligned} \tag{7}$$

As approximation of the variables at time t we use their values at time $(t-1)$ since strategies do not change dramatically between two periods (see also [7]). After evaluating the fitness of a string selection takes place in 9 groups where each group is characterized through type i holding good j, $i, j = 1 \ldots 3$. After selection one-point crossover with probability $p_c = 0.8$ and mutation with probability $p_m = 0.001$ take place. Afterwards each new individual is endowed with the good of her parents or if the parents had the consumption good with one unit of the good produced by her type.

We use in all environments the following parameter specifications: $\beta = 0.75, c_1 = 5, c_2 = 10, c_3 = 20$ and vary u in order to investigate different equilibria.

Environment A

In the first group of simulations we set $u = 70$, such that inequality (3) holds. Therefore the economy should converge to the fundamental equilibrium. Analyzing the simulations one obtains that every system out of 100 converges to the fundamental equilibrium. The analysis shows that individuals learn the optimal strategies quickly. If we look to a particular system after 100 periods nearly every individual plays the fundamental strategy and this holds until the end of simulation. The distribution of endowments π only slightly differs from its steady state distribution.

We obtain that due to lower storage costs individuals accept good 1 quickly as intermediate good.

We see that even we give up rationality and model some kind of learning by GA fundamental strategies get accepted, if inequality (3) holds.

Environment B

By means of this environment we want to study speculative equilibria. We investigate whether inequality (5) is sufficient that type 1 agents play speculative strategies. With $u = 100$ this inequality is for type 1 just satisfied and from the last section one would predict a speculative equilibrium.

But the analysis of the simulations shows a different result: only 11 simulations tend to the speculative equilibrium, in all other cases all individuals play fundamental strategies. It looks like fundamental strategies are the prefered response of individuals if we do not assume rational expectations. Agents prefer to use the good with the lowest storage cost as intermediate good, even though this is not optimal. Furthermore one obtains that even in simulations where speculative strategies evolve, type 1 agents need much more time to learn them. In none of these simulations speculative strategies are played by a majority of type 1 individuals before period 2000.

How can we interpret this non-optimal behavior? Individuals behave myopic in the following sense: Agents of type 1 give too much weight to lower costs in the present and less weight to a higher probability of finding a suitable trading partner in future.

Environment C

Using $u = 140$ we study if higher utility effects the strategies of type 1. We get that the number of simulations leading to a speculative equilibrium increases significantly. In 69 out of 100 simulations the economy converges to the speculative equilibrium and the speculative strategies come up earlier (about period 500) then under environment B. Furthermore one obtains that the distribution of endowments reached nearly the steady state distribution.

This result leads to the following proposition: The stronger inequality (5) holds, the higher is the probability that speculative strategies emerge. If the utility gain is very high relative to the additional costs, then also individuals with bounded rationality prefer to play speculative strategies, but in one third of all cases they continue to have fundamental strategies.

Environment D

To confirm the proposition made above we increase u to 210. With this choice of parameters 78% of all simulations converge to a speculative equilibrium, but in the remaining simulations the system converges once more to the fundamental equilibrium. The utility gain due to the higher probability of finding a type 3 individual, who supplies good 1 overcompensates the higher storage costs of good 3, so that also myopic agents prefer to play speculative strategies.

4 Conclusions

In this paper we derived a model where money comes up because individuals choose their trading strategies by taking into account the storage costs of goods and the probability of finding a trading partner. We distinguished two types of possible Markov-Nash-equilibria: a fundamental one where the good with the lowest storage costs as intermediate good and a speculative one where the good with the highest costs becomes medium of exchange. Which of these two is the steady-state equilibrium depends on the relation between costs and probabilities of finding some who supplies the consumption good.

The simulations with GA lead to the following results: The dynamic system always converges to the fundamental equilibrium, iff the condition for a fundamental equilibrium is satisfied. Otherwise the condition for a speculative equilibrium is necessary but not sufficient. A large number of simulations shows that individuals of type 1 prefer to hold the good with the lowest costs. The probability that type 1 agents use speculative strategies increases if the utility of consumption increases but fundamental strategies do not disappear. Hence we get that myopic individuals prefer fundamental strategies and the good with the lowest costs becomes money.

References

[1] Goldberg David E., Genetic Algorithm in Search, Optimization and Machine Learning, Addison Wesley, 1989.

[2] Kehoe T. & Kiyotaki N. & Wright R., More on Money as a Medium of Exchange, Economic Theory 3 (1993), 297-314.

[3] Kiyotaki N. & Wright R., On Money as a Medium of Exchange, Journal of Political Economy 97 (1989), 927-954.

[4] Kiyotaki N. & Wright R., A Contribution to the Pure Theory of Money, Journal of Economic Theory 53 (1991), 215-235.

[5] Kiyotaki N. & Wright R., A Search-Theoretic Approach to Monetary Economics, American Economic Review 83 (1993), 63-77.

[6] Marimon R., McGrattan E. & Sargent T., Money as Medium of Exchange in an Economy with Articficially Intelligent Agents, Journal of Economic Dynamics and Control 14 (1993), 329-373.

[7] Wright Randall, Search, Evolution and Money, CARESS Working Paper 93-22 (1993).

A DETERMINISTIC MODEL FOR MANPOWER SYSTEMS

Hong Gao[1], Baifu Gao[2] and Dehui Pan[3]
[1]Department of Adaptive Systems
Institute of Information Theory and Automation
Czech Academy of Sciences, E-mail: gao@utia.cas.cz
[2] Department of Mathematics
Teachers College, Shenyang University
Shenyang 110015, P.R.China
[3] Department of Automatic Control
Northeastern University
Shenyang 110006, P.R.China

Abstract. A distributed parameter system model is derived by analysing the internal dynamics of a manpower system in this paper. This model lays the foundation for analysis, prediction and control of manpower systems.

Introduction

As the planned economic system is being transferred to the market economic system in China, the personnel departments are changing the traditional experience administration into the scientific one. Prediction, agent and service of manpower will become the three important functions of the personnel departments in the future. Therefore, how to make good manpower planning is an important task for the personnel departments.

So far, many attempts have been done in order to make good manpower planning which meets the need of the social and economic development of China. Some mathematical models for manpower systems have been proposed [1][3]. However, there is no scientific mathematical model for manpower system which is generally acknowledged to perform reliably because of the complexity of manpower systems. In this paper a distributed parameter system model is derived by analysing the internal dynamics of a manpower system. Using this model, we can study the dynamical development of manpower with different specialities and ages, which lays the foundation for analysis, prediction and control of manpower systems.

Distributed parameter system model

The dynamic development of a manpower system is influenced by many factors such as the economic level, education level, natural enviorments and so on. However, manpower supplement, death, retirement and migration are the main reasons which determine dynamics of a manpower system. If we can quantify the relationship among the four factors, we'll obtain a mathematical model which describes the dynamic development of the manpower system.

First of all, we define the following functions before deriving the mathematical model.

(1) Manpower Function $S_i(x,t)$

The manpower function $S_i(x,t)$ is a bivariate function where x denotes age, t denotes time and i denotes speciality. $S_i(x,t)$ represents the manpower quantity of individuals whose ages are less than x at time t and working at the speciality i. Assuming that there are n specialities in all in the studied region gives $1 \leq i \leq n$.

The above definition of $S_i(x,t)$ gives its following properties:

- $S_i(x,t) \geq 0 \quad for \quad \forall x,t,i$
- $S_i(x,t)$ is a nondecreasing function of x, this is,
  $S_i(x_2,t) \geq S_i(x_1,t)$ if $x_2 > x_1$ for any fixed t and i
- $S_i(x_0,t) = 0$, where x_0 is the minimal age of all individuals of the manpower system

- $S_i(x_m, t) = S_i(t)$, where x_m represents the maximum age of all individuals of the manpower system and $S_i(t)$ is the sum of all individuals working at the speciality i at time t
- $S(t) = \sum_{i=1}^{n} S_i(t)$, which is the sum of all individuals of the manpower system at time t

For convenience's sake, we assume that $S_i(x,t)$, $\frac{\partial S_i(x,t)}{\partial t}$ and $\frac{\partial S_i(x,t)}{\partial x}$ are continuous functions.

(2) Manpower Density Function $q_i(x,t)$

The manpower density function $q_i(x,t)$ is defined by $q_i(x,t) = \frac{\partial S_i(x,t)}{\partial x}$. From the properties of $S_i(x,t)$, we know that

- $q_i(x,t) \geq 0$
- $q_i(x_m,t) = 0$
- $S_i(x,t) = \int_{x_0}^{x} q_i(\xi,t)d\xi$
- $S_i(x_m,t) = \int_{x_0}^{x_m} q_i(\xi,t)d\xi = S_i(t)$
- $S_i(x_2,t) - S_i(x_1,t) = \int_{x_1}^{x_2} q_i(\xi,t)d\xi$ which is the manpower quantity of the individuals whose ages are between x_1 and $x_2 (x_2 > x_1)$ and working at the speciality i at time t.

Let $\Delta x > 0$ is an enough small age interval, then the manpower quantity of the individuals whose ages are between x and $x + \Delta x$ and working at the speciality i at time t is $q_i(x,t)\Delta x$

(3) Retirement Rate Function $\mu(x,t)$

Before defining the retirement rate function, we need the following assumptions:

- The manpower death is regarded as another manifestation of the manpower retirement, that is, the manpower retirement includes the manpower mortality.
- The retirement rate function is defined in the sense of averaging over speciality, that is, the retirement rates for all specialities are the same.

Now let $M(x, \Delta x, t)$ be the average manpower retirement quantity of individuals whose ages are between x and $x + \Delta x$ within an unit period. At the same time the total manpower quantity of individuals whose ages are between x and $x + \Delta x$ is $\sum_{i=1}^{n} q_i(x,t)\Delta x$. So the retirement rate function $\mu(x,t)$ is defined by

$$\mu(x,t) = \lim_{\Delta x \to 0} \frac{M(x, \Delta x, t)}{\sum_{i=1}^{n} q_i(x,t)\Delta x}$$

.

Therefore, for enough small Δx and Δt, the manpower retirement quantity of individuals whose ages are between x and $x + \Delta x$ and working at the speciality i within t and $t + \Delta t$ is $M_i(x, \Delta x, t) = \mu(x,t)q_i(x,t)\Delta x\Delta t$.

(4) Supplement Rate Function $f_i(x,t)$

For a manpower system, the manpower supplement mainly comes from two sources: one is the graduates from universities and colleges; the other is the manpower migration. Suppose that the immigration quantity is positive and the emigration quantity is negative.

Let $F_i(x, \Delta x, t)$ be the average manpower supplement quantity of individuals whose ages are between x and $x + \Delta x$ and working at the speciality i within an unit period. The manpower supplement rate $f_i(x,t)$ is defined by

$$f_i(x,t) = \lim_{\Delta x \to 0} \frac{F_i(x, \Delta x, t)}{\Delta x}$$

which is the control variable of controlling the development of the whole manpower system.

Therefore, the manpower supplement quantity of individuals whose ages are between x and $x + \Delta x$ and working at the speciality i within t and $t + \Delta t$ is $f_i(x,t)\Delta x\Delta t$

(5) Speciality Transfer Rate Function $a_{ij}(x,t)$

In the development of a manpower system, the phenomenon of speciality transfer widely exists, and it will become very noticeable when the social need for some specialities surpasses the manpower supply or the manpower supply of some specialities surpasses the social need. Therefore, the individuals with the surplus specialities will largely transfer to the short specialities. Meanwile, the present administrative system of China are also partially responsible for it.

Let $A_{ij}(x,\Delta x,t)$ be the average manpower quantity of individuals whose ages are between x and $x+\Delta x$ and who transfer from the speciality i to the speciality j within an unit period, $i,j = 1,2,\cdots,n$. The speciality transfer rate $a_{ij}(x,t)$ is defined by

$$a_{ij}(x,t) = \lim_{\Delta x \to 0} \frac{A_{ij}(x,\Delta x,t)}{q_i(x,t)\Delta x}$$

Therefore, the manpower quantity of individuals whose ages are between x and $x+\Delta x$ and who transfer from other specialities to the speciality i within t and $t+\Delta t$ is

$$\sum_{k=1,k\neq i}^{n} a_{ki}(x,t)q_k(x,t)\Delta x\Delta t$$

Based on the above analysis, we can easily derive the mathematical model for a manpower system-Manpower Evolution Equation. The manpower quantity of individuals whose ages are between x and $x+\Delta x$ and working at the speciality i at time t is $q_i(x,t)\Delta x$. After Δt, at time $t+\Delta t$, the manpower quantity of the retired individuals with the speciality i is $\mu(x,t)q_i(x,t)\Delta x\Delta t$; the manpower quantity of individuals who transfer from other specialities to the speciality i is $\sum_{k=1,k\neq i}^{n} a_{ki}(x,t)q_k(x,t)\Delta x\Delta t$; the quantity of the graduates from universities and colleges and immigrated individuals who have the speciality i is $f_i(x,t)\Delta x\Delta t$. At the same time, the individuals whose ages are between x and $x+\Delta x$ at time t become the individuals whose ages are between $x+\Delta t$ and $x+\Delta x+\Delta t$ at time $t+\Delta t$. It is apparent that the total manpower quantity of individuals who are working at the speciality i at time $t+\Delta t$ is $q_i(x+\Delta t,t+\Delta t)\Delta x$. Therefore, the following formula holds

$$\begin{aligned}&q_i(x+\Delta t,t+\Delta t)\Delta x - q_i(x,t)\Delta x \\ &= -\mu(x,t)q_i(x,t)\Delta x\Delta t + f_i(x,t)\Delta x\Delta t + \sum_{k=1,k\neq i}^{n} a_{ki}(x,t)q_k(x,t)\Delta x\Delta t\end{aligned} \tag{1}$$

that is,

$$\begin{aligned}&q_i(x+\Delta t,t+\Delta t)\Delta x - q_i(x,t+\Delta t)\Delta x + q_i(x,t+\Delta t)\Delta x - q_i(x,t)\Delta x \\ &= -\mu(x,t)q_i(x,t)\Delta x\Delta t + f_i(x,t)\Delta x\Delta t + \sum_{k=1,k\neq i}^{n} a_{ki}(x,t)q_k(x,t)\Delta x\Delta t\end{aligned} \tag{2}$$

Dividing the two side of the formula (2) by $\Delta x\Delta t$ gives

$$\begin{aligned}&\frac{q_i(x+\Delta t,t+\Delta t) - q_i(x,t+\Delta t)}{\Delta t} + \frac{q_i(x,t+\Delta t) - q_i(x,t)}{\Delta t} \\ &= -\mu(x,t)q_i(x,t) + f_i(x,t) + \sum_{k=1,k\neq i}^{n} a_{ki}(x,t)q_k(x,t)\end{aligned} \tag{3}$$

Letting $\Delta t \to 0$ gives

$$\frac{\partial q_i(x,t)}{\partial x} + \frac{\partial q_i(x,t)}{\partial t} = -\mu(x,t)q_i(x,t) + f_i(x,t) + \sum_{k=1,k\neq i}^{n} a_{ki}(x,t)q_k(x,t) \tag{4}$$

which is a one-order linear partial differential equation, called Manpower Evolution Equation.

The initial and boundary conditions of equation(4) are

$$q_i(x,t_0) = q_{i0}(x), \quad q_i(x_m,t) = 0 \tag{5}$$

Combining equation (4) and (5) gives a integrated system of differential equatrions which describes the manpower evolution

$$\begin{aligned}
&\frac{\partial q_i(x,t)}{\partial x} + \frac{\partial q_i(x,t)}{\partial t} = -\mu(x,t)q_i(x,t) + f_i(x,t) + \sum_{k=1,k\neq i}^{n} a_{ki}(x,t)q_k(x,t)\\
&q_i(x,t_0) = q_{i0}(x)\\
&q_i(x_m,t) = 0\\
&i = 1,2,\cdots,n
\end{aligned} \tag{6}$$

where $q_{i0}(x)$ is the given initial manpower distribution.

The system of equations (6) also can be written as a matrix format

$$\begin{aligned}
&\frac{\partial Q(x,t)}{\partial x} + \frac{\partial Q(x,t)}{\partial t} = F(x,t) + M(x,t)Q(x,t)\\
&Q(x,t_0) = Q_0(x)\\
&Q(x_m,t) = 0
\end{aligned} \tag{7}$$

where

$$\begin{aligned}
&Q(x,t) = [q_1(x,t), q_2(x,t)...q_n(x,t)]^T \qquad (8)\\
&F(x,t) = [f_1(x,t), f_2(x,t)...f_n(x,t)]^T\\
&M(x,t) = \begin{bmatrix} -\mu(x,t) & a_{21}(x,t) & \cdots & a_{n1}(x,t)\\ a_{12}(x,t) & -\mu(x,t) & \cdots & a_{n2}(x,t)\\ \vdots & \vdots & \ddots & \vdots\\ a_{1n}(x,t) & a_{2n}(x,t) & \cdots & -\mu(x,t) \end{bmatrix}\\
&Q_0(x) = [q_{10}(x), q_{20}(x)\cdots, q_{n0}(x)]^T\\
&Q(x_m,t) = [q_1(x_m,t), q_2(x_m,t),\cdots, q_n(x_m,t)]^T
\end{aligned}$$

The system of equations (7) is the desired mathematical model of the manpower system

Conclusion

Although a continuous mathematical model of a manpower system is derived only in theory in this paper, it has been applied to prediction and planning of the manpower system of some region and gives good results, which provides reference for scientific decision of personnel departments.

Acknowledgment. This research was partially supported by GAAV ČR, No A2075606 and by Copernicus CT94-1174. Ms H. Gao is also supported by Central European School in Control and System Theory.

References

[1] Gu, J.D., Yie, Y.M., Li, M.Q and Jiang, C.Z., The Specialized and Quanlified Person Requirement Forecasting of Fujian Province in 2000. System Engineering Theory and Practice, 1(1992), 1-6.

[2] Song, J and Yu J.Y.-Population Control Theory: Science Press, Beijing, 1985.

[3] Zhang,S.D. and Sun,H.X.-The Manpower Forecasting Methods: Liaoning Science and Technology Press, Shenyang, 1987.

HOW DOES THE INTRODUCTION OF EXPECTATIONS AFFECT AN ECONOMIC MODEL?

Belén LÓPEZ and José M.PACHECO
Departamento de Matemáticas
Universidad de Las Palmas de Gran Canaria
Campus de Tafira Baja
35017 LAS PALMAS
SPAIN

ABSTRACT

A family of simple models is studied in order to assess how expectations on the price of an asset can affect the behaviour of prices. Several assumptions are made and explored, and, finally, one of the models is worked in depth.

INTRODUCTION

In an earlier paper by one of the authors [2] a reaction-diffusion system was taken as a model of an simple economic system. The variables were "amount of information", I(x,t), and "price of an asset", P(x,t), and the model read:

$$\frac{\partial I}{\partial t} = f(I,P) + \alpha \frac{\partial^2 I}{\partial x^2}$$

$$\frac{\partial P}{\partial t} = K_2[h(I)-P]$$

where $f(I,P)=K_1 \frac{\partial P}{\partial t}$ and $h(I)=bI\left(1-\frac{a}{b}I\right)$. It was shown that the model predicts no cyclic behaviour, either for information or for prices.

In order to deal with expectations on the price values and trends, a fact that can affect investors in their plans, a family of models has been obtained taking the above one as a starting point.

Expectations can be defined on two different grounds [1], [4] as follows:

1.- Adaptative Expectations are the expression of the study of price behaviour and trends in the past. Mathematically they can be expressed as predictions, so the following formula is an acceptable one:

$$\hat{p}(x,t)=\sum_{k=0}^{n} c_k(x)p(x,t-kT)$$

where T=T(x) is some subjetive unit delay and $c_k(x)\geq 0$ are weights attributed by investor x to the past values. A natural condition for the sequence c_k is $\forall k,\ c_k \geq c_{k+1}$ and $\lim_{k\to\infty} c_k = 0$.

2.- Rational Expectations are unbiased estimates and are random in nature. A deterministic expression often employed is:

$$\hat{p}(x,t) = m\frac{\partial P}{\partial t} + K_3 \frac{\partial^2 P}{\partial t^2}$$

As a rule, expectations should be included in the above model as modifications to the I equation, although this idea will not met in the worked model described below.

A FIRST MODEL

In this model we consider rational expectatives and we modify the basic model to obtain:

$$\frac{\partial I}{\partial t} = K_1 \frac{\partial P}{\partial t} + \alpha \frac{\partial^2 I}{\partial x^2} + \left(m\frac{\partial P}{\partial t} + K_3 \frac{\partial^2 P}{\partial t^2}\right)$$

$$\frac{\partial P}{\partial t} = K_2[h(I)-P]$$

The expectation terms plus $K_1 \frac{\partial P}{\partial t}$ can be grouped to yield a global timelike behaviour for P, i.e. $(K_1+m)\frac{\partial P}{\partial t}+K_3\frac{\partial^2 P}{\partial t^2}$, an expression that can be identified, via the telegraph equation, with a diffusion , to obtain the model:

$$\frac{\partial I}{\partial t} = \alpha \frac{\partial^2 I}{\partial x^2} + \omega \frac{\partial^2 P}{\partial x^2}$$

$$\frac{\partial P}{\partial t} = K_2\left[h(I)-P\right]$$

The diffusion in the P variable accounts for the random character of the rational expectative approach. Nevertheless, the coupling of the equations prevents a simple mathematical treatment, so we do not follow this modelling.

A SECOND MODEL

For this model adaptive expectatives are chosen, and in order to simplify things at a maximum, a single term will be preserved in the formula for $\hat{p}$, i.e.:

$$\hat{p}(x,t) = \sum_{k=0}^{n} c_k(x)p(x,t-kT) = c_0P(x,t-T)$$

to obtain the model equations:

$$\frac{\partial I}{\partial t} = K_1 \frac{\partial P}{\partial t} + \alpha \frac{\partial^2 I}{\partial x^2} + c_0P(x,t-T)$$

$$\frac{\partial P}{\partial t} = K_2\left[h(I)-P\right]$$

Following the approach in [2], one seeks for travelling wave solutions, i.e. solutions where dependence on x and t happens only through the variable $\xi = x-ct$. Here c is an unknown wave speed to be determined afterwards. The ansatz $\xi = x-ct$ yields the ODE system:

$$\alpha\, I''+cI'-K_1cP'+c_0P(\xi-cT)=0$$

$$-cP'=K_2\left[h(I)-P\right]$$

Integration of the first equation gives the integro-differential system:

$$K_1cP'-\int_{\xi_0}^{\xi} P(s-cT)ds=\alpha\, I'+cI$$

$$-cP'=K_2\left[h(I)-P\right]$$

Again, solving the model equations seems a difficult task and we give up this modelling and look for something easier by changing our view point.

A THIRD MODEL

For this model we employ again adaptive expectations, but, instead of changin the I equation, we assume that expectations affect the time trend of prices. Again, we look for travelling wave solutions and find the delayed ODE system:

$$\alpha\, I'' + cI' - K_1cP' = 0$$

$$-cP'+K_2P-c_0P(\xi-cT)=K_2h(I)$$

We do not dwell in this formulation, but rather we go back to the second model and modify it to write it in a more tractable way.

A FOURTH (AND LAST) MODEL

Let us consider the second model and let us drop the diffusion term on the I equation. This amounts to saying that expectations are a private affair of a single investor who does not share this information with anyone else. Now the model is an ODE-only one:

$$I' = K_1P' + c_0P(t - T)$$

$$P' = K_2\left[h(I)-P\right]$$

As long as there is only one investor, h(I), can be simplified and put simply to h(I)=I, a threshold phenomenon. Integration in t of the first equation yields:

$$I=K_1P+c_0\int_{t_0}^{t}P(s-T)ds$$

and plugging this expression into the second, a delayed integro-differential equation is obtained:

$$P'=K_2(K_1-1)P+K_2c_0\int_{t_0}^{t}p(s-T)ds$$

We write $K_2(K_1-I)=K$, $K_2c_0=C$ and our equation reads:

$$P'=KP+C\int_{t_0}^{t}P(s-T)ds$$

Taking the derivative with respect to t, we arrive at a second order ODE with delay:

$$P''(t)-KP'(t)-C[P(t-T)-P(t_0-T)]=0$$

And if we choose to put $P(t_0-T)=0$, we are left with the following delayed differential equation:

$$P''(t) - KP'(t) - CP(t - T) = 0$$

Solutions to this equations are sought for in the form $P=me^{\lambda t}$ and the quasi-characteristic transcendental equation $\lambda^2-K\lambda-Ce^{-\lambda T}=0$ is found. Under the assumptions made in [2], this equation has two real roots, a positive and a negative one. The positive one has no practical interest, for it predicts an endless growth of prices; the negative one is discarded as it predicts a constant price behaviour, a featureless situation. Thus, we are left with possible oscillatory solutions associated with complex roots of the quasi-characteristic equation. An existence proof is the following:

Let $\lambda=\frac{1}{z}$, with $z\in C$. Then, the transcendental equation is:

$$\frac{1}{z^2}-\frac{K}{z}-Ce^{-\frac{T}{z}}=0$$

The power expansion of $e^{-\frac{T}{z}}=1-\frac{T}{z}+\frac{1}{2!}\frac{T^2}{z^2}-\frac{1}{3!}\frac{T^3}{z^3}+...=1+\sum_{n=1}^{\infty}\frac{(-1)^nT^n}{n!z^n}$, shows that the function:

$$f(z)=\frac{1}{z^2}-\frac{K}{z}+1+\sum_{n=1}^{\infty}\frac{(-1)^nT^n}{n!z^n}$$

has an essential singularity at z=0. Therefore, an application of the Picard Theorem yields the existence of infinitely many roots to the equation f(z)=0.

Once existence has been established, let $\lambda = \alpha + i\beta$. Taking this into the quasi-characteristic equation yields, after equating real and imaginary parts:

$$\alpha^2 - \beta^2 - K\alpha = Ce^{-T\alpha}\cos(\beta T)$$

$$2\alpha\beta - K\beta = -Ce^{-T\alpha}\sin(\beta T)$$

Squaring both equations and adding , we obtain:

$$\alpha^4-2K\alpha^3+\left(K^2-2\beta-4\beta^2\right)\alpha^2+\left(2\beta K-4\beta^2K\right)\alpha^2+\left(1-K^2\right)\beta^2=C^2e^{-2T\alpha}$$

and putting $\alpha=0$ for purely oscillatory behaviour, we have:

$$\left(1-K^2\right)\beta^2=C^2 \quad \text{or} \quad \beta=\frac{C}{\sqrt{1-K^2}}$$

And by plugging this value into the second equation with $\alpha=0$ the following relationship between K, C and T is obtained:

$$\frac{KC}{\sqrt{1-K^2}} = -C\sin\left(\frac{TC}{\sqrt{1-K^2}}\right)$$

or

$$T = \frac{\sqrt{1-K^2}}{C}\arcsin\left(\frac{K}{\sqrt{1-K^2}}\right)$$

Thus we have obtained a time dalay T which is best fitted to yield oscillatory behaviour.

CONCLUSIONS

For a single investor, making predictions on the price value with a delay $T = \frac{\sqrt{1-K^2}}{C}\arcsin\left(\frac{K}{\sqrt{1-K^2}}\right)$ amounts to adopt a conservative policy where risk is certainly bounded.

The model predicts cyclic behaviour for this value of T, a fact seemingly in contradiction with the results in [1]. This apparent incoherence can be surpassed by thinking that the fourth model is a highly idealized one, and any small random perturbation can drive it away from the oscillatory behaviour.

ACKNOWLEDGEMENTS

Both authors acknowledge interesting remarks from Prof. Luis González (ULPGC) and from Prof. Alfonso Casal (Madrid).

REFERENCES

[1] **Chiang A.** (1984) "Fundamental Methos of Mathematical Economics". McGraw-Hill USA.

[2] **Pacheco J.** (1996) "Reaction-Diffusion Equations and Economic Cycles", J. of Institute of Mathematicas &Computer Sci (Math.Series), 9(2) (in the press).

[3] **Rudin W.** (1987) "Real and Complex Analysis", McGraw-Hill USA.

[4] **Samuelson P., Nordhaus W.** (1992), "Economics", McGraw-Hill USA.

FRUSTRA LABORAT QUI OMNIBUS PLACERE STUDET

Rudolf Starkermann
Grabemattweg 14, Ch-5443 N'Rohrdorf, Switzerland

Abstract

The essay investigates the validity of the well known saying "He that would please all and himself, too, undertakes what he cannot do". The basic assumption is that an unconscious relationship exists between a prime unit - who is the giver - and each one of the secondary units - which are several recipients. This unconscious relationship is called attitude. For the scope of this paper, the attitudes are devotional, or conciliatory. It is the meaning that the essay reflects the intention of the United States, namely, to help the world wherever there is a request for - and by doing so are damaging themselves. - The model is exercised in four different modes. In one, the prime unit does not give at all, i.e., there is only attitude. In another mode, the prime unit is in addition to attitude recognizing the demand of the secondary units. For a third case, the prime unit gives by evaluating the needs, or the demands, and provision happens accordingly. In a last case, the giving is without recognizing demand, i.e., giving is indiscriminately. - In all considered cases, the model is well in accordance with the proverb, i.e., it is true that the achievement of the prime unit not only becomes damaged by pleasing everybody, but, in more detail, that it suffers the more the greater the number of the units is the prime unit wants to, or has to, please. The worst case, the destruction of the prime unit, occurs when the prime unit observes the demands and gives accordingly, and if it strives to please more than one secondary unit.

Introduction

Referring to the title of the investigation, similar proverbes can be found in diverse cultures, what proves that the model verifies a natural law which is proven herein mathematically:

> Who serves everybody gets thanks from nobody. - Everybody's friend is nobody's friend.
> French: On ne peut contenter tout le monde et son père. - L'ami de tout le monde n'est l'ami de personne.
> German: Allen Leuten recht getan, ist eine Kunst, die niemand kann. Jedermanns Freund ist niemands Freund.
> Italien: Amico di tutti, amico di nessuno.

A more drastic saying is the French maxim:

> Dépens le pendu, il te pendra. - Release the man to be hung from the rope, and he will hang you. In other words, you pay drastically for being extremely good.

Wherever there is interaction among units, it is this interaction which sets the very frame for any other action within the system of the units. The more interactive information is exchanged, the more the system is in danger of losing its homeostasis, its stable behaviour - and in consequence, its achievement. Only stable systems can behave goal oriented and achieve. As a general observation in life, all investigations where there is continuous flow of information show this phenomenal outcome: The more interaction, the less stable and the lower the achievement; a prime axiom of Natural Laws.

The mathematical knowledge of this fact originates from investigations of automatic multiple control systems of technical powerhouses. But there is inherent physical interaction not only in technical-physical, but as well in biological plants between different factors which have to be controlled in order to maintain an overall stable system's behaviour whilst striving toward achievements.

The assumption herein is that attitudes among people – attitude as an interacting field of communication – are originated and stored in the unconscious and, therefore, cannot be controlled by the consciousness of the individual. It seems that attitudes are necessary for survival. Their necessity is, therefore, inherently determined through evolution, and they are always present and, therefore, always have to be taken into account for any consideration or investigation of social behaviour.

This unconsciously caused interaction is modelled in Fig. 1 with the transfer functions S_{ik} (i,k = 1, 2, 3, 4, 5, i ≠ k). It is mutual circular information exchange between the giver and each recipient.

Ensuring stability is the fundamental requirement for a system to remain goal oriented. Therefore, the stability domains, firstly, have to be found. In the psycho-social realm, stability runs under the name homeostasis. Secondly, the goal achievements upon a set goal of all the individuals can be calculated, assuming that all parameters are taken within the stable operation of the systems. This goal achievements in this essay are calculated for a final state, i.e., for the state after a reasonable time has elapsed after goals were set.

Fig. 1 depicts the basic structure of the prime unit U_1, which has unconsious attitude with four secondary units, U_2, U_3, U_4, and U_5. U_1 observes the demand of the four units and serves them.

u_i (i = 1...5): Goals; i.e., each unit's self-realization; u_i is considered to be constant in its amount;

S_{ik} (i,k = 1...5): Unconscious information transfer functions;

F_{i1} (i = 1...5): Transfer functions of the volition factor for self-realization, including speed of action;

V_{1k} (k = 2...5): U_1's observation transfer function observing the units U_2 to U_5;

V_{i1} (i = 2...5): U_1's service transfer function, serving the units U_2 to U_5;

$x_i(t)$(i = 1,...5): Goal variables of the units' self-realization;

UIB: Unconscious information block, the attitude fields of communication.

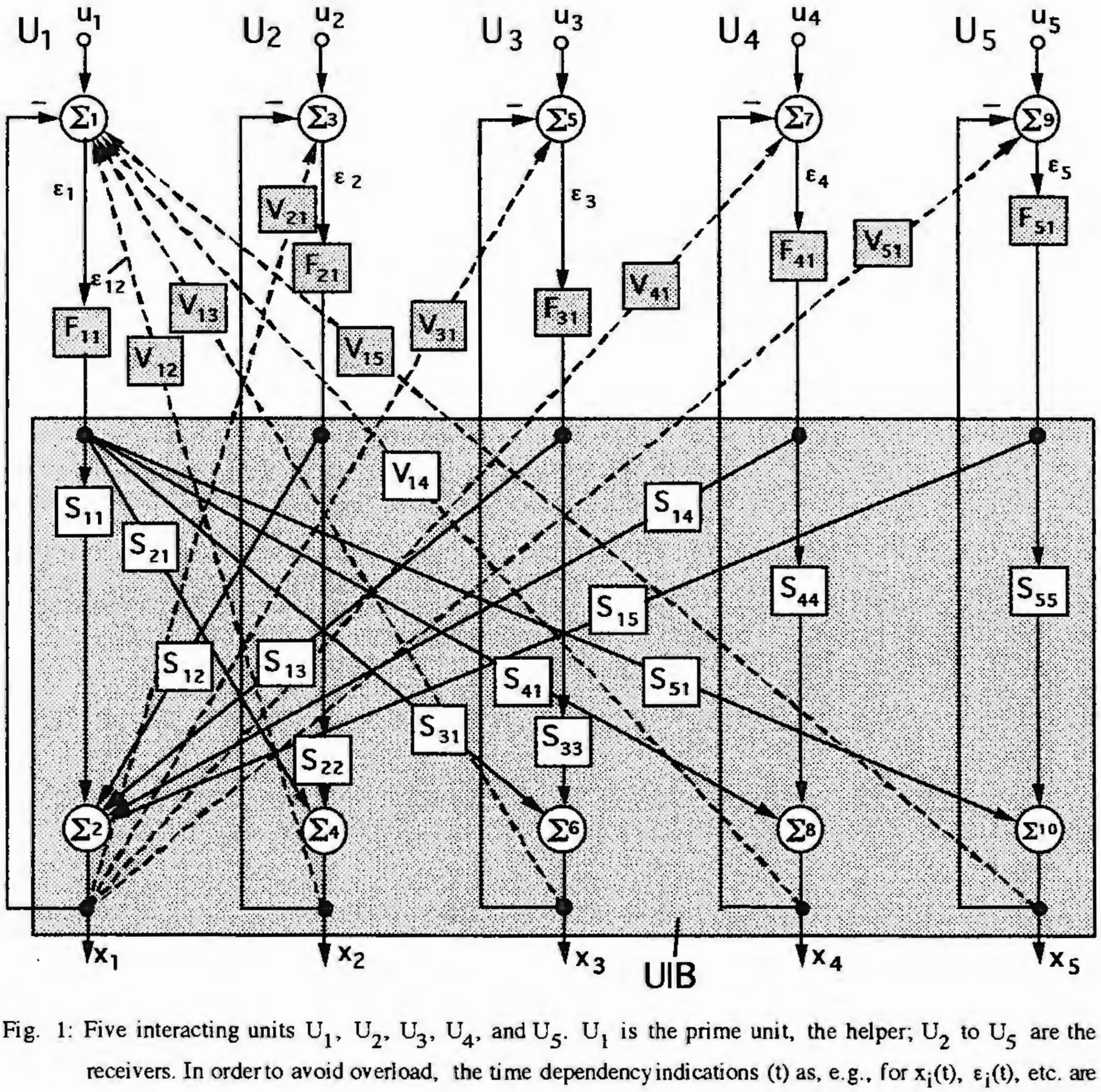


Fig. 1: Five interacting units U_1, U_2, U_3, U_4, and U_5. U_1 is the prime unit, the helper; U_2 to U_5 are the receivers. In order to avoid overload, the time dependency indications (t) as, e.g., for $x_i(t)$, $\varepsilon_i(t)$, etc. are omitted.

One unit

The model of a single goal oriented individual as a social unit, or a one–goal oriented group as a unit, has been described extendedly in [1]. There, it was given as a notion that each unit is a one–goal oriented functional entity which has self–control of its goal proximity. The goal is the unit's self–realization. Whatever a unit does, at any instant, it wants to realize itself (what else can it do than realize itself?). In addition, the unit has a dynamic of action and, therefore, needs time to act; furthermore, it can become unstable if it exceeds its volition beyond a certain value when striving toward its goal. (The pronoun for unit is sometimes „it", sometimes „he".)

More than one unit

Because in our consideration each secondary unit requires help or expects support from the prime unit (or because the prime unit's intention is to provide help to all secondary units) the relation between all secondary units with the prime unit is conciliatory. This relationship is based on attitude which originates in the unconscious of the individuals (or units), but manifests itself in the increase or decrease of the goal achievement of the prime unit. The secondary units are assumed to have no relation among each other, neither by attitude nor by mutual or unilateral observation. In other words, they are considered to be autonomous relative to each other. They all want from, say, the prime unit, the giver.

Fig. 1 illustrates the fundamental structure of the model. It shows five units U_i (i = 1, 2, 3, 4, 5), where U_1 is the prime unit, the giver, and U_2, U_3, U_4, and U_5 are the four secondary units, the recipients. Each unit exerts its volition factor G_i through F_{i1} in order to reach its own goal, u_i, as closely as possible. The momentary own amount of achieved self-realization of every unit is $x_i(t)$. For the secondary units this amount $x_i(t)$ is compared with the desired amount u_i and with the received amount $V_{i1}x_i(t)$ from the giver. The comparison results in the error $\varepsilon_i(t)$. $\varepsilon_i(t)$ is that part of the goal which is not achieved yet.This error $\varepsilon_i(t)$ is accomplished by taking $x_i(t)$ as negative feedback, $-x_i(t)$, and by adding it and adding $V_{i1}x_i(t)$ to u_i. Thus: $\varepsilon_i(t) = u_i - x_i(t) + V_{i1}x_i(t)$. For the prime unit the error $\varepsilon_1(t)$ is found by adding $-x_1(t)$ and $\Sigma V_{1i}x_i(t)$ to u_1, i.e., $\varepsilon_1(t) = u_1 - x_1(t) + \Sigma V_{1i}x_i(t)$. The transfer function F_{i1} contains the dynamic behaviour, i.e. the speed of action, of the unit in the form of the expression (1):

$$F_{i1} = \frac{G_i}{(1 + Ts)^3} \text{ ; } s = \text{Laplace operator; } G_i = \text{volition factor or willpower} \quad (1)$$

(T in F_{i1}, the time constant, is further down set to 1, i.e., one time unit).
The transfer functions S_{ii} (i = 1,...5) and S_{ik} (i,k, = 1, 2, 3, 4, 5, i ≠ k) are taken as instantaneously acting with a transfer factor of 1. Therefore, S_{ii} and $|S_{ik}| = 1$, (there is no time-dependency).

Investigated systems

A remark about conciliation is in order here. Reference is made to Fig. 1, the dualism system U_1–U_2.
If the closed circuit which interrelates U_1 and U_2

$$F_{11}-S_{21}-\Sigma 4-\Sigma 3-F_{21}-S_{12}-\Sigma 2-\Sigma 1-F_{11}$$

has an overall negative sign, the relation of the system U_1–U_2 is called conciliation or devotion. If this circuit's loop results in a positive sign, the relation is called aggression. As the two negative feedback signals, one of U_1 and one of U_2, cancel out, (minus times minus is plus), and as F_{11} and F_{21} are always positive, the negative sign for conciliation must come either from S_{12} or from S_{21}. If S_{12} and S_{21} are both positive, the relation is aggressive. Therefore, if the transfer function S_{12} of the information coming from U_2, is taken by U_1 in the negative sense, U_1 provides the devotional attitude. U_1 subtracts an unconscious argument which comes from U_2, from his own error he is striving to reduce to zero, i.e., from ε_1, in order to come closer to his goal u_1. Through this demeanour, U_1 does not only help himself for his self-realization, - he also helps U_2 for his u_2 for better achievement. Conciliatory attitude results in mutual help, whereas aggressive attitude results in mutual damage.- This is correct only in pure attitude constellations, i.e., with no V-transfer actions yet!

Fig. 2 shows the different assumptions of information transfer between the prime unit and one secondary unit. In order to see the decrease of the prime unit's goal achievement by increasing the number of secondary units to be served or pleased, systems are considered with one, two, three, and four secondary units. The different information connections, a) to d), in Fig. 2 mean the following:

Fig. 2,a: There is only compliant attitude between the two units. U_1 provides the conciliation. This unit is the one who feels obligated for establishing the lenient comportment.

Fig. 2,b: Conciliatory attitude is provided by unit U_1, the giver. In addition, U_1 observes the need or perceives the demand coming from unit U_2. This demand shall be called ε_{12}. U_1 combines this signal ε_{12} with his own error signal (which is $u_1 - x_1$) to the total error ε_1: $u_1 - x_1 + \varepsilon_{12} = \varepsilon_1$. But U_1 does not give yet; The signal ε_{12} is taken in a positive sense, because U_1 feels guilty, feels obligated to give, what increases his error feeling.

Fig. 2,c: Conciliatory attitude is provided by U_1. In addition, U_1 observes the need or the demands from U_2 and acts accordingly. For U_2 the signal ε_{21} is taken in the negative sense, because ε_{21} reduces his error ε_2 by the amount of support he was given by U_1. The total error of U_2 is $u_2 - x_2 - \varepsilon_{21}$.

Fig. 2,d: U_2 provides conciliation, U_1 gives indiscriminately, i.e., without observation - or not being able to observe or to evalue the need. The error ε_{21} is taken in a positive sense, because it is assumed that U_2 wants more and more!

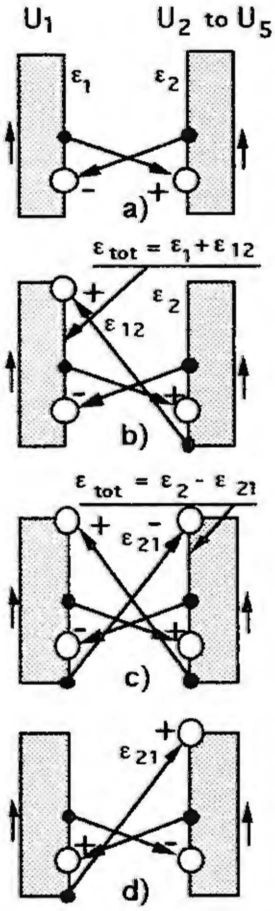


Fig. 2: Four different constellations between the prime unit U_1 and the secondary units U_2 to U_5.

It is obvious that a comprehensible picture requires a limitation of parameters and of parameter values which can be considered. Therefore, in taking into account both, stability limits and goal achievements, it is assumed, therefore, that all volitions G_i of the units are the same, i.e., all G_i are equal for any specific system. There is a reasonable justification for doing so: If the secondary units are in a state of demanding, then they are at least as powerful in their volition for their self-realization as the prime unit, who has to give. If the secondary units need help, then the prime unit, in order to be able to give, must be at least as powerful for his self-realization as the secondary units are. Therefore, joining both cases into one, parity of volition, i.e. equal volition for the prime and the secondary units, is a justified assumption for this preliminary investigation.

Although the mathematical development for the dynamics is surely cumbersome, it is elementary, and, thus, not given herein. All differential equations, up to the fifteenth degree, are of first order linear.

Table I gives the volitions G_i at the stability limits for the systems indicated in Fig. 2. It can be seen that the structures b) and d) result in the same values. The two systems, 2b) and 2d), are mathematically symmetrical.

Table I: Volitions G_i ($i = 1, 2, 3, 4, 5$) at the stability limit of the 4 configurations, Fig. 2.

System	Number of system units 1	2	3	4	5
a)	8.0	2.0	1.38	1.10	0.93
b)	8.0	1.36	0.86	0.65	0.53
c)	8.0	0.78	0.42	0.30	0.22
d)	8.0	1.36	0.86	0.65	0.53

Table II shows the goal achievements x_1/u_1 for the same 4 system configurations.

Table II: Goal achievements of the prime unit, the giver, of the 4 configurations with the volitions of Table I.

System	Number of system units				
	1	2	3	4	5
a)	0.89	0.77	0.75	0.74	0.73
b)	0.89	0.58	0.47	0.39	0.35
c)	0.89	0.35	0.25	0.21	0.17
d)	0.89	0.58	0.47	0.39	0.35

The stability limits were found by means of Matlab.
The steady state values of the prime unit's goal achievement x_1/u_1 in a general case can be found with the expression (2). Equation (2) is a quotient of two determinants.

$$\frac{x_1}{u_1} = \frac{\left.\begin{array}{|ccccc|} \multicolumn{1}{c}{n=1} & n=2 & n=3 & n=4 & \multicolumn{1}{c}{n=5} \\ b_1 & a_{12} & a_{13} & a_{14} & a_{15} \\ b_2 & a_{22} & a_{23} & a_{24} & a_{25} \\ b_3 & a_{32} & a_{33} & a_{34} & a_{35} \\ b_4 & a_{42} & a_{43} & a_{44} & a_{45} \\ b_5 & a_{52} & a_{53} & a_{54} & a_{55} \end{array}\right\} F_{11}}{\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{vmatrix}}, \qquad (2)$$

where:
$b_1 = S_{11}$; $b_2 = S_{21}$; $b_3 = S_{31}$; $b_4 = S_{41}$; $b_5 = S_{51}$;

$a_{11} = 1 + F_{11}S_{11} - S_{12}V_{21}F_{21} - S_{13}V_{31}F_{31} - S_{14}V_{41}F_{41} - S_{15}V_{51}F_{51}$;
$a_{12} = F_{21}S_{12} - V_{12}F_{11}S_{11}$; $a_{13} = F_{31}S_{13} - V_{13}F_{11}S_{11}$;
$a_{14} = F_{41}S_{14} - V_{14}F_{11}S_{11}$; $a_{15} = F_{51}S_{15} - V_{15}F_{11}S_{11}$;

$a_{21} = F_{11}S_{21} - V_{21}F_{21}S_{22}$; $a_{22} = 1 + F_{21}S_{22} - V_{12}F_{11}S_{21}$;

$a_{23} = -V_{13}F_{11}S_{21}$; $a_{24} = -V_{14}F_{11}S_{21}$;

$a_{25} = -V_{15}F_{11}S_{21}$;

$a_{31} = F_{11}S_{31} - V_{31}F_{31}S_{33}$; $a_{32} = -V_{12}F_{11}S_{31}$;

$a_{33} = 1 + F_{31}S_{33} - V_{13}F_{11}S_{31}$; $a_{34} = -V_{14}F_{11}S_{31}$;

$a_{35} = -V_{15}F_{11}S_{31}$;

$a_{41} = F_{11}S_{41} - V_{41}F_{41}S_{44}$ $a_{42} = -V_{12}F_{11}S_{41}$;

$a_{43} = -V_{13}F_{11}S_{41}$; $a_{44} = 1 + F_{41}S_{44} - V_{14}F_{11}S_{41}$;

$a_{45} = -V_{15}F_{11}S_{41}$;

$a_{51} = F_{11}S_{51} - V_{51}F_{51}S_{55}$; $a_{52} = -V_{12}F_{11}S_{51}$;

$a_{53} = -V_{13}F_{11}S_{51}$; $a_{54} = -V_{14}F_{11}S_{51}$;

$a_{55} = 1 + F_{51}S_{55} - V_{15}F_{11}S_{51}$; and

$$F_{i1} = \frac{G_i}{(1+Ts)^3}, \ s = i\omega, \text{ for } T = 1 \qquad (3)$$

by setting s, the Laplace operator, to zero (meaning that t = ∞. As mentioned above, only steady state situations are considered).

As all G_i are assumed to be equal per system, and only due to that, the expressions (2) for the different systems of one to five units, i.e., n = 1, 2, .., 5, simplify to the values given in Table III.

Table III: Goal achievements x_1/u_i for the systems a) to d).

System	x_1/u_i
a)	$\frac{(1 + nG)\,G}{1 + 2G + nG^2}$
b)	$\frac{G}{1 + G}$
c)	$\frac{G}{1 + nG}$
d)	$\frac{G}{1 + G}$

(4)

Discussion of the results

In order to put the values in Tables I and II in a clearer form, they are displayed as graphs.

Fig. 3 shows the maximum allowable volitions as a function of the numbers of the secondary units, (n–1). The autonomous unit, the unit U_1, has a maximum volition of 8. This can easily be shown by solving the characteristic equation of the single loop, equation (5) representing the characteristic behaviour of U_1. This value 8 serves as reference. As soon as the unconscious communication occurs with only one secondary unit, the volitions already have to drop for both units, U_1 and U_2, from 8 to 2, i.e., by the factor 4, $G_1 = G_2 = 2$. This system is the structure of 2,a). Having a conciliatory attitude communication with a second unit already requires an enormous reduction of the own ego. This is not easy to do, and it is the root of so many social conflicts.

$$\frac{G_i}{(1+Ts)^3} + 1 = 0, \ s = i\omega, \ T = 1 \qquad (5)$$

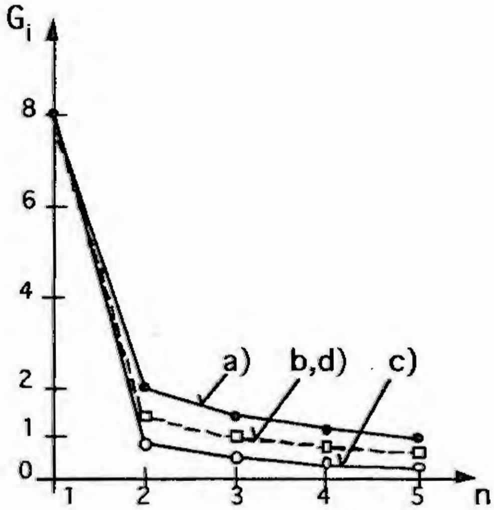


Fig. 3: Volitions G_i of the units n at the stability limit of the systems Fig. 2.
For each system of n units all volitions are equal. For n = 2 there is one secondary unit constituting the system; for n = 3, there are two secondary units constituting the system, etc, up to four secondary units at n = 5.

Multilateral communication (bilateral) in a devotionnal state requires yielding volition for the own self–realization for both partners (if they have equal volition. Imparity of volitions is not discussed within this essey.) One might refer to the saying: Two captains sink the ship. Both units in a two–unit–system cannot have high power for their own self-realization. The autonomous unit has the highest potential, the highest volition, for his success: *Der Starke ist am mächtigsten allein*. Schiller, Wilhelm Tell, 1/3. (The strong man is the mightiest being on his own.)

It seems that with $n \rightarrow \infty$, i.e., by having attitudes to more and more units, the volitions G_i tend to approach zero magnitude for all four configurations! Interaction consumes volition (and due to flow of energy among the units, interaction creates enthropy).

The assumed parameters, $|S_{ik}| = 1$, indeed, mean a strong attitude. For example with $S_{ik} = -1/2$, and $S_{ki} = +1/2$, the diminuation of the volitions would be less. Adding more communication channels, beside the ones for attitude, reduces the volition of the systems further. The worst case is, as demonstrated, case 2,c), a dual–bilateral communication pattern: attitude, observation and giving. The volitions have to be reduced tremendously.

Fig. 4 illustrates the also interesting fact that the more communication channels and the more units there are involved, the slower the total systems work. The saying "He travels fastest who travels alone" seems to be appropriate here. If you give, give slowly only! Otherwise the system you are in as a giver becomes unstable!

Fig. 5, which is of the main interest, clearly demontrates the saying's truth: The more partners U_1 tries to satisfy, the lower is his own achievement. USA, do not help all over in the world. You ruin yourself! Paddle your own canoe!

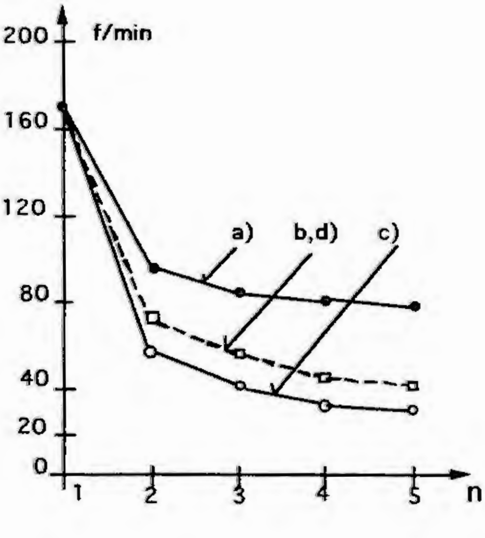


Fig. 4: Oscillation freqncy of the systems with the volitions of Fig. 3. The oscillation frequency is a measure of the speed of action.

The case 2,a), where there is attitude without any help yet, does not damage the prime unit's achievement remarkably. The self–achievement drops from 89% to about 73% if he has attitude toward 4 units (n = 5). It seems that the curve levels out at about this percentage, 73%, independent of how many more units are encountered within the prime unit's attitude. But as soon as the prime unit wants or has to serve others beside looking after himself, his achievement drops drastically; in the case 2,c from 89% down to 18% if observing and serving 4 units.

If, e.g., the minimum self–realization of, say, 30%, is needed for survival, then fulfilling the demand of more than one unit, the prime unit faces destruction: No man can serve two masters: for either he will hate the one, and love the other; or else he will hold the one, and despise the other (Matthew, 6/24).

The case 2,b), the prime unit's recognition of the needs – or demands – of the secondary units is already a burden for U_1, a reduction of its self–realization from 89% to 35% in case of beeing considerate for four secondary units.

To have modest strength of attitude and not wanting to being able to sense the need of too many others, however, does not do much harm to yourself! You just throw the mail of daily begging letters in the waste basket! But don't feel guilty in doing it! You obey a Natural Law.

The situation of prime interest might be the relationship of two units, U_1 and U_2 as husband and wife, in the case 2,c). If both give up (they have to give up) their independency, i.e., their autonomous state, their volitions drop from 8 to 0.78 (by a factor of 10), and their self–realizations (if there is no common goal) drop from 89% to 35% (by a factor 2.5). Qui se marie se bride! He who maries brides himself.

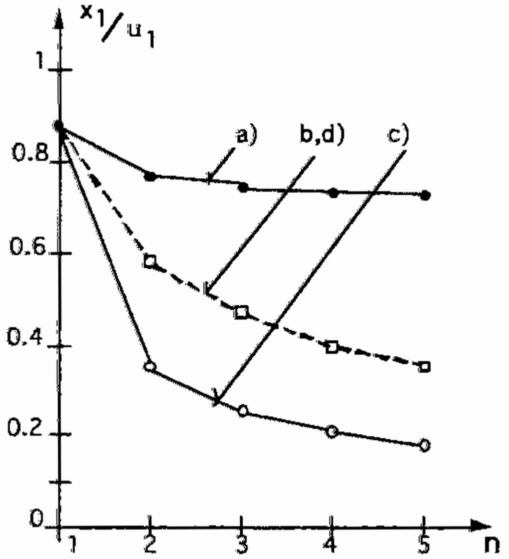


Fig. 5: Goal achievements $\frac{x_1}{u_1}$ of the prime unit for the systems Fig. 2 for increasing number of secondary units (n–1).

Conclusions

In all four considered cases, the model is well in accordance with the saying "Frustra laborat qui omnibus placere studet", i.e., it is true that the achievement of the prime unit, i.e., what he wants to do – or is supposed to do – for himself, not only becomes damaged by pleasing everybody, but, in more detail, that he becomes further reduced the greater the number of the secondary units the prime unit is requested to please. The worst case, (2,c), the destruction of the prime unit, occurs when the prime unit observes the demands and gives accordingly, and if he strives - or is forced to please - more than one secondary unit. One has to keep in mind that very generally each living being requires more than nature - or the environment - can provide. And if you give fast, the second request comes right away: Bis dat qui cito dat; He gives twice who gives fast. The less and the slower you give, in better a state you are with yourself. Nature does not care about our „ethics" and social rules. She goes her own way! So, don't give too much, look after your own survival!

References

1. Starkermann, R., „Modelling Invariances of Social Behaviour", Proceedings of the 38th Annual Meeting of the ISSS, 1994, Pacific Grove, California, USA, Vol. I, pp. 687-715.

COMPUTER SUPPORT FOR MACROECONOMIC DECISIONS

Janusz Babarowski[*], **Jakub Gutenbaum**[*], **Michal Inkielman**[*]
[*] Systems Research Institute, Polish Academy of Sciences, 6, Newelska, 01-447 Warsaw, Poland

Abstract. A macro model of the transition processes in the Polish economy is presented. The structure of the model and a number of simulation scenarios are considered. Some results of simulation of macroeconomic processes are described.

Introduction

A simulation dynamical model of macroeconomic processes has been worked out at the Systems Research Institute of the Polish Academy of Sciences. This model takes into consideration most of specific features of an economy in transition. The model aims at simulating main relations among the basic macroeconomic variables and forecasting of the transformation process for economic decision support.

The main area of interest is prices formation (inflation) process in the output market. Resulting inflation depends on many factors, such as budget deficit, credit supply, consumption demand changes and rise of production costs (cost-pushed inflation). Supply of and demand for output are strictly connected with many macroeconomic variables. This is a reason why a wide spectrum of economic phenomena is to be modelled.

From the mathematical point of view, the model is a set of algebraic equations describing relations among variables at a given time moment and a set of difference equations with time delays. The level of model complexity is medium. It includes about 20 state variables, 100 output variables and statistical indices, 50 exogenous input variables and time dependent parameters. Ten to twenty parameters have to be adjusted in the course of identification process.

Computer program for model solving is of open type. Additional possibilities of closing optimization loops for selected decision variables as well as iterative coordination of submodels exist. Moreover, it is possible to introduce additional dynamic models describing decision variables. The following output variables, among others, are observed: price index, liabilities, net output, costs. This choice makes it possible to compare the output of model and values measured in real economy.

The paper presents the basic assumptions introduced, the structure of the model and some simulation results.

General structure of the simulation model

The model describes the most important macroeconomic processes, such as production process with its cost and income balances, pricing mechanism and investment strategy. Moreover, banking system balance and foreign exchange balance are taken into account. The general structure of the model is shown in Fig. 1.

Properties of the model are determined by the dynamics of balances of the following feedback loops: 1) capital - output - profit - investment - capital; 2) production - labour - wages - demand for output - production; 3) production - wages - deposits - investment - capital - production. In these loops, variables of resource type are as follows: fixed capital, total amount of deposits, total amount of credit and output stocks. All of them are considered as the state variables in each simulation experiment. The choice of the remaining state variables depends on hypotheses concerning price adjustment mechanism in the output market, employment dynamics, indexation of wages and control principles applied.

The model is a medium term one. It is analyzed in 4 to 8 years time horizon. The sampling period is one quarter. Time scale of the model follows from the relatively rapid changes of market equilibrium conditions and relatively slow changes of property structure, production structure, labour productivity, investment, etc.

Aggregation level of the model is a compromise between the necessity of distinguishing the main economic agents, such as private and government enterprises (privatization process), consumers, banking sector, budget and requirement of having the simulation process under control.

It is assumed that the output market is one product with one price. Moreover, it is assumed that financial flows are homogeneous and prices are represented by generalized price level p, which corresponds to GNP deflator, or by inflation rate f.

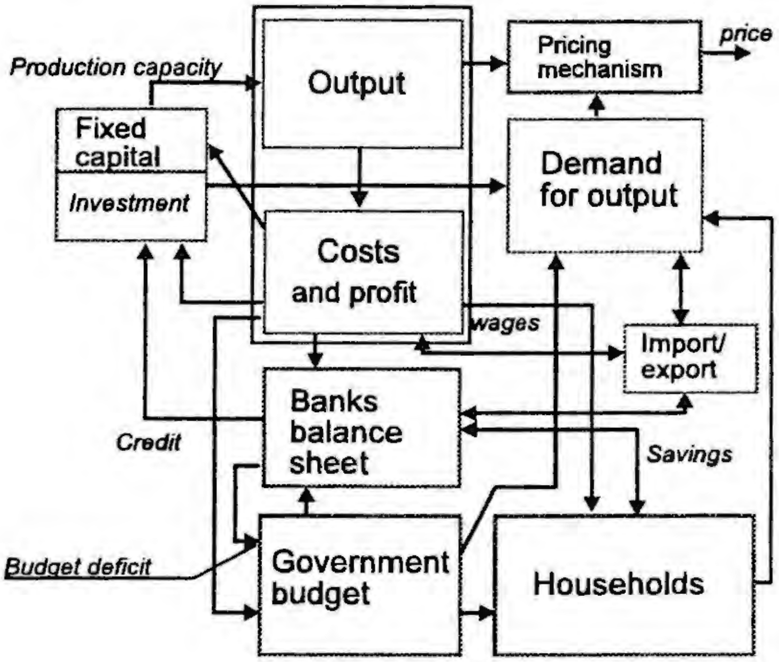


Fig. 1. The structure of the model

The flexibility of the model makes it possible to investigate different price adjustment mechanisms, such as static model of equilibrium price, dynamical model based on excess demand hypothesis, markup model based on production costs under given profit and profit maximisation model. The simplest, numerically stable model based on inertia adjustment rule is considered. The price level in this model is described by the following equation

$$p_t = p_{t-1} + f(Y_{dt}, Y_{st}, p_{t-1}); \quad p_0 = 1 \qquad (1)$$

where Y_{dt} and Y_{st} are aggregate demand for and supply of output at the time instant t, respectively.
Alternatively the following models are considered
- equilibrium model:

$$p_t = \{p: Y_{dt}(p) = Y_{st}(p)\} \qquad (2)$$

- cost model:

$$p_t = f(p_t^p(N_t^p), p_t^g(N_t^g), p_{t-1}) \qquad (3)$$

where N_t^p, N_t^g - unit costs for private and state enterprises,
p_t^p, p_t^g - prices desired by private and state enterprises.

Some variables may be exogenous or decision or output variables. The choice of exogenous variables depends on the questions formulated by the model user as well as on different possibilities of representing of decision makers. A decision maker can be described by a decision rule which becomes an element of the model.

For instance, it may be the Central Bank and its decision referring to interest rate. It also may be a "distributed" decision makers, representing e.g. producers and their decisions on supply of output or on level of wages.

Simulation results

A series of charts which illustrating computational possibilities of the simulation program, various types of results as well as qualitative and quantitative characteristics of preliminary adjusted model will be presented. All charts follow from simulation experiments carried out in the accordance with the basic scenario (1990-1993 -1996) and its extrapolation up to 1998. The basic scenario was modified to introduce some disturbances and decision variables changes; also in retrospective way (for periods with known real data).

The influence of budget deficit and credit policy

In order to study expected real GNP and anticipated inflation as functions of planned budget deficit and planned new credit, a series of about 100 simulation experiments with 3 years horizon has been made. As a result Fig.2 is obtained. Levels of constant values of real GNP obtained in three year period are indicated by solid lines. Levels of constant values of anticipated annual inflation after 3 years are indicated by dashed lines. The maximum of GNP occurs. The initial point of simulation is also marked. A wide flat area of the GNP values higher then the initial one is observed. Such a situation makes it possible to forecast the expected GNP value higher then the initial one. However, maximisation of the GNP with respect to the planned budget deficit and new credit is hazardous procedure because of large sensitivity of the GNP function with respect to parameter changes. For example, the increase in producers propensity to enlarge production leads to the substantial growth of the GNP values in the flat area and to significant shift of the maximum point toward greater new credit and lower deficit values.

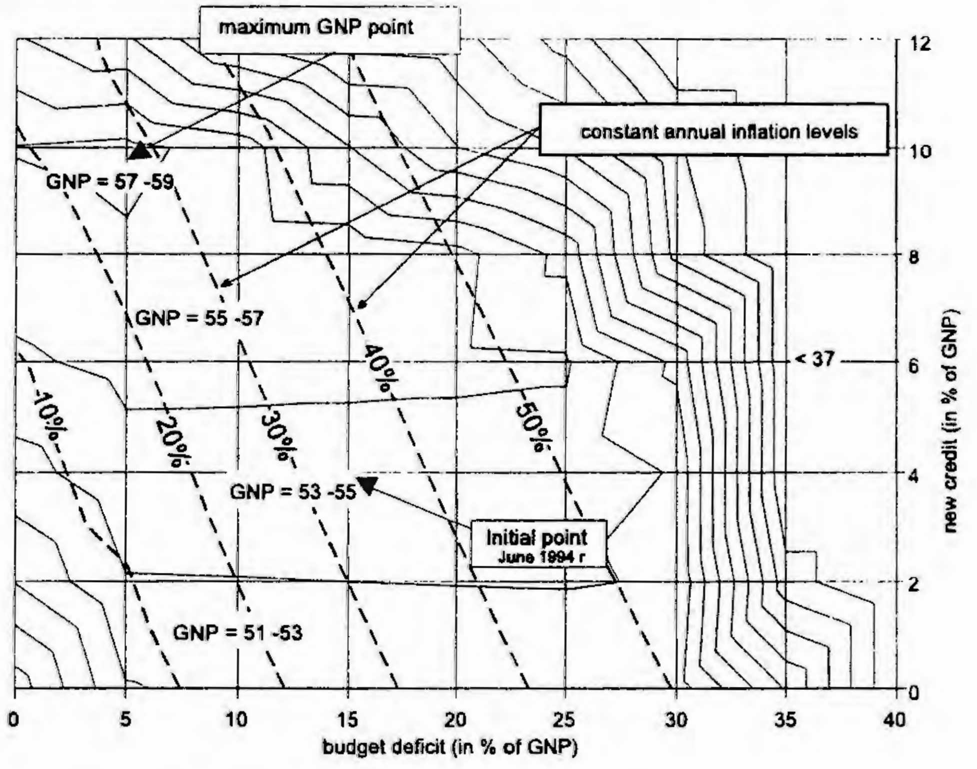


Fig. 2. The real quarterly GNP as a function of budget deficit and new credit (under assumption that during 3 years decision variables values are constant)

Larger number of parameters taken into account when determining sensitivity improves the forecast validity of the GNP growth. Unfortunately, it leads to rapid enlargement of computational efforts and to diminishing of clarity of results.

This way of utilizing the model allows us to avoid high cost investigations of a real process during construction of a black - box type model, being another type of approximation of reality then the previous model itself. The computation time for 150 scenarios, each of them generates 20 quarter values of about 200 variables of the modelled process, is about 3 minutes on the PC486 computer.

The flat area of the investigated GNP function occurs in the realistic range of the control variables: 0 - 15% of GNP for the budget deficit and 0 - 12% of GNP for investment credit. In this range it is difficult to obtain GNP growth greater then 10% during 3 years (out of this range the GNP rapidly diminishes). However, the assumption of 5% growth of producers propensity to enlarge production gives 2 - 3 times greater increment of GNP. Moreover, it does not require augmented financial supply (hence, lower level of inflation and a higher profit rate is obtained).

In Fig. 2 the restrictive influence of shortage of financial resources is shown. Insignificant budget deficit and low new investment credit results in diminishing demand for output and as a consequence - the fall of the GNP under braked inflation. For large credit and budget deficit a decrease in the GNP is interpreted as a result of a destructive power of high inflation.

The simulation model can be also used to investigate the impact of interest rate changes on the credit level. In Fig. 3 the influence of interest rate on some economic indices is shown. It can be seen that long term effects are substantially different from the short term ones.

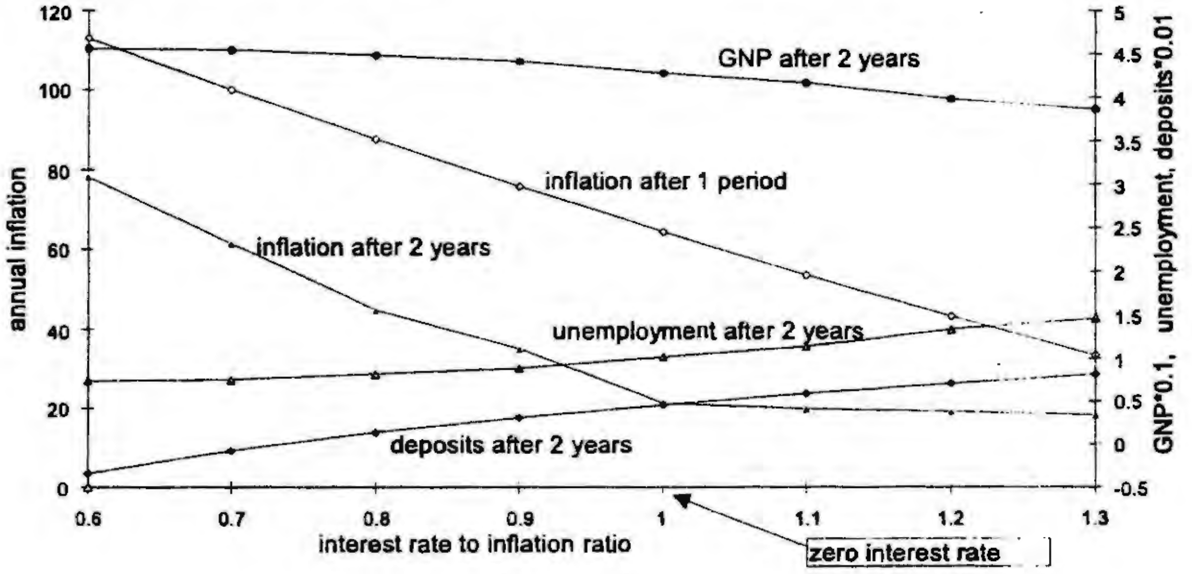


Fig. 3. Selected economical indices as functions of the real interest rate

Summary

The described mathematical model of inflation process was implemented using EXCEL spread-sheet package. As a result, a macroeconomic, medium - term, dynamical and nonlinear simulation model of important inflation factors has been obtained. The model was evaluated using data of the Polish economy for 1 - 3 year forecast horizon.

The main endogenous variable is the price level; the main exogenous variables are: bank rate, supply of commercial credit, exchange rate, export level, effective indexation parameter of wages, budget deficit, privatization flow and labour resources.

A number of simulation experiments was carried out. Detailed investigation of different simulation scenarios allows us to distinguish some important parameters, which affects the process simulated. If parameters such as propensity to saving and preferred production level do not change, then the model is characterized by relatively autonomic behaviour in a wide range of control decisions. This property of the model can be observed in the case of simulating GNP as the function of budget deficit and new commercial credit (Fig. 2.). A flat area occurs for a wide range of control variables. However, when propensity to enlarge production is observed, then a significant growth of GNP takes place. This growth does not require any financial support, in other words the level of inflation is lower and enterprises profits are higher. This property of the model illustrates the opinion that the positive will is a driving force in economy that cannot be replaced with money. In some cases the model characteristics depend also on the effectiveness of fixed capital, i.e. the ratio of production capacity to fixed capital, and on the ratio of GNP to global output.

The simulation model can be used as a tool for forecasting of the most important development indices such as: GNP, unemployment, consumption level, inflation rate, budget deficit, total amount of debts and savings and the balance of payments, e.g., in the Fig. 4, simulation results of the real output and inflation forecasting are presented.

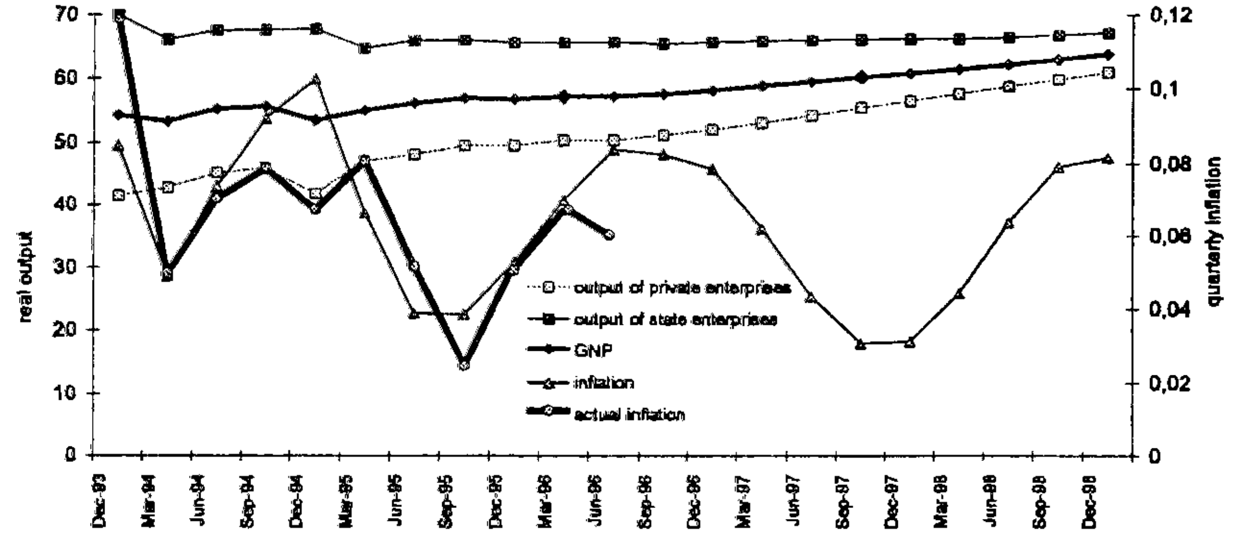


Fig. 4. Example of results of simulating the real output and inflation in Poland, 1994-1998

References

[1] BABAROWSKI J., GUTENBAUM J., INKIELMAN M., *Equations of Basic Markets for Inflation Modelling*, in Transition to Advanced Market Economies, edited by J. Owsiński, J. Stefański, A. Straszak, The Association of Polish Operational Research Societies, Warsaw 1992, pp. 223-232.

[2] BABAROWSKI J., GUTENBAUM J., INKIELMAN M., *Inflation Modelling at the Macro Level*,. in Problems of Building and Estimation of Econometric Models; Part B, edited by W. Welfe, W. Zatoń, Committee of Statistics and Econometrics; Polish Academy of Sciences, Łódź 1994, pp. 9-28.

[3] BABAROWSKI J., GUTENBAUM J., INKIELMAN M., *Simulation model for macroeconomic decision support*, in Decision Support. Expert Systems (in Polish), edited by R. Kulikowski, L. Bogdan, Polish Academy of Sciences;,, Warsaw 1995, pp. 57-64.

[4] GUTENBAUM J., BABAROWSKI J., INKIELMAN M., *Mathematical Modelling of Inflation Process in Restructurized Economy* (in Polish). Report of Research Project KBN No. 1 1062 91 01, Systems Research Institute, Warsaw 1995.

[5] INKIELMAN M., *Modelling and computer simulation of transition processes in macroeconomy (the case of Poland, 1990-1994)* (in Polish), Systems Research Institute Bulletin, No 3, SRI PAS, Warsaw 1995

[6] BABAROWSKI J., GUTENBAUM J., INKIELMAN M., *Modelling of an economy in transition (some* COMPUTER *simulation results)* in Proc. of Macromodels'95, Macromodels and Forecasts, edited by W. Welfe, M. Majsterek, Committee of Statistics and Econometrics; Polish Academy of Sciences, Łódź 1995, pp. 29-43.

MOTIVE-OBJECT MODEL OF TASK DECISION PROCESS

P.I. Sosnin
Professor, Ulyanovsk State Technical University
Ulyanovsk, Severny Venetch, 32, 432000, Russia
e-mail: SVS @ NITPT.PTI.SIMBIRSK.SU

Abstract

This is to show the research results of correlation of inner motives and objects of a person with the other characteristics of decision-making process. The sphere of researches lies in expert systems and artificial intelligence in CAD/CAM systems.

Introduction

An efficiency and result achievement in decision-making process depends greatly on methods and means used by a person on a meta-task level. This fact demonstrates the wide development of expert systems. The basis of any expert are in some version of productions. The presentation of productions and their systems is directed on classified query which is limited by one-level implication relation between if-condition and then-components. The presented motive-object model transforms the work with productions to three-level presentation. This is more adequate to the decision-making practice and has a useful reverse influence on the decision-making process.

The basis of motive-object model of task decision process

1. An activity of a person is a set of the following "conditioned reflexes":

```
Di(t): as [motives { M } ]
             since [ goals { C } ]
   c           if [ preconditions  { U} ]                      (1)
   h             then [ reaction Rg(t)]
   o              and so [ postconditions { U} ]
   i
   c  -----------------------------------------------------
   e                     were alternatives [{Rp}]
```

which take place in a specific "Person-World" system of cause-effect relations (motives, goals, pre/post-conditions, alternatives).
The practice of reactions proved an efficiency of presentation Di(t) as a unit of experience Ei, grouping these units into a system of experience E=S({Ei}) and the use of such experience E for creating the following reaction Doi and the second reaction according to the gotten sample.

2. The cause-effect character of the reaction is based on the multi-level process of "situation-experience"comparison aimed on the search of an appropriate unit of experience according to the "sample" situation or on the creating the new one (it will be included into the system of E). At any case the reaction is the specific type of reality which shows itself through the set of physical and mental processes. It is necessary to understand that the laws of nature are added here by the rules of an activity. The rules of an activity are formed and studied implicitly (as a result of an occasion "discovery" and successful application) or explicitly (when they are extracted from research, real practice or when the rules are invented and implemented in life).
A practical research and a use of such reality can be the principal point in development of a new informational technology. This reality together with an activity of a person is open for the experimental and theoretical research.

3. In activity processes we can mark out the "task situations". Decision-making process generates special objects "task decisions". The "level of meta-task" fixes the logic structure of an activity. The "level of a task" fixes the results in time.

4. A reaction Diz(t) begins from the moment of task or problem definition. Using the logic (1) we ought to know that:

- the task is a sort of question and a person has no answer in his experience. This situation requires inclusion decision-making methods into the process of task solution;
- the essence of the problem is in disagreement between the new situation and the old experience;
- it is enough knowledge to find a problem decision.

The rules of problem decision take the separate place in the rules of an activity. Their explicit application, for example in CAD/CAM systems, requires creation, use and development of special computer tools as an efficient help system. These rules play their role not only on the task level but also on the meta-task level. In the first case they "switch" during decision-making (f.e. through engineers reaction included into decision method). In the second case they are realized through reactions in research process and creation the method of task decision.

5. The registered protocols of the mental work of a person can play the role of the primary experimental information about an activity (physical and motive components of his reaction). Adequaty and quality of such protocols mainly depends on WHO and with the help of WHAT measuring tools fixes these protocols and HOW protocols influence on the activity process. The physics and practice of measurement can't exclude "undesired fault" of influence of such tools on the evaluated process. So, this is true to protocolling of the mental work and for this reason the questions "WHO?" and "WHAT?" are principal for experimental research. The answer to "HOW?" must take into consideration the negative and positive components because these are the rules of measurement. in research and practice.

6. The auther has systematically used and studied the mental protocols (during the decision-making) in some CAD/CAM applications and this experience led him to the task decision method with the following principles:

A. Protocol fixing question-answer process of mental work is registered in a form saving cause-effect hierarchical relations between the units (texts) of Q-A protocol structure.

B. Question-Answer structure of protocol. We can name the questions of the following types: "problem", "task", "query" and the answers as "idea", "description", "prescription","decision". These types are interpreted as Q-A model or structure of task decision process.

C. Rational influence of Q-A structures on the process of decision-making is realized through:

- predicate control (specification) of the every Q-A element;
- distribution of Q-A elements according to the logic form (1) and systematization;
- analysis of the current situation (the work shedule) and the results of work.

The rational points are:

- the control of decision-making process (planning and performing of current processes, result control);
- objectivation of the metal work (definition of an operational logic, motives, substantiations, conclusions and proofs):
- support of understanding and definition of its level;
- forming and systematization of an experience, so as the experience of decision-making.

On pict.1 you can see question-answer structure of real problem solution process. The aim of this presentation is to show the general specific logic and dinamics of the process. Dinamics of Q-A process is reflected through the ordered index of Q-A elements (texts). The texts are added to the each related group and generate the total protocol of the process. The fact of definition of argumentation logic during decision-making can be evaluated as a proof version (conclusion) and extraction the true decision route.

7. Q-A protocols and the results of their process and systematization as a theory are informational base of task decision, which has explicit or implicit answers to the essential questions (and meta-questions). The answers must be worked off inspite of a situation and a level of informational network of the process. It is expedient to assign the functions of such specification to the task meta-presentation.

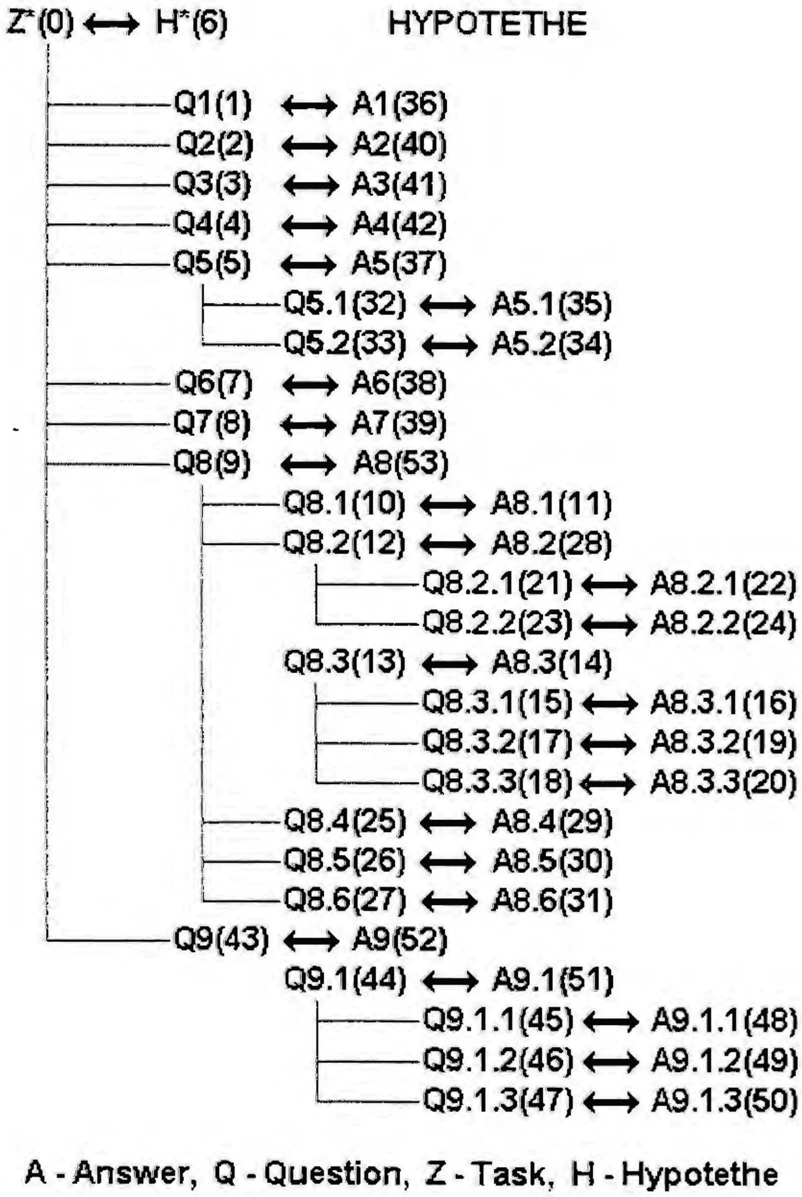
So, for decision-making process the auther suggests a special approach with the creation of the hierarchical cause-effect net in a form called " task meta-model" with the following specifications:

A. Task definition with the corresponding net of cause-effect relations is dinamic object constantly detailed in current work.

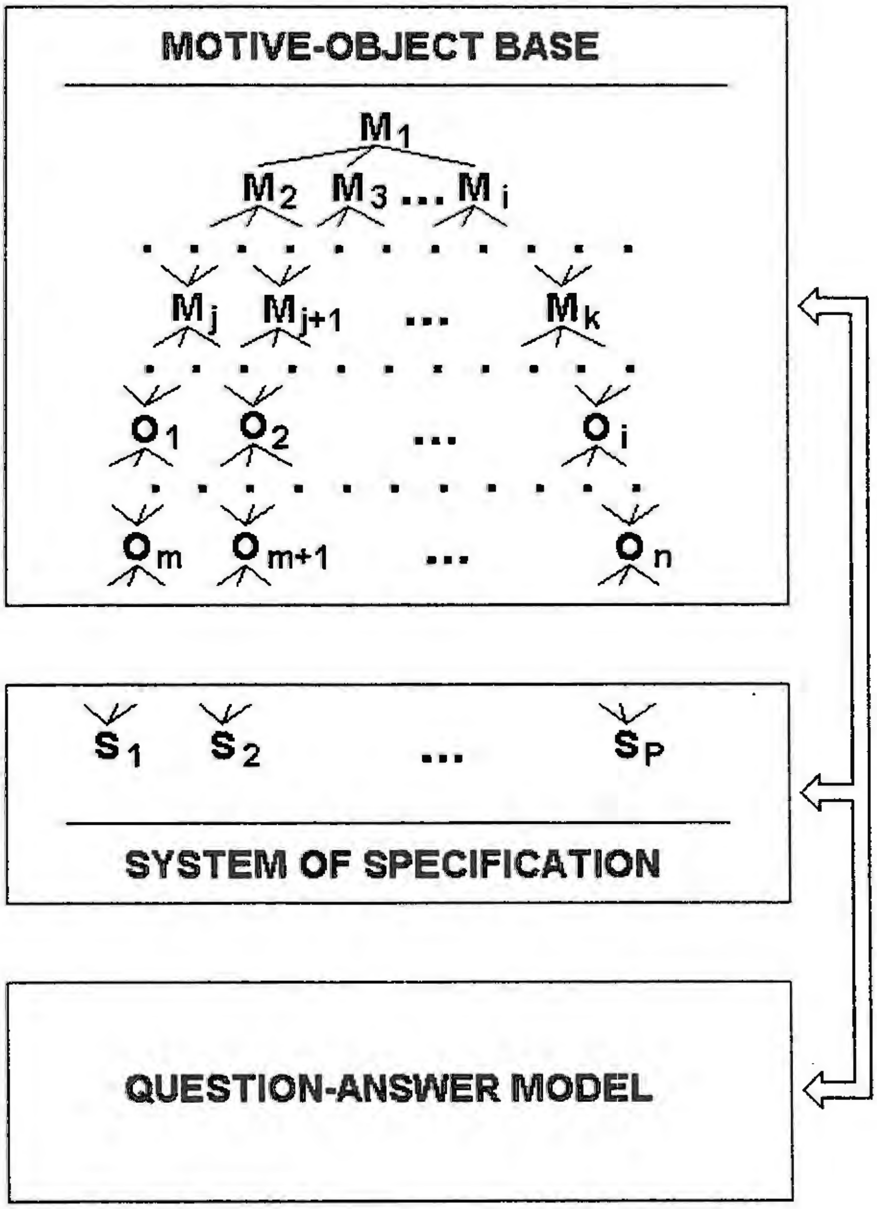
B. The net of cause-effect relations is presented for the every task condition as oriented graph:

The nodes of the graph are motives, aims and specifications. The related connections between nodes show the causes and effects.

C. The next node is included to graph if it plays the role of consequence.



Pic.1 Question-answer structure of real problem solution process



Pic.2 The components of decision-making process

Such model is called **"productional meta-model of a task"** and its main function is an explicit presentation of logic of a task on the meta-task level. The general scheme of the model and its relations with Q-A task structure are shown on pict.2.

An application of this approach is based on constructional work with the motives. Every motive is such "Person-World" relation which:

- controls a human activity for achievment the positive effect in the system "Person-World";
- contains not only the aim component but also psychological one and this fact supposes the quality of characteristics, subjectivity in result evaluation, the level of its achievement, hypothesis character of motive setup.

Summary

Addressing to the described model (its reading, analysis, transformation or development) means an explicit transition to the meta-task level which can bring positive results. Productional model of task definition is valuable for decision-making and for the result achievement. Its including into the results of work:

- increases the faith to result (the method of decision as a result of search and realized routes);
- helps "to suit" the task decision to the new task conditions;
- is open for use in search of the compatible units of experience;
- helps to understand the problem (this fact increases an efficiency of human activity)
- contributes to mutual understanding in a group and coordination of activity of workers with the related motives and goals;
- is highly valuable in education and experience exchange between professionals.

The approach adds "conditioned reflex" on task situation by "switching" the mechanism of explicit conscious forming and proved cause-effect network of decision-making on meta-task level.

References

1. Moric K. Acquiring domain models - Int.J.Man-Machine Studies. Vol.26, N 1, p. 93-104, 1987.

2. Sosnin P.I. Technology of Imitative Activity education. - Interactive systems: The problems of human-computer interaction, thesis of international scientific conference, part 2, p.31-35, Ulyanovsk, 1995.

MODELS OF COMPLEX ANTHROPOCENTPAL OBJECTS FOR THE SYSTEM DESIGN OF ON-BOARD ALGORITHM AND DISPLAY SUPPORT.

Boris E.Fedunov
Laboratory of system researches of the State Research
Institute of Aviation Systems (Gos NIIAS),Moscow, Russia.
e-mail: fed@tm.niias.msk.su

Abstract.

Environment and function process models of anthropocentralobjects are discussed. They are used to develop specifications and structures of on-board algorithms and crew's activity algorithms for system designing functionally-complete on-board algorithm and display support to the operation of an anthropocentral object. As an example of such an object an aircraft can be taken to consider.

Introduction.

An anthropocentral object (Anth/object) is a totality of onboard measuring and operating equipment (the on-board M&OpEq), a system of on-board computer algorithms (Com/Alg), crew members that realize their activity algorithm (Ac/Alg) through an informational-controlling field (ICF) of the cabin, a carrier of the above-mentioned items providing their integrity and necessary physical conditions for their functioning. The crew takes a leading part in Anth/object's functioning.

The totality of on-board Com/Alg, the information on the ICF displays, the ICF controls signals, and crew Ac/Alg - all these form an on-board algorithm and display support (A&DS).

A common semantical model of an anthropocentral object.

In the analysis of Anth/object's function [1,3,4] process three global levels of control (GLC) are always distinguished on the board of the Anth/object: GLC-1,GLC-2 and GLC-3.

GLC-1: operative fixing of the current purpose of the object function;
GLC-2: the choice of the way of achieving the fixed purpose;
GLC-3: the realization of the chosen way.

The on-board A&DS and the crew should solve all the tasks of the GLC-1,-2,-3.

There is a notion of a function session (func/session) existing with respect to complex Anth/objects, and actually being defined in technical documantation. For each type of Anth/objects there is a finite multitude of the func/session.

Informational preparation for the func/session consists of instructing the crew to use an a priori model of the func/session and including the information on the model into the on-board A&DS of the Anth/object.During the func/session the Anth/object should correct the func/session model successively and efficiently (this refers to GLC-1 and GLC-2 tasks).

The successive realization of the apriori model being permanently corrected ing the apriory func/session model is implemented by the Anth/object through the solutio of GLC-3 tasks.

Structuring the environment and the process of functio ning of the Anth/object.

The purpose of studying structures of the environment and of the process of functioning is to limit and describe a domain of the outer world where an Anth/object will operate.

The following theorems are true.
Theorem 1. All the work (W) of the Anth/object can be presented through the semantical network (S/N) of the problem situations (PS) with a finite number of the tops. Each func/session is identical to a certain fragment of this S/N.
W = S/N (PS); func/session = the fragment of P/N.

The PS is a part of the work W that is functionally closed and very important for GLC-1. For designing A&DS PS is preset by a technical text which includes: a description of the PS purpose, the multitude of the permissible ways of achieving the purpose, hampering and assisting conditions for the purpose achievement.

For designing it is very important to classify typical situa tions:

* accoding to conditions arisen: the PS-s imposed by exterior and interior (on board) situations are indicated by us as PS (A), and the PS-s expedient for the current conditions of the func/session are indicated as PS (B).
* according to the rate of situation changes: the PS-s with "severe" time limits are indicated as the problem quickly-changed situations (PQS), and other situations that have no time limits are indicated as PS.

Theorem 2. Each PS can be presented through PS= S/N (PbS/S) where there are all permissible ways of achieving the fixed current aim.

Theorem 3. The algorithm structures of the PbS/S problems can be researched with the optimization tasks of a certain type.

Theorem 4. Each PS contains the finite number of significant events. The time moment of their emergence can be forecasted with the help of mathematical models of the PbS/S in the A&DS.

The multitude of significant events is partially regulated by the reason - consequence ratio.

The on-board algorithms and display support for each GLC.

Recent Anth/object researches are aimed at supporting GLC-3. For this GLG designers have mastered the certain algorithm structures (we call them conventional algorithms).They meet the GLC-3 peculiarities. The distinguishihg peculiarity of the GLC-2 tasks and, especially of the GLC-1 tasks, is the necessity of using incomparably larger and qualitatively new " knowledges of the world". For these levels it is necessary to use new algorithm structures containing elements of artificial intelligence. The on-board operative advising expert systems (OAES) are algorithm structures of a new class that are able to provide solving GLC-2 tasks just tomorrow. Their specific characters are as follows: the OAES knowledge base should be constantly corresponded to the crew's knowledge of the current PS; the dialogue, practically lacked, between the crew and the OAES; the on-board computer possibilities are limited. The OAES knowledge base has the following structure: the rules for the PbS/S activization (theorems 2 and 4), the mathematical models for significant events that are important for this PS (theorems 3 and 4), a number of rules for each PbS/S containing in this PS. The OAES specifications of the Anth/object are presented in Fig.1,and as for their structure it is described in literature [2].

For the CGL-1 the awareness of situations should be provided (theorem 1). Under this we understand the possibility to discover a coming PS(A) (or the PS(A)) already appeared on the Anth/object, and to determine the expediency of the realization of the certain PS(B). During the preliminary func/session the Anth/object's board and its crew are being prepared mainly for the GLC-1 task solution (the fragment of the S/N (PS) expected in the future func/session).The design problems are as follows: the search for the cognitive images of showing the environment situation to the crew to "detect" new PS, and to relate its purpose to the current PS (through experiments, factor analysis); the on-board development of significant stimulus signals for the crew to detect PS(A) (through on-board conventional algorithms, experiments); development of the theory and technology of designing OAES, " PS Appointing " (through applied and fundamental science).

For system designing on-board algorithm schemes (SBA), and the graphs of the crew's decisions (CDG) are made successively, from Pb S/S-s, PS-s to the multitude of all the PS. For the PQS-s the Ac/Alg structure is formed of "to speak-to think" decisions and simmultational decisions (for other PS-s heuristic decisions are also used); the crew's operation as a part of the watching system; and realization of making decisions.

The semantical structure of the on-board A&DS is presented in Fig.1.

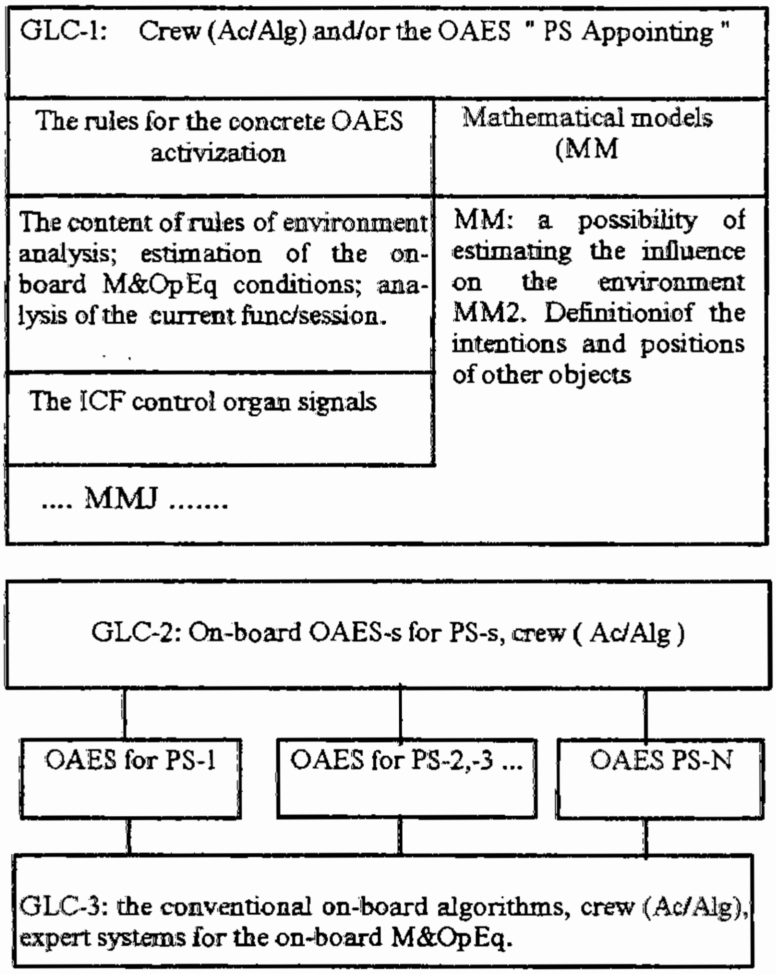

GLC-1: Crew (Ac/Alg) and/or the OAES " PS Appointing "
The rules for the concrete OAES activization
Mathematical models (MM
The content of rules of environment analysis; estimation of the on-board M&OpEq conditions; analysis of the current func/session.
MM: a possibility of estimating the influence on the environment MM2. Definitioniof the intentions and positions of other objects
The ICF control organ signals
.... MMJ ....
GLC-2: On-board OAES-s for PS-s, crew (Ac/Alg)
OAES for PS-1
OAES for PS-2,-3 ...
OAES PS-N
GLC-3: the conventional on-board algorithms, crew (Ac/Alg), expert systems for the on-board M&OpEq.

Fig.1. The on-board A&DS semantical structure for the Anth/object.

Conclusion.

1. The formmal (but natural for the designers) conception of the function environment of Anth/object through PS, PbS/S, and the onboard semantical structure through GLC-s gives the constructive basis for the development of functionally complete objects.

2. Today out of three GLC presented designers can provide mainly the GLC-3. We refer the success in mastering GLC-2 to the OAES development. As far as GLC-1 is concerned it is necessary to continue the search for adequate structures of the needed "knowledge of the world" and "put-out" procedures of OAES "PS Appointing".

References.

1. Моисеев.Теория управления и проблемы “Человек-окружающая Среда”. Вестник Н СССР, №1, 1980.
2. Федунов. Бортовые оперативно-советующие экспертные системы. Журнал Известия РАН. Теория и системы управления. №5, 1996 г.
3. Позняков, Б.Е. Федунов. Основы информационной интеграции бортовой аппаратуры, МАИ, 1993 г.
4. Fedunov, T.A.Kondrikova "On-board Algorithms and Displaed Information for Anthropocentral System: Problems and Methods of Systematic Design.1993. JE Systems Man and Cybernetic Conference Proceedings. Vol.2. Le Tougut. France.

ILL-POSEDNESS ASPECTS OF SOME NONLINEAR INVERSE PROBLEMS

G. Fleischer and B. Hofmann
Technical University Chemnitz-Zwickau
D-09107 Chemnitz, Germany

Abstract. In this paper we deal with aspects of classifiying the ill-posedness of nonlinear inverse problems based on the discussion of some examples. In particular, specific parameter identification problems in differential equations and their ill-posed linear components are under consideration. A new approach to the treatment of ill-posedness properties for multiplication operators completes the paper.

Introduction

In the last few years the accelerated coupling of applied mathematics, natural sciences and engineering has enormously stimulated the inverse problem theory and practice. The rapidly growing number of inverse problems occurring in numerous practical situations also implied a qualitative and quantitative development of the mathematical modelling of inverse problems. This process was accompanied with a large number of new articles and books on the mathematics of inverse problems from analytical and numerical viewpoints (see, e.g., Anger et al. [1], Engl [4], Engl, Hanke and Neubauer [5] and Kirsch [9]). In the framework of many authors the different aspects of ill-posedness for linear and nonlinear inverse problems were intensively studied. For the choice of appropriate methods and approaches aimed at solving inverse problems in a unique and stable manner, the characterization of the kind and degree of ill-posedness of any specific inverse problem plays an important role. In the class of nonlinear inverse problems, written as an operator equation

$$F(x) = y, \quad x \in D(F) \subset X,\ y \in Y \tag{1}$$

in Banach spaces X and Y, the ill-posedness behaviour may depend on the location of the solution point x^* under consideration inside the domain $D(F)$ (cf. [6] and [7]), whereas linear inverse problems

$$Ax = y, \quad x \in X,\ y \in Y \tag{2}$$

have a global degree of ill-posedness associated with the smoothing properties of the linear operator $A : X \to Y$ characterizing the corresponding direct problem (cf. [8]).

We are going to discuss some aspects of ill-posedness based on an example concerning a parameter identification problem in ordinary differential equations. This example will be given in the following paragraph. Identification problems of this type were treated by many authors (cf., e.g., Banks and Kunisch [2] and Colonius and Kunisch [3]). It can be seen that the mentioned inverse problem leads to a nonlinear operator equation (1). In most cases, we will choose pairs of separable Hilbert spaces X and Y for our considerations. For the example we will show that the ill-posedness properties of the nonlinear example can be verified by analyzing ill-posed linear problems (2), which were obtained as decomposition components of the problem (1). Following the classification of Nashed in [10] we distinguish the often treated case of linear ill-posed problems with compact operators A and the fewer ill-posed form, where A is non-compact, but the range $R(A)$ is not closed. We will give some contributions to the discussion of this latter type in a further paragraph. In this context, we focus our attention to multiplication operators.

An ordinary differential equation problem

To motivate our ideas presented below we consider as an example the following ordinary differential equation:

$$-\frac{d}{dt}\left(q(t)\frac{d}{dt}u(t)\right) = f(t), \quad t \in (0,1), \tag{3}$$

where the parameter function $q \in Q$, the state function $u \in U$, and the right-hand side $f \in V$ are elements of Banach spaces Q, U and V, respectively, which will be specified below. Moreover, for parameter functions $q \in Q$, we want to denote by $L(q) : D(L) \subset U \to V$ the operator defined by

$$[L(q)u](t) := -\frac{d}{dt}\left(q(t)\frac{d}{dt}u(t)\right). \tag{4}$$

Then the differential equation (3) can be rewritten as an operator equation

$$L(q)\,u = f. \tag{5}$$

We should note that the domain $D(L)$ of the linear differential operator L is chosen in such a way that the elements of this domain automatically satisfy imposed boundary conditions, as for example

(i) (homogeneous) Dirichlet conditions, $u(0) = u(1) = 0$,

(ii) (homogeneous) Neumann conditions, $u_t(0) = u_t(1) = 0$,

(iii) or co-normal conditions, $q(0)u_t(0) = \beta_0$, $u_t(1) = 0$.

Additionally we assume that $q(t) \geq \underline{q} > 0$, $t \in [0,1]$. The aim of our considerations is to identify the parameter function q from observations of the state u. Setting $X := Q$, $Y := U$, $x := q$ and $y := u$, this corresponds with the solution of a nonlinear inverse problem (1), where F expresses the parameter-to-state mapping $q \mapsto u$. Note that the parameter-to-state mapping F is nonlinear in general, although the corresponding differential equation (3) is linear. Taking into account the bilinear structure of the differential equation we formulate a lemma which can be found in a similar form in Tautenhahn's paper [12]:

Lemma 1 *For Banach spaces Q and U let $\hat{L} : Q \times U \to U^*$ be a bilinear operator, where U^* is the dual space of U. Then for the operator equation*

$$\hat{L}(q,u) = f, \quad q \in Q,\ u \in U,\ f \in U^* \tag{6}$$

we assume the existence of positive constants M and m with

$$||\hat{L}(q,u)||_{U^*} \leq M||q||_Q||u||_U \quad \forall q \in Q,\ \forall u \in U, \tag{7}$$

$$(\hat{L}(q,u),u)_{U^*U} \geq m||u||_U^2 \tag{8}$$

($(\cdot,\cdot)_{U^*U}$ is the duality product).
If we denote by $F : Q \to U$ the parameter-to-state operator with $u := F(q)$, then the following assertions hold:
1. F is Fréchet-differentiable with $F'(q) : Q \to U$.
2. If we define $L(q)\,u := \hat{L}(q,u)$ for all $q \in Q$, then the Fréchet-derivative can be expressed by

$$F'(q^*)\,q = -(L(q^*))^{-1}\hat{L}(q,u^*) \tag{9}$$

for elements q^* and u^* satisfying $F(q^*) = u^*$.
3. Moreover, we have the estimation

$$||F(q) - F(q^*) - F'(q^*)(q-q^*)||_U \leq \frac{M}{m}||q-q^*||_Q||u-u^*||_U \tag{10}$$

whenever $F(q) = u$.

Using Sobolev spaces we may apply this lemma to our operator equation (5), where we set $Q := X := H^1(0,1)$, $U := Y := H_0^1(0,1)$, $U^* := Y^* := (H_0^1(0,1))^* = H^{-1}(0,1)$ and $\hat{L}(q,u) := L(q)u$. Then it can easily be proven that the assumptions of Lemma 1 are fulfilled. The main result of this lemma is for our differential equation problem the estimation

$$||F(q) - F(q^*) - F'(q^*)(q-q^*)||_{H_0^1(0,1)} \leq k||q-q^*||_{H^1(0,1)}||u-u^*||_{H_0^1(0,1)}. \tag{11}$$

Moreover, by modified specifications of the Hilbert spaces under consideration the assertions of Lemma 1 can be generalized to other spaces. First, if we choose $U := H^1(0,1)$, but with a domain $D(L) := H_0^1(0,1)$, then (10) also holds for all $u = F(q) \in D(L)$. As a consequence of Friedrichs' inequality we then have

$$||F(q) - F(q^*) - F'(q^*)(q-q^*)||_{H^1(0,1)} \leq \tilde{k}||q-q^*||_{H^1(0,1)}||u-u^*||_{H^1(0,1)} \tag{12}$$

for all u from $H^1(0,1)$ with imposed homogeneous Dirichlet conditions. A generalization of inequality (12) to other boundary conditions is possible. On the other hand, one can shift such an estimation along the Sobolev scale. For example, if we take $f \in L^2(0,1)$ and u from $H^2(0,1)$ plus boundary conditions $D(L) := H^2(0,1) \cap H_0^1(0,1)$, this implies the inequality

$$\|F(q) - F(q^*) - F'(q^*)(q-q^*)\|_{H^2(0,1)} \leq \bar{k}\|q-q^*\|_{H^1(0,1)}\|u-u^*\|_{H^2(0,1)} \tag{13}$$

for all $u \in D(L)$. Note that in contrast to (11) the last two estimations (12) and (13) are no direct consequences of the lemma. Nevertheless, the proofs are omitted here.

We may now apply the results of Proposition 3 and Remark 2 of [7] to obtain from (12) and (13), respectively, the estimations

$$c_2\|u_t^*(q-q^*)\|_{L^2(0,1)} \geq \|u-u^*\|_{H^1(0,1)} \geq c_1\|u_t^*(q-q^*)\|_{L^2(0,1)} \tag{14}$$

and

$$c_2|u_t^*(q-q^*)|_{H^1(0,1)} \geq \|u-u^*\|_{H^2(0,1)} \geq c_1|u_t^*(q-q^*)|_{H^1(0,1)} \tag{15}$$

for all q in a ball $B_\rho(q^*)$, where ρ does not depend on u. In this context, $|\cdot|_{H^1(0,1)}$ is the $H^1(0,1)$ seminorm and c_1, c_2 are two positive real constants, which may differ in both cases. In order to get such stability estimates, the topology of the data space $U = Y$ must be stronger than the topology of the parameter space $Q = X$. Really, such inequalities cannot be found in the case that the spaces Q and U have the same topology. For a specific counterexample we refer to Example 2.6 in [2]. As a consequence one can state that the identification problem is ill-posed everywhere if the data space has the same topology as the parameter space. The estimations (14) and (15) imply that we get stable reconstructions of q from the smooth data u in a neighbourhood of the parameter q^* provided that the parameter space topology is weaker than the topology in the data space. In such pairs of topologies the parameter-to-data mapping is continuously invertible. If, however, the natural topology of the real data $\tilde{u}$ is not stronger than the parameter topology, we have additionally to solve an ill-posed linear operator equation

$$\mathcal{E}u = \tilde{u}, \tag{16}$$

where $\mathcal{E} : H^1(0,1) \to L^2(0,1)$ in (14) and $\mathcal{E} : H^2(0,1) \to H^1(0,1)$ in (15) are compact embedding operators. Then the nonlinear inverse problem of recovering q from $\tilde{u}$ is decomposed into a linear ill-posed problem (16) and a nonlinear well-posed problem characterized by (14) or (15) (cf. [6]). In such a case the degree of ill-posedness of the whole problem is determined by the decay rate of the singular values of the embedding operators $\mathcal{E}$.

As the inequalities (14) and (15) show, the stable identification of q from u is possible only in weighted L^2 and H^1 parameter space norms. The weight factor is the function u_t^*. Hence, the reconstruction of q from $\bar{q} := u_t^* q$ corresponds to the solution of a linear operator equation

$$\mathcal{M}q = \bar{q} \tag{17}$$

with a multiplication operator $\mathcal{M}$. This equation can be both well-posed and ill-posed. To obtain well-posedness and stability estimations for $\|q-q^*\|$ instead of $\|u_t^*(q-q^*)\|$, it is sufficient to assume that u_t^* is bounded and has no zeros, i.e.,

$$0 < \operatorname*{inf\,ess}_{t\in(0,1)} u_t^*(t) \leq \operatorname*{sup\,ess}_{t\in(0,1)} u_t^*(t) < \infty. \tag{18}$$

Otherwise, ill-posed situations arise whenever u_t^* has zeros. For example, if we have homogeneous Dirichlet boundary conditions for u^*, that means $u^*(0) = u^*(1) = 0$ then u_t^* must have a point t_0 with $u_t^*(t_0) = 0$. In the case that u_t^* has a zero point we can formulate the inequality (15) in such a way that it holds for the H^1 parameter space norm, not only for the seminorm, i.e.

$$c_2\|u_t^*(q-q^*)\|_{H^1(0,1)} \geq \|u-u^*\|_{H^2(0,1)} \geq c_1\|u_t^*(q-q^*)\|_{H^1(0,1)}. \tag{19}$$

The first inequality is trivial. The second can be obtained if we could show that in our case $|\cdot|_{H^1(0,1)}$ is not only a seminorm, but a norm. For this we must show that $|u_t^*(q-q^*)|_{H^1(0,1)} = 0$ implies that $\|u_t^*(q-q^*)\|_{L^2(0,1)} = 0$. Namely, we then have $[u_t^*(q-q^*)]_t \equiv 0$ and $u_t^*(q-q^*) = \text{const}$. Consequently, if u_t^* has a zero and $q - q^*$ lies in $H^1(0,1)$, this constant must be 0. Under such conditions it becomes evident that the H^1 seminorm and the H^1 norm are equivalent. Hence, the estimation (19) holds if u_t^* has zeros.

We see that there occur two different types of ill-posedness in our simple identification problem. On the one hand, there may be to solve an operator equation (16) with compact embedding operator $\mathcal{E}$. Nashed (cf. [9]) calls this ill-posedness of type II. On the other hand, the multiplication operator $\mathcal{M}$ is for no function u_t^* a compact operator (except in the trivial case $u_t^* \equiv 0$). If the operator equation (17) is ill-posed, then we have ill-posedness of type I in Nashed's sense. When the data and parameter spaces are chosen in such a way that we need no embedding, then this fewer ill-posed situation of multiplication operators is characteristic for the whole identification problem.

Multiplication operators

Now we want to analyze multiplication operators $\mathcal{M}$ more in detail. In this context, we would like to use the discussions of the previous paragraph as a good motivation to do so. We define an operator $\mathcal{M} : L^2(0,1) \to L^2(0,1)$ of multiplication with $\varphi \in L^\infty(0,1)$ by

$$[\mathcal{M}x](t) := \varphi(t)x(t), \quad t \in (0,1) \ a.e. \tag{20}$$

It is clear that φ has to be chosen from $L^\infty(0,1)$ to guarantee that the product is still in $L^2(0,1)$. Usually, such a φ is a piecewise continuous function. It can easily be shown that the adjoint $\mathcal{M}^*$ of the operator $\mathcal{M}$ is given as

$$[\mathcal{M}^*x](t) = \overline{\varphi(t)}x(t), \quad t \in (0,1) \ a.e. \tag{21}$$

From this it follows

$$[\mathcal{M}^*\mathcal{M}x](t) = |\varphi(t)|^2 x(t), \quad t \in (0,1) \ a.e. \tag{22}$$

and

$$[(\mathcal{M}^*\mathcal{M})^{\frac{1}{2}}x](t) = |\varphi(t)|x(t), \quad t \in (0,1) \ a.e. \tag{23}$$

Therefore, we can change between these operators as it is usual in the case of compact operators or we can simply assume that φ is a real and non-negative function. This will be done throughout this paragraph. Then $\mathcal{M}$ is a self-adjoint and non-negative operator.

Every self-adjoint operator of multiplication can be written in its spectral representation

$$\mathcal{M}x = \int_0^1 \lambda dE_\lambda x, \tag{24}$$

where the spectral family E_λ is given by

$$[E_\lambda x](t) = \begin{cases} x(t), & \varphi(t) \le \lambda \\ 0, & \varphi(t) > \lambda \end{cases}. \tag{25}$$

If we introduce the multiplication operator with the independent variable by

$$[\tilde{\mathcal{M}}x](t) := tx(t), \quad t \in (0,1) \ a.e., \tag{26}$$

then its spectral family is

$$[\tilde{E}_\lambda x](t) = \begin{cases} x(t), & t \le \lambda \\ 0, & t > \lambda \end{cases}. \tag{27}$$

It is well-known that any function $\varphi(\tilde{\mathcal{M}})$ of this operator can be written as

$$\varphi(\tilde{\mathcal{M}})x = \int_0^1 \varphi(\lambda) d\tilde{E}_\lambda x \tag{28}$$

whenever the integral exists. Obviously we then have

$$\mathcal{M} = \varphi(\tilde{\mathcal{M}}). \tag{29}$$

For compact linear operators A the degree ν of ill-posedness can be defined by the convergence rate of the singular values $s_j(A)$ to zero as j tends to infinity, i.e., as the value ν for which

$$s_j(A) \sim j^{-\nu} \tag{30}$$

holds. Note that such a proportionality need not be valid in general (cf. [8]). For a compact operator A with a degree $\nu = \nu(A)$ of ill-posedness, however, any power A^α has a degree $\alpha\nu$ of ill-posedness.

To generalize this statement to non-compact operators we have to give a definition first:

Definition 2 *Let $\varphi \in L^\infty(0,1)$. Then we define the distribution function $p(\lambda)$ as*

$$p(\lambda) := \text{meas}\{t : \varphi(t) \in [0,\lambda]\}. \tag{31}$$

This is the measure of the pre-image of the interval $[0,\lambda]$. Now we define the increasing rearrangement $\tilde{\varphi}$ of φ by the function

$$\tilde{\varphi}(s) = \sup\{\lambda : p(\lambda) \leq s\}. \tag{32}$$

It is clear that the function $\tilde{\varphi}$ is in general the inverse function of p. Furthermore, $\tilde{\varphi}$ is a monotone increasing function. As it can be seen, this function becomes zero in the point $t = 0$, if the problem is is ill-posed (i.e. φ has zeros). Now we want to assume that the multiplication operator $\mathcal{M}$ is injective. This property is equivalent to the fact that $\varphi(t) > 0$ a.e. in $(0,1)$. In this case we find that $\tilde{\varphi}(t) > 0$ whenever $t > 0$. Therefore we can define the degree of ill-posedness as follows:

Definition 3 *Let $\mathcal{M}$ be the operator of multiplication with the function φ, $\varphi \in L^\infty(0,1)$. Then we define the real value $\nu = \nu(\mathcal{M})$ to be the degree of ill-posedness of this multiplication operator, if it holds*

$$\tilde{\varphi}(t) \sim t^\nu, \tag{33}$$

that means there are two positive constants $\underline{c}$ and $\overline{c}$ such that

$$\underline{c}t^\nu \leq \tilde{\varphi}(t) \leq \overline{c}t^\nu, \quad t \in (0,1) \text{ a.e.} \tag{34}$$

Note that such a constant also need not exist. It is quite obvious that the operator $\tilde{\mathcal{M}}$ of the multiplication with t gets a degree $\nu = 1$ of ill-posedness. Moreover, its αth powers (i.e., the multiplication operators with t^α) have by definition a degree $\nu = \alpha$ of ill-posedness.

Now we want to give a characterization of the degree of ill-posedness by the properties of the function φ. First we need another definition:

Definition 4 *If $\varphi \in L^\infty(0,1)$ and if, moreover, the value α_0 with*

$$\alpha_0 := \sup\left\{\alpha : \inf_{B_\epsilon(t_0)} \text{ess} \left|\frac{\varphi(t)}{(t-t_0)^\alpha}\right| = 0\right\} \tag{35}$$

exists and is greater than 0 for a value $t_0 \in (0,1)$, then we call t_0 a zero point and α_0 its order.

The essential infimum has to be taken over all balls $B_\epsilon(t_0)$ around the point t_0 with sufficiently small radius $\epsilon > 0$. Without to give a proof we present the following characterizing proposition:

Proposition 5 *If $\varphi \in L^\infty(0,1)$ has only a finite number of zero points, then the degree of ill-posedness of the associated operator of multiplication is not greater than the maximum of all the orders of the zero points of φ.*

Finally, we want to give some brief remarks on Tikhonov regularization for multiplication operators. This methods works here in a very simple manner. Namely, we have to solve the equation

$$(\mathcal{M}^*\mathcal{M} + \alpha I)x_\alpha = \mathcal{M}^*y \tag{36}$$

instead of

$$\mathcal{M}x = y. \tag{37}$$

So we have only to solve

$$(|\varphi(t)|^2 + \alpha)x_\alpha(t) = \overline{\varphi(t)}y(t), \tag{38}$$

where $\alpha > 0$ is the regularization parameter. In [11], Neubauer gives a necessary and sufficient condition for the convergence of Tikhonov regularized solutions to the exact solution. For $0 < \gamma < 1$ we find there the following assertion:

$$\|x^\dagger - x_\alpha\| = \mathcal{O}(\alpha^\gamma) \iff \int_0^\mu d\|E_\lambda x^\dagger\|^2 = \mathcal{O}(\mu^{2\gamma}). \tag{39}$$

In the context of this statement we can also give a proposition, which shows connections between the degree of ill-posedness and the order of convergence of Tikhonov regularized solutions:

Proposition 6 *If the degree of ill-posedness of a multiplication operator $\mathcal{M}$ is given by $\nu = \nu(\mathcal{M})$, then the Tikhonov regularized solutions converge for every $x^\dagger \in L^\infty(0,1)$at least with the order $\frac{1}{4\nu(A)}$, i.e.,*

$$\|x^\dagger - x_\alpha\| = \mathcal{O}(\alpha^{\frac{1}{4\nu(A)}}). \tag{40}$$

Note that the Landau symbol"$\mathcal{O}$" is used in a sense that does not exclude the case "o". Norms in this paragraph are always $L^2(0,1)$ norms.

Summary

In this paper, we have presented some new ideas of analyzing the degree of ill-posedness of linear operator equations with multiplication operators. The use of increasing rearrangements played an important role for this approach. By studying some nonlinear inverse problems of identification type the two classes of ill-posed linear operator equations with compact embedding operators and with non-compact multiplication operators occurred as characteristic components for classifying the kind of ill-posedness of the nonlinear problem.

References

1. Anger, G. et al. (eds.), Inverse Problems: Principles and Applications in Geophysics, Technology, and Medicine. Akademie-Verlag, Berlin, 1993.
2. Banks, H.T. and Kunisch, K., Estimation Techniques for Distributed Parameter Systems. Birkhäuser, Boston, 1989.
3. Colonius, F. and Kunisch, K., Stability for parameter estimation in two point boundary value problems. J. Reine Angewandte Math., 370 (1986), 1 - 29.
4. Engl, H.W., Regularization methods for the stable solution of inverse problems. Surveys on Mathematics for Industry, 3 (1993), 71 - 143.
5. Engl, H.W., Hanke, M. and Neubauer, A., Regularization of Inverse Problems. Kluwer, Dordrecht, 1996.
6. Hofmann, B., On the degree of ill-posedness for nonlinear problems. Journal of Inverse and Ill-Posed Problems, 2 (1994), 61 - 76.
7. Hofmann, B. and Scherzer, O., Factors influencing the ill-posedness of nonlinear problems. Inverse Problems, 10 (1994), 1277 - 1297.
8. Hofmann, B. and Tautenhahn, U., On ill-posedness measures and space change in Sobolev spaces. Preprint IP 7, TU Chemnitz-Zwickau, Faculty of Mathematics, Chemnitz, 1996.
9. Kirsch, A., An Introduction to the Mathematical Theory of Inverse Problems. Springer, New York, 1996.
10. Nashed, M.Z., A new approach to classification and regularization of ill-posed operator equations. In: H.W. Engl and C.W. Groetsch (eds.), Inverse and Ill-posed Problems, Academic Press, Orlando, 1987, 53-75.
11. Neubauer, A., On convergence and saturation results for Tikhonov regularization of linear ill-posed problems. SIAM J. Numer. Anal. (to appear).
12. Tautenhahn, U., Tikhonov regularization for identification problems in differential equations. In: J. Gottlieb and P. DuChateau (eds.), Parameter Identification and Inverse Problems in Hydrology, Geology and Ecology, Kluwer, Dordrecht, 1996, 261 - 270.

EFFICIENT DETECTION OF AN INCLUSION IN A CONDUCTOR. *

G. Alessandrini and E. Rosset
Dipartimento di Scienze Matematiche - Università di Trieste
P.le Europa 1, 34100 Trieste - Italy

Abstract. In this presentation we want to illustrate, by the analysis of a specific inverse boundary value problem, a possible approach when dealing with a severely ill-posed problem. We shall show how to extract from the data a limited *ad hoc* amount of information which, on one hand, is not sufficient to recover the full set of the unknown parameters, but, on the other hand, is stable and useful from the applications point of view.

1. Introduction.

We are interested in determining, within a region Ω whose electrical conductivity is $\sigma \equiv 1$, an unknown inclusion $D \subset\subset \Omega$ of different conductivity $\sigma \equiv k$, $k > 0$, $k \neq 1$, by measuring from the exterior of Ω the current density and the voltage corresponding to one applied electrostatic distribution. This inverse problem is relevant to geophysical prospection, medical imaging, nondestructive testing. In mathematical terms, if u denotes the electrostatic potential within Ω, our aim is to recover D in the equation

$$\operatorname{div}((1+(k-1)\chi_D)\nabla u) = 0 \qquad \text{in } \Omega, \tag{1.1}$$

by the knowledge of boundary voltage and current measurements g, φ respectively:

$$u = g \qquad \text{on } \partial\Omega, \tag{1.D}$$

$$\frac{\partial u}{\partial \nu} = \varphi \qquad \text{on } \partial\Omega, \tag{1.N}$$

where ν denotes the unit exterior normal to $\partial\Omega$. Here χ_D denotes the characteristic function of D.

There is a clear evidence that this problem is severely ill-posed. In fact it appears that, in order to determine the unknown inclusion D, it is necessary to analitically continue the potential u from the Cauchy data (1.D), (1.N) up to the (unknown) interior boundary ∂D. As is well known, the process of solving such a Cauchy problem is highly ill-posed. Besides, despite the numerous efforts [B-F], [Al], [B-F-I], [F-I], [I-P], [C], [Po], [A-I-P], [A-I], [B-F-S], [S], the problem of uniqueness in the determination of D from the data $\{g, \varphi\}$ has never been completely solved. Therefore, it seems reasonable to focus on goals which are less ambitious than the precise, complete recovery of the inclusion D. More precisely, we pose the question whether it is possible to extract, in an efficient way, from the boundary data $\{g, \varphi\}$, a restricted number of practically useful parameters associated to D. The parameter we shall consider here is related to the size of D.

We shall give constructive estimates on the measure of the unknown inclusion D in terms of the boundary data g, φ, provided an a priori $C^{1,\alpha}$ bound on ∂D is given. In order to illustrate our main result (Theorem 2.2) let us assume here, for the sake of simplicity, that the conductivity in D is $k \equiv 2$. Consider the solution u_0 to the Dirichlet problem (1.1)-(1.D) when D is replaced by the empty set

$$\Delta u_0 = 0 \qquad \text{in } \Omega, \tag{1.1$_0$}$$

$$u_0 = g \qquad \text{on } \partial\Omega. \tag{1.D_0}$$

We prove that if we know a priori that D satisfies some regularity bound then its measure is comparable with the boundary integral $\delta\mathcal{W} = \int_{\partial\Omega}(\varphi - \varphi_0)g$, where $\varphi_0 = \partial u_0/\partial\nu$ denotes the Neumann data for u_0. Namely

$$C_1 \frac{\delta\mathcal{W}}{(\operatorname{osc}_{\partial\Omega} g)^2} \leq \text{meas}\,(D) \leq C_2 \frac{\delta\mathcal{W}}{(\operatorname{osc}_{\partial\Omega} g)^2}. \tag{1.2}$$

Notice that the above integral $\delta\mathcal{W}$ has a clear physical interpretation. Denoting by $\mathcal{W} = \int_\Omega (1+\chi_D)|\nabla u|^2$, $\mathcal{W}_0 = \int_\Omega |\nabla u_0|^2$ the powers required to maintain the boundary voltage g on $\partial\Omega$ when the inclusion D is respectively present and absent, we have by standard integration by parts $\delta\mathcal{W} = \int_{\partial\Omega}(\varphi - \varphi_0)g = \mathcal{W} - \mathcal{W}_0$.

* Work supported in part by Fondi MURST 40% and 60%.

Our approach to (1.2) is based on the following considerations. By elementary variational arguments (see for details Lemma 2.4) we are led to the estimates

$$\int_D |\nabla u|^2 \leq \delta \mathcal{W} \leq \int_D |\nabla u_0|^2. \tag{1.3}$$

Hence, by estimating $\sup_D |\nabla u_0|$ in terms of the boundary data, it is rather easy to obtain the inequality on the left hand side of (1.2). Conversely, in order to get an upper bound on the measure of D, one needs to have lower bounds on $|\nabla u|$ within D and in principle the gradient of u, which is harmonic in D, might vanish of any order in interior points of D. It is possible, however, to obtain an integral lower bound on $|\nabla u|$ in terms of the boundary data, by the application of stability estimates for the Cauchy problem for Laplace's equation, separately in D and $\Omega \setminus \bar{D}$. We shall obtain (see (2.30) in the proof of Theorem 2.2)

$$\int_D |\nabla u|^2 \geq K(\mathrm{osc}_{\partial\Omega} g)^2 \mathrm{meas}\ (D), \tag{1.4}$$

where $K > 0$ only depends on smoothness bounds on $\partial\Omega$ and ∂D and on the ratio $\|g\|_{C^{1,\alpha}(\partial\Omega)}/\mathrm{osc}\ _{\partial\Omega} g$. By (1.3) and (1.4), the right hand side in (1.2) easily follows. Our technique applies also when the conductivities in $\Omega \setminus \bar{D}$ and in D are nonuniform and nonisotropic, and in fact our Theorem 2.2 will be stated in this greater generality. On such conductivities we shall only require bounds on ellipticity, Lipschitz continuity and a bound on the jump between the two conductivities. That is, roughly speaking, we require that the conductivity within D is definitely higher (or smaller) than the conductivity outside. See (2.1)-(2.5) below for a precise statement.

In this connection, the pioneering work of Friedman ([F]) must be mentioned (see also the subsequent analysis of Bryan ([Bry]), where a related, although nonconstructive, approach was developed.

In the next section we shall state our main result, Theorem 2.2, and illustrate the main lines of a proof. For more details the reader is referred to [A-R].

2. The main theorem.

We shall fix the space dimension $n \geq 2$ throughout the paper. Therefore we shall omit the dependence of the various quantities on n. Let us introduce some notations and definitions.

We shall denote the Lebesgue measure of a set $E \subset \mathbb{R}^n$ by $|E|$.

Definition 2.1. Let Ω be a bounded domain in $\mathbb{R}^n$. Given α, $0 < \alpha < 1$, we shall say that $\partial\Omega$ belongs to the *class* $C^{1,\alpha}$, *with constants* r_0, E_0, if for any $x_0 \in \partial\Omega$, $\partial\Omega \cap B_{r_0}(x_0)$ is a connected surface the equation of which, in a cartesian coordinate system having origin at x_0 and x_n-axis in the direction of the outer normal ν to Ω at x_0, is of the form $x_n = \Phi(x_1, ..., x_{n-1})$, $\Phi(0) = 0$, $\nabla\Phi(0) = 0$, with Φ of class $C^{1,\alpha}$ in the $(n-1)$-dimensional ball $B_{r_0}(x_0) \cap \{x_n = 0\}$ and $\|\Phi\|_{C^{1,\alpha}} \leq E_0$. Given a domain $D \subset\subset \Omega$, we shall say that ∂D is of *class* $C^{1,\alpha}$ *with constants* r_0, E_0 *relative to* Ω if D satisfies the above condition and moreover $B_{r_0}(x_0) \subset \Omega$ for any $x_0 \in \partial D$. Notice that in this latter case $\mathrm{dist}(D, \partial\Omega) \geq r_0$.

Let us represent by two symmetric $n \times n$ matrix valued functions in Ω, $A = A(x)$, $B = B(x)$, the conductivity matrices in $\Omega \setminus \bar{D}$ and in D respectively. On such conductivities we shall assume

(i) (*uniform ellipticity*) there exists λ, $0 < \lambda \leq 1$ such that for every $x \in \Omega$, $\xi \in \mathbb{R}^n$:

$$\lambda|\xi|^2 \leq A(x)\xi \cdot \xi \leq \lambda^{-1}|\xi|^2, \tag{2.1}$$

$$\lambda|\xi|^2 \leq B(x)\xi \cdot \xi \leq \lambda^{-1}|\xi|^2. \tag{2.2}$$

(ii) (*Lipschitz continuity*) there exists $\Gamma > 0$ such that for every x, $y \in \Omega$

$$|A(x) - A(y)| \leq \Gamma|x - y|, \tag{2.3}$$

$$|B(x) - B(y)| \leq \Gamma|x - y|. \tag{2.4}$$

(iii) (*bounds on the jump*) there exist μ, η, $0 < \mu \leq \eta$ such that either

$$\mu|\xi|^2 \leq (B - A)(x)\xi \cdot \xi \leq \eta|\xi|^2 \tag{2.5+}$$

or

$$\mu|\xi|^2 \leq (A - B)(x)\xi \cdot \xi \leq \eta|\xi|^2 \tag{2.5−}$$

holds for every $x \in \Omega$ and $\xi \in \mathbb{R}^n$.

Given a boundary voltage $g \in C^{1,\alpha}(\partial\Omega)$, $g \not\equiv const.$, if the inclusion D is present then the electrostatic potential in Ω will be the weak solution $u \in W^{1,2}(\Omega)$ to the Dirichlet problem

$$\begin{cases} \mathrm{div}((A\chi_{\Omega\setminus\bar{D}} + B\chi_D)\nabla u) = 0 & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases} \tag{2.6}$$

whereas, if the inclusion is absent, then the electrostatic potential in Ω will be the weak solution $u_0 \in W^{1,2}(\Omega)$ to the Dirichlet problem

$$\begin{cases} \operatorname{div}(A\nabla u_0) = 0 & \text{in } \Omega, \\ \qquad u_0 = g & \text{on } \partial\Omega. \end{cases} \tag{2.7}$$

Theorem 2.2. *Let Ω be a bounded domain in $\mathbb{R}^n$ whose boundary is of class $C^{1,\alpha}$ with constants r_0, E_0 and let $D \subset\subset \Omega$ be a domain whose boundary ∂D is of class $C^{1,\alpha}$ with constants r_0, E_0 relative to Ω. Let A, B satisfy (2.1)-(2.4). Given $g \in C^{1,\alpha}(\partial\Omega)$, let u, $u_0 \in W^{1,2}(\Omega)$ be the weak solutions to (2.6), (2.7) respectively and let $\varphi = A\nabla u \cdot \nu|_{\partial\Omega}$, $\varphi_0 = A\nabla u_0 \cdot \nu|_{\partial\Omega}$ be the corresponding current measurements. If (2.5+) holds then we have*

$$\eta^{-1}C_1^+ \frac{\int_{\partial\Omega}(\varphi - \varphi_0)g}{(\mathrm{osc}_{\partial\Omega} g)^2} \le |D| \le \mu^{-1}C_2^+ \frac{\int_{\partial\Omega}(\varphi - \varphi_0)g}{(\mathrm{osc}_{\partial\Omega} g)^2}, \tag{2.8+}$$

if, conversely, (2.5-) holds then we have

$$\eta^{-1}C_1^- \frac{\int_{\partial\Omega}(\varphi_0 - \varphi)g}{(\mathrm{osc}_{\partial\Omega} g)^2} \le |D| \le \mu^{-1}C_2^- \frac{\int_{\partial\Omega}(\varphi_0 - \varphi)g}{(\mathrm{osc}_{\partial\Omega} g)^2}, \tag{2.8−}$$

where C_1^+, C_1^- depend on r_0, E_0, α, $|\Omega|$, λ, Γ only, and C_2^+, C_2^- only depend on the same quantities and in addition on $||g||_{C^{1,\alpha}(\partial\Omega)}/\mathrm{osc}_{\,\partial\Omega}g$.

Here and in the sequel we shall denote

$$u^e = u|_{\Omega\setminus\bar{D}}, \qquad u^i = u|_D.$$

Let us start by recalling the following regularity result due to DiBenedetto, Elliot and Friedman ([D-E-F]).

Theorem 2.3. *In the hypotheses of Theorem 2.2 we have that $u^i \in C^{1,\beta}(\bar{D})$, $u^e \in C^{1,\beta}(\bar{\Omega}\setminus D)$ and*

$$\max\left(||u^i||_{C^{1,\beta}(\bar{D})}, ||u^e||_{C^{1,\beta}(\bar{\Omega}\setminus D)}\right) \le K||g||_{C^{1,\alpha}(\partial\Omega)}, \tag{2.9}$$

where $\beta \in (0,1)$ and K depend on r_0, E_0, α, λ, Γ, $|\Omega|$ only. Moreover the following transmission conditions hold

$$u^e = u^i \qquad \text{on } \partial D, \tag{2.10}$$

$$A\nabla u^e \cdot \nu = B\nabla u^i \cdot \nu \qquad \text{on } \partial D. \tag{2.11}$$

The first step in the proof of Theorem 2.2 consists in estimating the power gap $\delta\mathcal{W} = \int_{\partial\Omega}(\varphi - \varphi_0)g$ in terms of the Dirichlet integrals of u and u_0 in D. Here the powers $\mathcal{W}$, $\mathcal{W}_0$ are given by:

$$\mathcal{W} = \int_\Omega (A\chi_{\Omega\setminus\bar{D}} + B\chi_D)\nabla u \cdot \nabla u = \int_{\partial\Omega} g\varphi, \qquad \mathcal{W}_0 = \int_\Omega A\nabla u_0 \cdot \nabla u_0 = \int_{\partial\Omega} g\varphi_0.$$

Lemma 2.4. *Let the hypotheses of Theorem 2.2 be satisfied. If (2.5+) holds, then we have*

$$\mu\int_D |\nabla u|^2 \le \int_{\partial\Omega}(\varphi - \varphi_0)g \le \eta\int_D |\nabla u_0|^2, \tag{2.12+}$$

if instead (2.5-) holds, then we have

$$\mu\int_D |\nabla u_0|^2 \le \int_{\partial\Omega}(\varphi_0 - \varphi)g \le \eta\int_D |\nabla u|^2, \tag{2.12−}$$

Proof. By (2.6), (2.7) we have:

$$\mathcal{W} = \int_{\partial\Omega} g\varphi = \min_{\substack{v\in W^{1,2}(\Omega)\\ v|_{\partial\Omega}=g}} \int_\Omega (A\chi_{\Omega\setminus\bar{D}} + B\chi_D)\nabla v \cdot \nabla v, \tag{2.13}$$

$$\mathcal{W}_0 = \int_{\partial\Omega} g\varphi_0 = \min_{\substack{v\in W^{1,2}(\Omega)\\ v|_{\partial\Omega}=g}} \int_\Omega A\nabla v \cdot \nabla v. \tag{2.14}$$

Consequently

$$\int_{\partial\Omega} g\varphi \leq \int_{\Omega} (A + (B - A)\chi_D)\nabla u_0 \cdot \nabla u_0 = \int_{\partial\Omega} g\varphi_0 + \int_D (B - A)\nabla u_0 \cdot \nabla u_0, \tag{2.15}$$

$$\int_{\partial\Omega} g\varphi_0 \leq \int_{\Omega} A\nabla u \cdot \nabla u = \int_{\partial\Omega} g\varphi - \int_D (B - A)\nabla u \cdot \nabla u, \tag{2.16}$$

and (2.12+), (2.12-) immediately follow from (2.5+), (2.5-) respectively. □

The next three Lemmas are essentially based on estimates for the Cauchy problem for elliptic equations. It is convenient to introduce the following notations.

Notations. Given a domain $G \subset \mathbb{R}^n$, for any $h > 0$, we shall denote

$$G_h = \{x \in G \mid \text{dist}(x, \partial G) > h\}, \qquad G^h = \{x \in \mathbb{R}^n \mid \text{dist}(x, G) < h\}.$$

Given $y \in \mathbb{R}^n$, $\xi \in \mathbb{R}^n$, $|\xi| = 1$, $\theta > 0$, $r > 0$, we shall denote by

$$C(y, \xi, \theta, r) = \{x \in \mathbb{R}^n \text{ s. t. } \frac{(x - y) \cdot \xi}{|x - y|} > \cos\theta, |x - y| < r\}, \tag{2.17}$$

the intersection of the ball $B_r(y)$ with the open cone with vertex y, axis in the direction ξ and width 2θ.

Lemma 2.5. *Let G, G' be bounded domains in $\mathbb{R}^n$ such that $G \subset G^h \subset G'$. Let v be a solution to the equation*

$$div(A\nabla v) = 0 \qquad in\ G', \tag{2.18}$$

where A satisfies assumptions (2.1) and (2.3), and let $||v||_{C^\sigma(G')} \leq E$, $0 < \sigma \leq 1$. Let $y \in \partial G'$ be such that $C(y, \xi, \theta, (k_\theta + 1)h) \subset G'$ and $\tilde{y} := y + k_\theta h\xi \in G$, where $k_\theta = 1/\sin\theta$. One can determine a strictly increasing and continuous function ω_1 on $[0, +\infty)$, with $\omega_1(0) = 0$, which depends on λ , Γ , h, $|G'|$, θ, σ only, such that for every $x \in G$

$$\frac{||v||_{L^\infty(B_{h/4}(x))}}{E} \geq \omega_1\left(\frac{|v(y)|}{E}\right). \tag{2.19}$$

Sketch of the Proof. The result is obtained by considering a chain of pairwise disjoint balls of radii $\rho \leq h$ joining x and $\tilde{y}$ and by repeated application of a three sphere inequality for solutions to (2.18). Such kind of inequality is well-known when the coefficient matrix A is smooth. However, when A is Lipschitz continuous it can be derived, through minor adaptations, from the estimates found by Garofalo and Lin in their proof of the unique continuation properties for equations like (2.18), [G-L]. □

Let us denote by F any connected component of $\Omega \setminus \bar{D}$ such that $\partial F \cap \partial\Omega \neq \emptyset$.

The following Lemma is based on a stability estimate for the Cauchy problem due to Trytten [T].

Lemma 2.6. *Let the hypotheses of Theorem 2.2 be satisfied. There exist $h_0 > 0$ depending on r_0, E_0, α only and $z_0 \in F$ such that $B_{h_0}(z_0) \subset F$ and, for any $c \in \mathbb{R}$,*

$$||u^e - c||_{L^\infty(B_{h_0}(z_0))} \leq C\left(||u^e - c||_{L^\infty(\partial D)} + ||\nabla u^e||_{L^\infty(\partial D)}\right)^{\delta'} ||u^e - c||^{1-\delta'}_{L^\infty(\Omega\setminus\bar{D})}, \tag{2.20}$$

where $\delta' \in (0, 1)$ and $C > 0$ depend on r_0, E_0, α, λ, Γ only.

Lemma 2.7. *Let us assume the hypotheses of Theorem 2.2. Let $h \leq r_0/(k_0+1)$, where $k_0 = \sqrt{1 + E_0^2 r_0^{2\alpha}}$ and let $c \in [\min_{\partial\Omega} g, \max_{\partial\Omega} g]$. One can determine a strictly increasing and continuous function ω_2 on $[0, +\infty)$, with $\omega_2(0) = 0$, which depends on λ , Γ , h, $|\Omega|$, r_0, E_0, α only, such that for every $x \in D_h$*

$$||u^i - c||_{L^\infty(B_{h/4}(x))} \geq osc\ _{\partial\Omega}g\ \omega_2\left(\frac{osc\ _{\partial\Omega}g}{||g||_{C^{1,\alpha}(\partial\Omega)}}\right). \tag{2.21}$$

Proof. By (2.9) and since $c \in [\min_{\partial\Omega} g, \max_{\partial\Omega} g]$,

$$\max(||u^i - c||_{C^{1,\beta}(\bar{D})}, ||u^e - c||_{C^{1,\beta}(\bar{\Omega}\setminus D)}) \leq E, \tag{2.22}$$

where

$$E = K\|g\|_{C^{1,\alpha}(\partial\Omega)}, \tag{2.23}$$

where $\beta \in (0,1)$ and K depend on r_0, E_0, α, λ, Γ, $|\Omega|$ only. Let $y_1 \in \partial D$ be such that $|(u^i - c)(y_1)| = \|u^i - c\|_{L^\infty(D)}$ and let $y_2 \in \partial(\Omega \setminus \bar{D})$ be such that $|(u^e - c)(y_2)| = \|u^e - c\|_{L^\infty(\Omega\setminus\bar{D})}$. Let F be the connected component of $\Omega \setminus \bar{D}$ such that $y_2 \in \partial F$. It is easy to see that $C(y_1, -\nu, \theta_0, (k_0+1)h) \subset D$ and $C(y_2, -\nu, \theta_0, (k_0+1)h) \subset \Omega \setminus \bar{D}$, for $h \leq r_0/(k_0+1)$, where $\theta_0 = \arcsin\left(1 + E_0^2 r_0^{2\alpha}\right)^{-1/2}$. Applying Lemma 2.5 to $u^i - c$ with $G' = D$, $G = D_h$, $y = y_1$, $\sigma = 1$ and to $u^e - c$ with $G' = F$, $G = F_h$, $y = y_2$, $\sigma = 1$ respectively, we have

$$\frac{\|u^i - c\|_{L^\infty(B_{h/4}(x))}}{E} \geq \omega_1\left(\frac{\|u^i - c\|_{L^\infty(D)}}{E}\right), \tag{2.24}$$

for every $x \in D_h$, where ω_1 depends on h, $|\Omega|$, r_0 , E_0, α, λ, Γ only;

$$\frac{\|u^e - c\|_{L^\infty(B_{h_0/4}(z_0))}}{E} \geq \omega_1\left(\frac{\|u^e - c\|_{L^\infty(\Omega\setminus\bar{D})}}{E}\right), \tag{2.25}$$

where $z_0 \in F$, h_0 have been introduced in Lemma 2.6. By the interpolation inequality (see [M], §33)

$$\|\nabla u^i\|_{L^\infty(D)} \leq C\|u^i - c\|^t_{L^\infty(D)}\|u^i\|^{1-t}_{C^{1,\beta}(\bar{D})}, \tag{2.26}$$

where $t = \beta/(\beta+1)$ and C is a positive constant which depends on r_0, E_0, α, $|\Omega|$ only, it follows that

$$\frac{\|u^i - c\|_{W^{1,\infty}(D)}}{E} \leq C\left(\frac{\|u^i - c\|_{L^\infty(D)}}{E}\right)^t. \tag{2.27}$$

Recalling the transmission conditions (2.10) and (2.11) we have

$$\|u^e - c\|_{L^\infty(\partial D)} + \|\nabla u^e\|_{L^\infty(\partial D)} \leq \sqrt{1 + 4\lambda^{-4}}\|u^i - c\|_{W^{1,\infty}(D)}. \tag{2.28}$$

Combining (2.24), (2.27), (2.28), (2.20), (2.25), the trivial estimate $\|u^e - c\|_{L^\infty(\Omega\setminus\bar{D})} \geq (\text{osc}_{\,\partial\Omega}\, g)/2$ and (2.23) and taking into account that ω_1 is strictly increasing, it is easy to determine a strictly increasing and continuous function ω_2, with $\omega_2(0) = 0$, such that (2.21) holds for every $x \in D_h$. □

In order to complete the proof of Theorem 2.2 we need the following Lemma of geometrical character, of which we shall omit the proof.

Lemma 2.8. *One can determine $h_1 > 0$, only depending on r_0, E_0, α, such that for every $h \leq h_1$:*

$$|D_h| \geq \frac{1}{2}|D|. \tag{2.29}$$

Proof of Theorem 2.2. We shall consider only the case when (2.5+) holds, the opposite case (2.5-) being analogous. As a consequence of (2.12+), we have: $\int_{\partial\Omega}(\varphi - \varphi_0)g \leq \eta \max_D |\nabla u_0|^2|D|$.

The lower bound appearing in (2.8+) then follows by using the classical gradient estimate (see [G-T])

$$\max_D |\nabla u_0| \leq C\text{osc}_{\,\partial\Omega}\, g,$$

where C depends on λ, Γ, $\text{dist}(D, \partial\Omega)$ only, and recalling that $\text{dist}(D, \partial\Omega) \geq r_0$.

Let $\epsilon = \min\left(r_0/(k_0+1), h_1/(\sqrt{n}+1)\right)$, where k_0, h_1 are the numbers introduced in Lemmas 2.7, 2.8 respectively. Let us cover D_{h_1} with internally nonoverlapping closed cubes Q_k of side ϵ, for $k = 1, ..., K$. By the choice of ϵ, the cubes Q_k are contained in D.

$$\int_D |\nabla u|^2 \geq \int_{\cup_{k=1}^K Q_k} |\nabla u|^2 \geq \frac{|D_{h_1}|}{\epsilon^n}\int_{Q_{\bar{k}}} |\nabla u|^2,$$

where $\bar{k}$ is chosen in such a way that $\int_{Q_{\bar{k}}} |\nabla u|^2 = \min_k \int_{Q_k} |\nabla u|^2$. Let $\bar{x}$ be the center of $Q_{\bar{k}}$. Now, from Poincaré inequality

$$\int_{B_{\epsilon/2}(\bar{x})} (u-\bar{c})^2 \leq C\epsilon^2 \int_{B_{\epsilon/2}(\bar{x})} |\nabla u|^2,$$

where $\bar{c} = (1/|B_{\epsilon/2}(\bar{x})|) \int_{B_{\epsilon/2}(\bar{x})} u$, the local estimate (see [G-T])

$$\int_{B_{\epsilon/2}(\bar{x})} (u-\bar{c})^2 \geq C\epsilon^n ||u-\bar{c}||^2_{L^\infty(B_{\epsilon/4}(\bar{x}))},$$

where C depends on λ only, and applying Lemma 2.7 with $c = \bar{c}$ and Lemma 2.8 we have

$$\int_D |\nabla u|^2 \geq C\epsilon^{-2}|D|(\mathrm{osc}_{\partial\Omega} g)^2, \tag{2.30}$$

where C depends on r_0, E_0, α, $|\Omega|$, λ, Γ, $||g||_{C^{1,\alpha}(\partial\Omega)}/\mathrm{osc}\ _{\partial\Omega} g$ only. □

References.

[Al] Alessandrini, G., Remark on a paper by Bellout and Friedman. Boll. Un. Mat. Ital. A, 23 (1989), 243-249.

[A-I] Alessandrini, G and Isakov, V., Analiticity and uniqueness for the inverse conductivity problem, preprint.

[A-I-P] Alessandrini, G., Isakov, V. and Powell, J., Local uniqueness in the inverse conductivity problem with one measurement. Trans. Amer. Math. Soc., 347 (1995), 3031-3041.

[A-R] Alessandrini, G. and Rosset, E., The inverse conductivity problem with one measurement: bounds on the size of the unknown object. Submitted.

[B-F-S] Barceló, B., Fabes, E. and Seo, J. K., The inverse conductivity problem with one measurement: uniqueness for convex polyhedra. Proc. Amer. Math. Soc., 122 (1994), 183-189.

[B-F] Bellout, H. and Friedman, A., Identification problems in potential theory. Arch. Rational Mech. Anal., 101 (1988), 143-160.

[B-F-I] Bellout, H., Friedman, A. and Isakov, V., Stability for an inverse problem in potential theory. Trans. Amer. Math. Soc., 332 (1992), 271-296.

[Bry] Bryan, K., Single measurement detection of a discontinuos conductivity. Comm. Partial Differential Equations, 15 (1990), 503-514.

[C] Cherednichenko, V. G., A problem in the conjugation of harmonic functions and its inverse. Differential Equations, 18 (1982), 503-509.

[D-E-F] DiBenedetto, E., Elliot, C. M. and Friedman, A., The free boundary of a flow in a porous body heated from its boundary, Appendix: A diffraction problem. Nonlinear Anal., 10 (1986), 879-900.

[F] Friedman, A., Detection of mines by electric measurements. SIAM J. Appl. Math., 47 (1987), 201-212.

[F-I] Friedman, A. and Isakov, V., On the uniqueness in the inverse conductivity problem with one measurement. Indiana Univ. Math. J., 38 (1989), 563-579.

[G-L] Garofalo, N. and Lin, F., Monotonicity properties of variational integrals, A_p weights and unique continuation. Indiana Univ. Math. J., 35 (1986), 245-68.

[G-T] Gilbarg, D. and Trudinger, N. S., Elliptic partial differential equations of second order. Springer, New York, 1983.

[I-P] Isakov, V. and Powell, J., On the inverse conductivity problem with one measurement. Inverse Problems, 6 (1990), 311-318.

[M] Miranda, C., Partial differential equations of elliptic type. Springer-Verlag, New York, 1970.

[Po] Powell, J., On a small perturbation in the two dimensional inverse conductivity problem. J. Math. Anal. Appl., 175 (1993), 292-304.

[S] Seo, J. K., A uniqueness result on inverse conductivity problem with two measurements. to appear in J. Fourier Anal. Appl.

[T] Trytten, G. N., Pointwise bounds for solutions of the Cauchy problem for elliptic equations. Arch. Rational Mech. Anal., 13 (1963), 222-244.

MODEL VALIDATION IN ITERATIVE IDENTIFICATION AND CONTROLLER DESIGN

Mina Žele, Đani Juričić
Jožef Stefan Institute
Jamova 39, 1111 Ljubljana, Slovenia
e-mail: mina.zele@ijs.si

Abstract. The paper deals with validation of model purposivity in iterative identification and controller design. It is known that in the case of too high closed loop requirements, the model resulting from the iterative procedure might conflict with the prior knowledge about the process. However, in some cases violated plausibility of the identified models does not necessarily imply its violated purposivity. Therefore, it is the matter of practical relevance to have a confident indication whether the given model will result in stable closed loop design or not. If not, the iterative identification and controller design should be stopped, i.e. more appropriate models structures should be chosen. In the paper a stochastic robustness measure is proposed which relies on the estimated model error obtainable by stochastic embedding technique. In the simulated example we consider three models as a result of iterative procedure that have different qualities regarding plausibility and purposivity. It is shown that the stochastic robustness measure provides a reliable estimate of the designed loop stability.

1. Introduction

The final step in any identification process is model validation, which consists of checking of the model plausibility and purposivity. The first step is merely to examine whether the model conforms to the a priori knowledge about the process. In the second step the model ability to serve the intended purpose is evaluated [2].

The purpose of this paper is to outline the role of model quality in iterative identification and controller design and to point out its significance, particularly in cases when plausibility and successful closed-loop design provide conflicting evidence about the model validity. For example, a model which violates the a priori assumptions may well accomplish the task, i.e. may result in stable closed loop design. From the condition for the robust stability we see that in order to guarantee the closed loop stability the model error should be made small at the frequency range where the sensitivity function is high. Since good model fit is not required at low frequencies, identification can, in some cases, results in a model which is not plausible.

There are several approaches to the estimation of model quality [1,3]. In this paper it is expressed in terms of soft (probabilistic) bounds on the model error. To estimate the model error in the frequency domain, which is convenient for the robust stability analysis, we use the stochastic embedding approach [4].

The paper is organised as follows. Section 2 deals with the role of model quality in iterative identification and controller design. In section 3 the stochastic embedding approach to the estimation of model quality is considered which is used to derive the stochastic robustness measure introduced in section 4. In section 5 the results of the iterative procedure for the second order simulated process are given. Finally, some concluding remarks are drawn in section 6.

2. Iterative identification and controller design

The aim of the iterative scheme is to improve the closed loop performance by repeating the controller design on the basis of the model obtained from the closed loop data. Improving the performance means that we wish to get the achieved closed loop as close as possible to the designed closed loop. For that purpose the model error should be kept small at the frequencies where process and model closed loop sensitivities are large [3]. This is achieved by identification in the closed loop and appropriate closed-loop data filtering. The role of filtering is to realise the frequency weighting of the model error in the identification criterion.

However, it is not obvious whether the procedure generally converges or not. Namely, the iterative procedure can be viewed as a discrete process described by means of a non-linear difference equation relating the model parameters $\hat{\theta}$ from the two successive iterations

$$\hat{\theta}_{i+1} = f(C_{i+1}(\hat{\theta}_i)) \tag{1}$$

where C represents the controller designed on the basis of the previous process model. Its behaviour depends greatly on the closed loop requirements [1]. In the case of using a low order model, reasonable estimates are achieved when the desired closed-loop bandwidth is appropriately low compared to the process one. This statement is based on the fact that a low order model can approximate a process reasonably well only at low frequencies. Mismatch at higher frequencies, which occurs when insisting on fast closed-loop response, results in oscillatory behaviour, limit cycles or even divergence of the recursion (1).

The most strange effect, however, is convergence towards the parameters, which do not make sense. This comes from the fact that low-order approximation of the higher order dynamic system at high frequencies leads to high bias in the estimated parameters. Since, the model error influences the stability of the achieved closed loop system, good quantification of the model error is needed to make an accurate enough estimate that the closed-loop system will remain stable. The stochastic embedding [3] approach to the estimation of model error bounds has been employed as a basis for calculating the probability for stably designed loop.

3. Stochastic embedding approach to the estimation of model uncertainty bounds

The idea of the stochastic embedding approach [3] is based on the assumption that the unmodelled dynamics could be though of as realisation of a random process with parametrised second order statistics. Only the structure of the stochastic models for the undermodelling error and the noise is required, while the accompanying free parameters are determined from the measured data.

The true system is supposed to be

$$y(t) = G_T(q^{-1},\theta)u(t) + n(t) \tag{2}$$

where q is shift operator, u stands for the input, y is system output, n(t) is additive noise and G_T is "true" transfer function of the process.

The true transfer function is considered to be composed of a nominal model $G(e^{-j\omega},\theta_0)$ and a random part $G_\Delta(e^{j\omega})$

$$G_T(e^{j\omega}) = G(e^{j\omega},\theta_0) + G_\Delta(e^{j\omega}) \tag{3}$$

where

$$E\left[G_T(e^{j\omega})\right] = G(e^{j\omega},\theta_0) \tag{4}$$

We assume that the model of the process is linear in the parameters θ

$$G(e^{j\omega},\theta) = \Lambda(e^{j\omega})\theta \tag{5}$$

where

$$\Lambda = [\Lambda_1(e^{j\omega}),\ldots,\Lambda_p(e^{j\omega})] \tag{6}$$

Here $\left\{\Lambda_i(e^{j\omega})\right\}$ is a set of rational transfer functions and p is the number of parameters. The model applies not only for the fixed denominator model, but also for all types of transfer function models that can be approximated by the first order Taylor expansion around the estimated parameter.

Note that the term $G_\Delta(e^{j\omega})$ can be represented by means of the impulse response form, i.e.

$$G_\Delta(q^{-1}) = \sum_{k=1}^{\infty} \eta_k q^{-k} \tag{7}$$

To fully specify the error term G_Δ it is necessary to specify the set of coefficients $\{\eta_k\}$ which is not a feasible task. To make the problem tractable, it is helpful to put it into the stochastic framework and to start with some a priori assumption about the *distribution* of the error terms.

The term G_Δ is assumed to be a realisation of the exponentially vanishing stochastic process. In other words, the coefficients η_k are supposed to be normally distributed with zero mean and variance

$$E\left[\eta_k^{\ 2}\right] = \alpha\lambda^k \qquad 0 < \lambda < 1 \tag{8}$$

Using the probabilistic model formulated above we see that only two parameters, i.e. α, λ, are required to fully specify $E\{G_\Delta\}$. They are estimated from the input/output data by maximising the corresponding likelihood function [4].

More evident representation of the model error is accomplished by expressing the real and imaginary component of the total model error in the Nyquist diagram. For that reason we define the following matrix

$$\tilde{g}(e^{j\omega}) = \begin{bmatrix} \mathrm{Re}\left\{G_T(e^{j\omega}) - G(e^{j\omega}, \hat{\theta}_N)\right\} \\ \mathrm{Im}\left\{G_T(e^{j\omega}) - G(e^{j\omega}, \hat{\theta}_N)\right\} \end{bmatrix} \tag{9}$$

with

$$E\left\{\tilde{g}(e^{j\omega})\right\} = 0 \tag{10}$$

and

$$E\left[\tilde{g}(e^{j\omega})\tilde{g}(e^{j\omega})^T\right] = \Sigma_{\tilde{g}}^{-1} \tag{11}$$

where $\hat{\theta}_N$ are the estimated parameters. The covariance matrix $\Sigma_{\tilde{g}}^{-1}$ consists of the contribution of the undermodelling error and the error induced by the noise. The term $\tilde{g}(e^{j\omega})$ is described by the Gaussian distribution while the quadratic expression of $\tilde{g}(e^{j\omega})$ has χ^2 distribution with 2 degrees of freedom. The confidence ellipses in the Nyquist diagram are thus defined as

$$\tilde{g}(e^{j\omega})^T \Sigma_{\tilde{g}}\, \tilde{g}(e^{j\omega}) = c(P) \tag{12}$$

where c is a constant given in the χ^2 distribution table for a desired degree of confidence (P).

4. A stochastic robustness measure

The controller design provides a controller which stabilises the *designed* loop, i.e. loop around the process model. However, the nominal model itself does not provide for full information about robustness of the *actual* closed loop. For that purpose we used results from section 3. Having the estimated bias and variance, it is possible to set quite realistic bounds of the model uncertainty and the robustness of the control system.

Since the model uncertainty is expressed in probabilistic terms, we propose a robustness measure for the designed closed loop expressed in probabilistic terms too. To investigate robustness of the actual closed loop system, we calculate the error bound $H_\Delta = CG_\Delta$ of the return ratio H=CG. The confidence ellipsoid is associated to the return ratio at any frequency ω as follows:

$$\tilde{h}(e^{j\omega})^T \Sigma_{\tilde{h}}\, \tilde{h}(e^{j\omega}) = c \tag{13}$$

where

$$\tilde{h}(e^{j\omega}) = \begin{bmatrix} \mathrm{Re}\left\{H_\Delta(e^{j\omega})\right\} \\ \mathrm{Im}\left\{H_\Delta(e^{j\omega})\right\} \end{bmatrix} \tag{14}$$

$$\Sigma_{\tilde{h}}^{-1} = E\{\tilde{h}(e^{j\omega})\tilde{h}^T(e^{j\omega})\} \tag{15}$$

If the uncertainty region is narrow and the nominal return ratio Nyquist curve is properly far from the point (-1,0), robustness of the designed closed loop is high. However, it is not enough to have only the nominal return ratio robustly shaped but it is also important that the *envelopes*, which define the uncertainty region, do not encircle the point (-1,0). Since the calculation of the probability that (-1,0) is encountered in the region defined by envelopes does not make sense, a different approach is used. In Figure 1 the situation with two different uncertainty regions is presented. The first one is associated to the probability measure P_1, addressed to the confidence ellipsoids, whilst the second one is associated to the probability measure P_{min} $(P_1<P_{min}<1)$. A way to express robustness of the designed loop is to find such P_{min} that

$$P_{min} = \arg\min_{P}\{c(P)|(-1,0) \in \tilde{h}(e^{j\omega})^T \Sigma_{\tilde{h}} \tilde{h}(e^{j\omega}) = c(P), \omega \in (0,\infty)\} \tag{16}$$

Then, with the *probability* P_{min} *we can guarantee* that the point (-1,0) is not encircled by the Nyquist curve. That means that higher P_{min} indicates greater robustness.

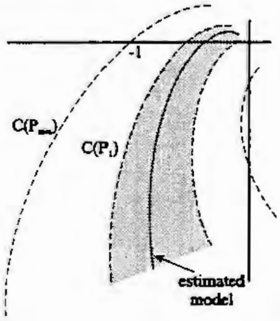


Figure 1: Illustration of the proposed stochastic robustness measure

5. Simulation study: stable low-pass second order system and first order model

The simulated process is a second order system with the transfer function

$$G_T(q^{-1}) = \frac{0.016429q^{-1} + 0.013451}{1 - 1.4891q^{-2} + 0.5488q^{-1}} q^{-1} \tag{17}$$

The system is identified by using ARX model with the transfer function parametrised as follows

$$G_0(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})} = \frac{bq^{-1}}{1 + aq^{-1}} \tag{18}$$

The PI controller

$$C(q^{-1}) = \frac{Q(q^{-1})}{P(q^{-1})} = \frac{q_0 + q_1 q^{-1}}{1 - q^{-1}} \tag{19}$$

is to be designed by the iterative procedure. Closed-loop requirements are formulated by means of the closed loop characteristic equation

$$A_C(q^{-1}) = (1 + \tau q^{-1})^2 \tag{20}$$

The controller parameters can be easily computed by taking into account that A_C=BQ+AP. The data filter used in identification has the following form

$$F_i(q^{-1}) = \frac{1-q^{-1}}{(1+\tau q^{-1})^2} \qquad (21)$$

To calculate the stochastic robustness measure, firstly the parameters of the bias and the noise model $n = N(0,\sigma_n^2)$ had to be estimated from the data. The confidence ellipsoids for c=2 were calculated for the models indicated in Table 1 by using (12). The plots are shown in Figure 2.

The iterative procedure has been performed for different values of the desired closed loop pole. The results are shown in Table 1. For the closed loop pole at τ=-0.96079 we obtain a stable model and the actual closed loop is also stable. At τ=-0.88692 the iterative procedure converges to the unstable model, while the designed controller results in stable actual closed loop system. For the closed loop pole at 0.86935 we also obtain the unstable model, but in this case the actual closed loop system is unstable, definitely showing that the model is not valid for the desired purpose.

From Figure 2 it can be seen that the uncertainty ellipsoids are considerably larger for both unstable models due to larger mismatch between the models and the process (Figure 2). In Figure 3 the corresponding confidence regions for the return ratio are depicted. As can be seen from Figure 3 the envelope around the model frequency response does not encircle the point (-1,0) for the stable model 1 (Figure 3a) and for the unstable model 2 (Figure 3 b). In order to achieve the envelope going through the critical point (-1,0) the constant c has to be increased. The enlarged envelope corresponds to a higher probability that the actual closed loop system is stable.

In Table 1 the robustness measures introduced in section 4 are given at different closed loop requirements. Each example represents a different situation with respect to plausibility and purposivity. The first model seems to be a reasonable description of the process and, at the same time, it is well suited for the controller design. The second model is unstable, but the stochastic robustness measure indicates a high probability of the actual closed loop stability. The last model is not plausible and is also not valid for designing a stable control loop.

Model	τ	a	b	c_{min}	P_{min}
1	-0.96079	-0.8727	0.0531	88.54	0.99
2	-0.88692	-1.0239	0.0419	84.64	0.99
3	-0.86935	-1.1375	0.0378	0.38	0.17

Table 1: The model parameters obtained in the iterative procedure and associated stochastic robustness measure P_{min}

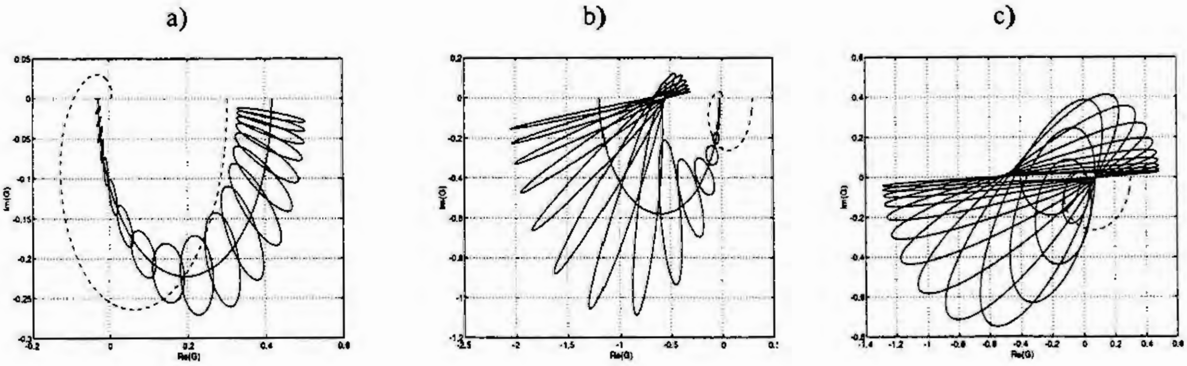


Figure 2: Nyquist diagram of the process and the model with the error bounds : a) τ= -0.96079; b) τ= -0.88692; c) τ= -0.86935 (solid line - model; dashed line - process)

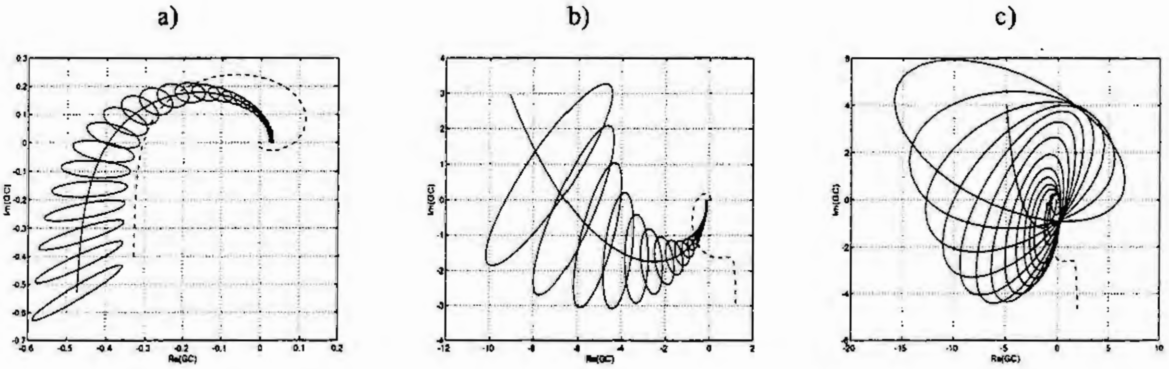


Figure 3: Nyquist diagram of CG_T (dashed line) and $C\hat{G}$ (solid line) with the error bound a) τ= -0.96079; b) τ= -0.88692; c) for τ=-0.86935

6. Conclusions

In this paper we focused on the importance of model validation in iterative identification and controller design. The model quality can be viewed as a carrier of key information for the purpose of model validation during the iterative procedure. The statistically described model error is usually directly correlated to the model plausibility, while by examining solely the estimated model error we can not come to the conclusions about its purposivity. Namely, the most suitable model for the controller design may produce high errors at low frequencies and may be in contradiction with the a priori knowledge about the process. We gave three simulation examples showing that purposivity and plausibility are not related.

For the aim of determining the model purposivity in the iterative identification and controller design, we introduced a stochastic robustness measure. It is shown how it can be derived from the model error bound obtained by stochastic embedding method. The stochastic robustness measure represents the reliable measure of probability of the actual closed-loop stability. The information about the chance of actual loop to get unstable is considered in each iteration step. If it exceeds a certain value, it indicates that the bias in estimated parameters is too high and a higher order model should be identified for a given controller design.

7. References

1. Åström, K. J. and Nilsson, J., Analysis of a scheme for iterated identification and control. Proc. SYSID'94, Copenhagen, Denmark, pp. 171-176,1994.

2. Sage, A.P., Validation. In: Concise Encyclopaedia of Modelling and Simulation, (Eds.: Atherton, D.P. and Borne, P.) Pergamon, Oxford, 1992, pp 477-488.

3. Gevers, M., Essays on Control: Perspectives in the Theory and its Applications. Birkhauser, Boston,1993.

4. Goodwin,G., Gevers, M. and Mayne, D. Q., Bias and variance distribution in transfer function estimation. Preprints Identification and system parameters estimation, 9th IFAC/IFORS Symposium, Budapest, Hungary, vol. 2, pp.952-957,1991.

4. Ninness, B. and Goodwin, G., Estimation of model quality. Proc. SYSID'94, Copenhagen, Denmark, pp.25-44, 1994.

6. Schrama, R., Accurate identification for control: The necessity of an iterative scheme. IEEE Transactions on Automatic Control, vol. 37, No 7, pp. 991-994,1992.

THE ROLE OF MODEL STRUCTURE VALIDATION IN PHARMACOKINETIC STUDIES

R.Karba[1], A.Mrhar[2], A.Belič[1], I.Grabnar[2], S.Primožič[2], B.Zupančič[1]
[1]Faculty of Electrical Engineering
Tržaška 25, 1000 Ljubljana
[2]Faculty of Pharmacy
Aškerčeva 7, 1000 Ljubljana
University of Ljubljana, Slovenia
E-mail: rihard.karba@fe.uni-lj.si.

Abstract. It is well known that mathematical models validation represents very delicate stage in the cyclic procedure of model development. Among very different and loosely defined categories, which are used in model validation, the descriptive realism deals with models which base on the correct assumptions about the mechanisms of the modelled process. The mentioned category is very usable in pharmacokinetic studies which often deals with models developed on the basis of some average data while the question how good they can handle the individual data becomes more and more actual due to the need of model based individual therapies design. The work deals with the mentioned problems by the aid model development and validation for nitrendipine being a modern antihypertensive drug. It came out that the discussed model contains all important mechanisms of nitrendipine kinetics and therefore through the certain changes of model parameters also individual profiles can be satisfactorily fitted, what indicates the structural validity of the model. The latter is therefore usable also for the individual therapy design.

Introduction

It is well known that mathematical models validation represents very delicate stage in the cyclic procedure of the model development [3, 5]. On one hand invalidated model has questionable value while on the other hand validation process can be tedious and unreasonably long. The corresponding compromise must be therefore found to obtain the usable model which helps elucidating the real problems. Among very different and loosely defined categories, which are used in model validation, the descriptive realism deals with models which base on the correct assumptions. The modification of the definition adds the requirement that the assumptions are connected with mechanisms of the process which are included in the model [4]. Such a modification is usable also in the pharmacokinetic studies.Pharmacokinetical models are namely frequently developed on the basis of some average literature data. The question is, however, if such model structure covers also the individual data profiles. As dosage forms and dosage regimen design go more and more in the direction of individualization of drug usage, the question of descriptive realism is crucial for the use of the model in concrete cases of individual therapies design. In the work the mentioned questions are discussed for the case of concrete drug.

Nitrendipine [1, 6] produces antihypertensive effect through blocade of the calcium conducting channels.After peroral application it is rapidly and completely absorbed in portal blood, while in the liver it is subject to extensive presystemic metabolism the consequence of which is approximately 25% absorption into the systemic blood circulation. The absolute biovailability does not depend on the size of the administered dose and on its release rate what is manifested in linearity of kinetics. It has been proved that no saturation of the liver enzymes occurs with the doses of 5 to 40 mg. The biological half life of nitrendipine is about 12 hours. The influence of the analytical method is also evident in the results of the pharmacokinetic analysis. The liquid chromatography (HPLC) permitted the identification of two phases, while gas chromatography (GC) drew the attention to the existence of the third phase in the plasma concentration profile. This finding has a key meaning in understanding the relation between the concentration and the effect. Namely, in literature no data could be found that would combine the plasma concentration of nitrendipine with the hypertensive effect. These investigations resulted in the presumption that there exists a correlation between the antihypertensive action and the nitrendipine concentration in calcium channels which was confirmed with the improved sensitivity of the analytical method. The existence of the "deep" peripheral compartment where the drug accumulates, maintains its pharmacodynamic effect and returns very slowly back in the central compartment, may be the reason for the observed low plasma concentrations in the terminal phase. The data base for this study was taken from the reference study [6] in which the authors investigated the pharmacokinetics of nitrendipine following the application in different doses, in an intravenous injection and perorally in a gelatinous capsule containing solution as well as in tablet with instant release. Nitrendipine plasma concentrations were determined with HPLC method. Only the plasma concentration profile obtained by following an intravenous injection of 2 mg of nitrendipine was pharmacokinetically analysed with the method of stripping. Their statement that a two-compartment model (with two phases in the concentration profile) represents a satisfactory approximation of nitrendipine pharmacokinetics served as a starting point for a gradual development of a model [7] in accordance with the purposes of modelling which was the development of sustained release oral dose.

As a tool for developing a model, compartmental pharmacokinetic modelling and digital simulation package Simulink was chosen. The structure validation of the final nine compartment model [2] with respect to individual data profiles was performed by the aid of the extensive "in vivo" study. So through the comparison of model responses and measured values for each volunteer the question about the descripitve realism of the proposed model structure was investigated.

Modelling Nitrendipine pharmacokinetics

As mentioned above the final stage of nitrendipine pharmacokinetics modelling procedure is given in a form of nine compartment model shown in Figure 1.

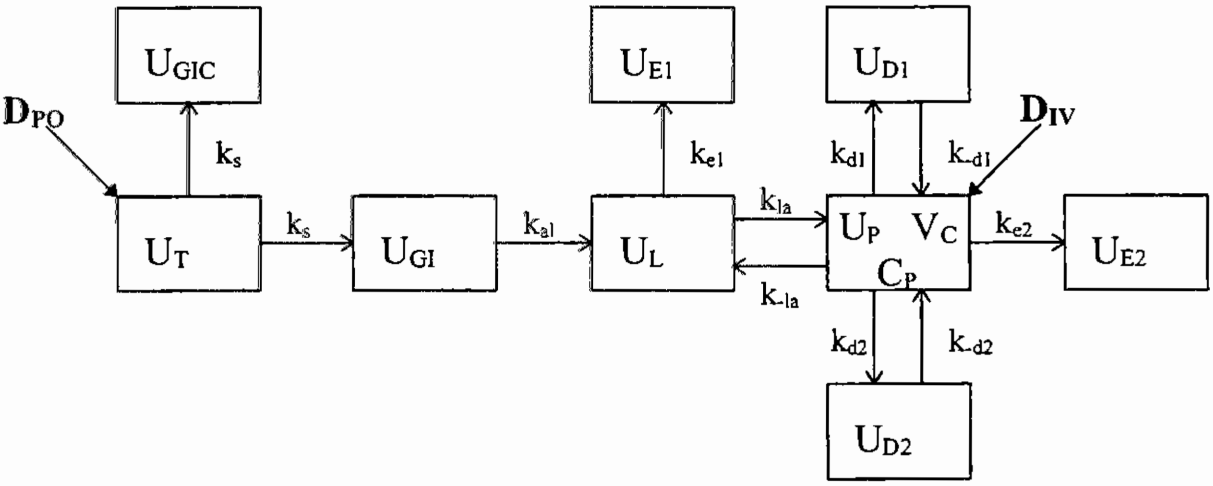


Figure 1: Pharmacokinetic model of nitrendipine

In Figure 1 the symbols have the following meaning:
U_T-levels in the tablet, U_{GI}-levels in the gastrointestinal tract, U_{GIC}- cumulative levels in the gastrointestinal tract, U_L-levels in the liver, U_P- levels in the plasma (the central compartment), C_P- plasma concentrations, V_C-volume of distribution of the central compartment, U_{D1} levels in the "shallow" peripheral compartment, U_{D2}-levels in the "deep" peripheral compartment, U_{E1}-levels of metabolites after presystemic metabolism, UE2-levels of metabolites after systemic metabolism, D_{PO}-peroral dose, D_{IV}- intravenous dose, k_s-dissolution rate constant, k_{al}-absorption rate constant into the portal blood circulation, k_{la} and k_{-la}-absorption rate constants into the central blood circulation, k_{d1} and k_{-d1} distribution rate constants into the "shallow" compartment, k_{d2} and k_{-d2}-distribution rate constant into the "deep" compartment, k_{e1} and k_{e2}-metabolizing rate constants.

The modelling results are in Figure 2.

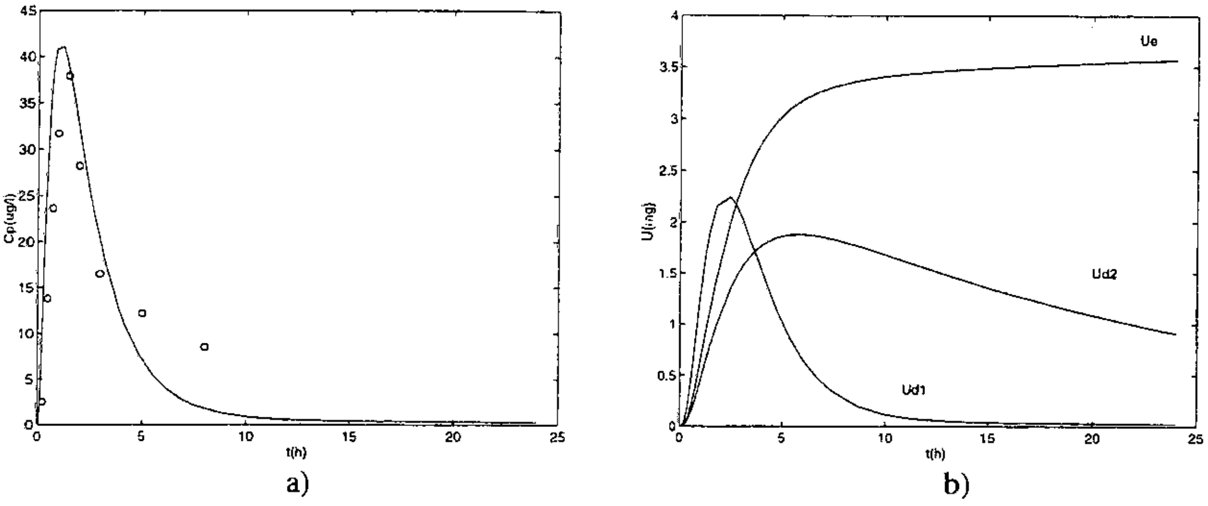


Figure 2: Nitrendipine levels following the application of a 30-mg dosage in a peroral tablets with instant release,
a - concentrations in the plasma, (dots - average in vivo measurements, curve - model response),

b - quantities in the peripheral compartments and metabolite compartment

In Figure 2 relatively good fitting of the model response to the measured average concentrations in plasma can be seen as well as the significance of the deep peripheral compartment supporting the presumption about its influence on the drug antihypertensive effects.

Model structure validation

The mentioned trends against model based individual therapies design indicate that the models being developed from some average measured data must be correspondingly validated in the sense that their structure must enable also the model response fitting to individual data profiles.

It is known from the literature [4] that certain model characteristics exist which bear on the question how good the models are. They are as follows:

- accuracy,
- descriptive realism,
- precision,
- robustness,
- generality,
- fruitfulness.

By definition a model is said to be descriptively realistic if it is based on the assumption which are correct. Additionally it must be deduced from a correct (or at least believable) description of the mechanisms involved in the modelled object. Therefore descriptive realism in a category usable also in pharmacokinetic studies like the discussed one.

To validate developed nitrendipine model in the mentioned sense the result of a large in vivo study were used. The latter was undertaken to investigate bioequivalence of two nitrendipine products (one Slovene and one German) in the form of tablets. The study was single dose (20 mg), blind, randomized, four way cross over, for fourty normal, healthy male volunteers. As everyone obtained each product two times the inter and intraindividual differences can be clearly visible from the measured plasma concentration profiles obtained by specific GC/MS analytical method. The data base of 160 profiles gave enough possibilities also for the developed nitrendipine model validation in the sense of descriptive realism. The measured plasma concentration profiles for both nitrendipine products together with minimal and maximal values are shown in Figure 3.

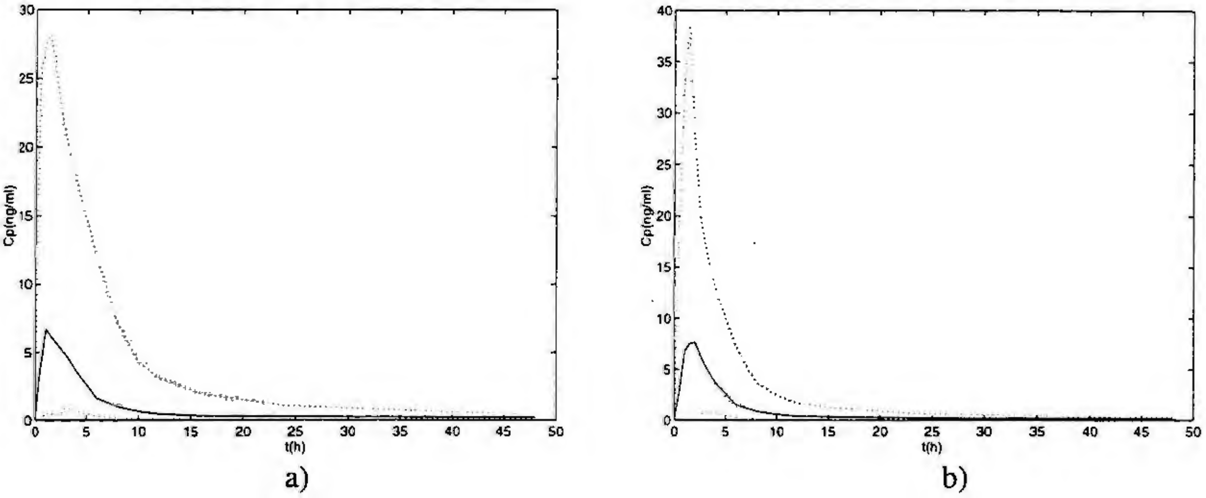


Figure 3: Measured nitrendipine plasma concentrations for both products (solid line-average values, dotted lines-minimal and maximal values)

a - first product

b - second product

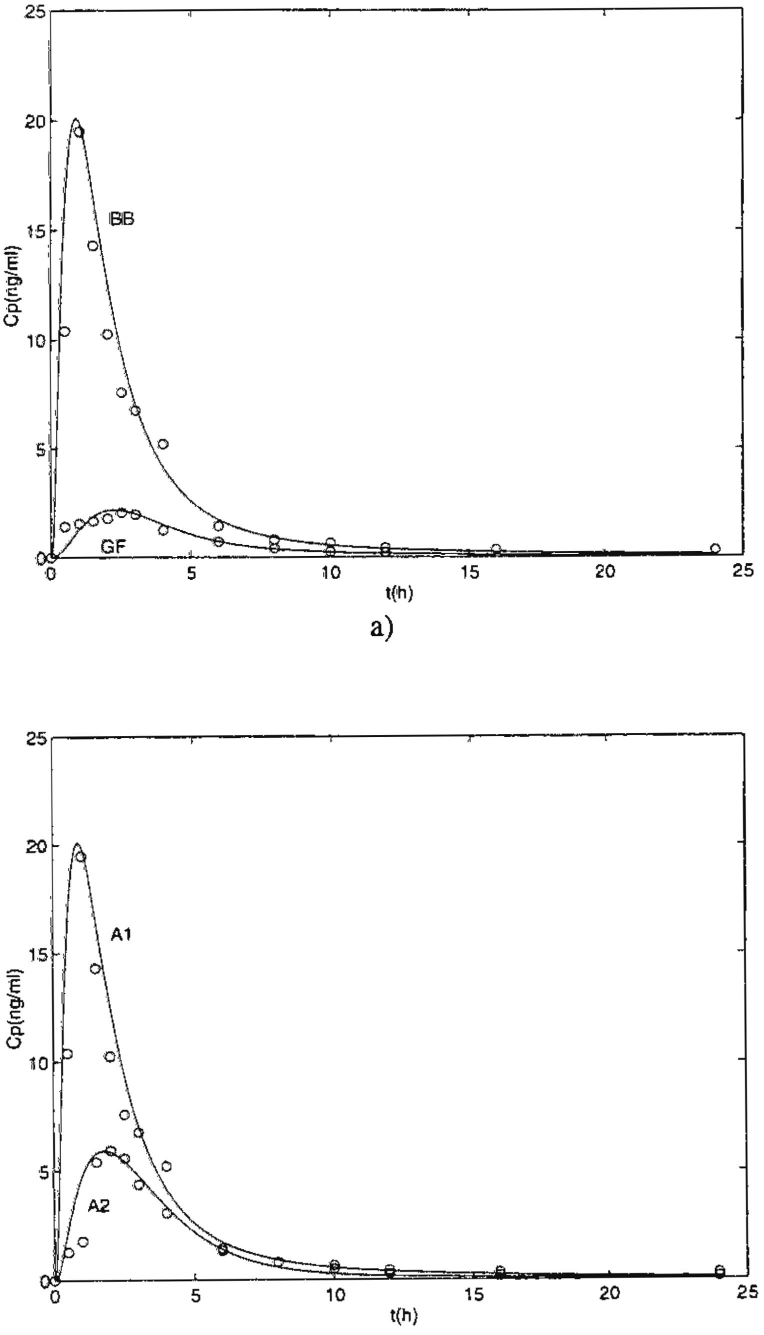
From Figure 3. large variability of profiles can be seen. They are the consequence of inter and intraindividual differences. The latter are shown in Figure 4 where the measured values for two products in one volunteer and the case of one product for two volunteers are shown together with the corresponding model responses.

Due to the lack of the space only two examples of the large data base can be given. Figure 4. shows how big differences can occur but also that the developed model responses can be satisfactorily fitted also to individual data. Even when they are so different than the average data of the pharmacokinetic study on the basis of which the model was developed. So it can be stated that the model is descriptively realistic.

Conclusions

We can conclude as follows:

- the structural validation in the sense of descriptive realism is usable in pharmacokinetic studies,
- the developed model is descriptively realistic and so it contains all important mechanisms of nitrendipine kinetics,
- the model of nitrendipine is stiff and therefore requires corresponding integration method in digital simulation (in our case MATLAB with SIMULINK),
- the developed model is usable for the individual therapy design.



a)

b)

Figure 4: Individual nitrendipine plasma concentration profiles (dots - measured data, line - model response)

a - interindividual differences - one product (A1) and two volunteers (BB and GF)

b - intraindividual differences - one volunteer (BB) and two products (A1 and A2)

References

[1] Goa, K.L. and Sarkin, E.M., Nitrendipine, a review of its pharmacodynamic and pharmacokinetic properties and therapeutic efficacy in the treatment of hypertension. Drugs, 33 (1987), 123-155.

[2] Grabnar, I., Mrhar, A., Belič A., Karba R., The role of pharmacokinetic modelling and computer simulation in drug formulation design. In: Proc. 9th International Conference on Mechanics in Medicine and Biology, Ljubljana, 1996, 291- 294.

[3] Matko, D., Karba, R., Zupančič B., Simulation and Modelling of Continuous Systems: A Case Study Approach, Prentice Hall, London, 1992.

[4] Meyer, W.J., Concepts of Mathematical Modelling, McGraw Hill, Singapore, 1985.

[5] Murray-Smith, D.J., Continuous System Simulation, Chapman&Hall, London, 1995.

[6] Remsch, K.D. and Sommer, J., Pharmacokinetics and metabolism of nitrendipine. In: Scriabne et. al. (Eds.), Urban&Schwarzenberg, 1984, 409-420.

[7] Remunan,C., Mrhar, A., Primožič, S., Karba, R., Vila-Jato, J.L. Sustained release infedipine formulations: moment, modelling and simulation as pharmacokinetic analysis approach. Drug Dev. Ind. Pharm, 18 (1992), 187-202.

APPLICATIONS OF THE DISTORTION METHOD FOR MODEL VALIDATION

G.J. Gray, Lew Koi Voon and D.J. Murray-Smith
Centre for Systems and Control and
Department of Electronics & Electrical Engineering
University of Glasgow

Abstract. The model distortion approach to the external validation of linear and nonlinear dynamic models uses the time histories of model parameter variations needed to ensure that the model output exactly fits experimental data from the real system as a basis for assessing the acceptability or otherwise of a given representation. It must then be established whether the variance of parameter distortions is less than the uncertainty of parameter values of the associated model. If the distortions are all within acceptable limits the model may be regarded as a suitable representation of the system. This paper describes applications of the distortion approach using both simulated and experimental response data. Problems associated with measurement noise are highlighted in the results.

Introduction

In 1986 Butterfield and Thomas [1] published an account of an external model validation technique which was based upon estimation of the distortion needed to achieve an exact fit of model variables to equivalent experimental response data through variation of model parameters as a function of time. This approach to the external validation of dynamic models is based upon the premise that "any model can be made to follow any observed transient by introducing enough distortion: the less the distortion, the better the model". Hence the parameter distortion time history itself provides a quantitative criterion of model validity. In the investigation of a model developed from physical principles, parameter distortions of variance smaller than the uncertainties associated with the relevant parameters of the model suggest that the chosen representation cannot be improved upon on the basis of the available information. On the other hand any parameter distortion variances larger than the inherent uncertainties of the given model suggest that there is some problem with the chosen model structure and parameter values.

In general the dynamic model under consideration consists of a set of nonlinear ordinary differential equations. Unmeasured state variables are considered as parameters and time histories of these variables are estimated along with the time histories of parameters. In the general case there are more parameters than equations. The problem is then over-determined and the solution involves a constrained optimisation based on an appropriate cost function and the requirement that the model output at all times equals the measured response of the real system.

There are only a few published accounts of applications of the approach. Apart from illustrations provided by Butterfield and Thomas [1,2] and their co-workers [3], the only investigations involving this approach appear to be those by Cameron [4] and by Gray [5]. Cameron has considered the application of the model distortion method to linear state space models and has developed methods which allow the derivation of explicit relationships for the parameter distortion. Gray has investigated the use of model distortion methods for linear and nonlinear models of systems of relatively low order and also for a linear helicopter model. The main conclusions of the investigations by Cameron and Gray were that measurement noise can create major problems in the application of the method to system response data. Cameron has suggested that the main area of application may be in the "assessment" of a linearised model against the nonlinear model from which it is derived.

The aim of the work described iu this paper is to re-assess the potential of the distortion approach for external validation through the application of the technique to a real system which is described by a model which has well known and well understood limitations.

Theoretical Basis of the Distortion Method

The model is assumed to involve a set of nonlinear ordinary differential equations of the form

$$\dot{x} = f(x, q, u) \tag{1}$$

where x = vector of model state variables measured quantities
q = vector of model parameters
u = system and model input

This equation is solved at each time instant for the parameter vector q, given the measured experimental response x(t) and the input u(t). It is necessary to have available, from measurements, the variable $\dot{x}(t)$ or to derive this by differentiation of the experimental data. Any state variables which are not measured are defined as parameters for the first run of the optimisation and their time histories are used as experimental data in subsequent optimisation runs.

In general there are more parameters than state variables and the problem is overdetermined. It is therefore necessary, in the general case, to postulate a cost function

$$C = \sum_{i=1}^{k} \frac{(q_i - p_i)^2}{w_i^2} \qquad (2)$$

where q_i = parameter to be estimated

p_i = a priori estimate of parameter q_i

w_i = weighting factor based on nominal parameter value.

and to minimise C with the constraint that the model output is at all the time instants exactly equal to the experimental data. This constrained optimisation problem may be solved using Lagrange multipliers and this reduces the problem to the solution of a set of nonlinear algebraic equations. The process must be carried out at each point in the time history of the experimental data.

Application to Coupled Tanks System

Figure 1 is a schematic diagram of a system which consists of two coupled tanks of liquid, a pump, depth sensors and associated electronic hardware. Such a system can provide a basis for introducing students to some practical aspects of automatic control [6] and can also be useful for system modelling investigations and model validation studies [7].

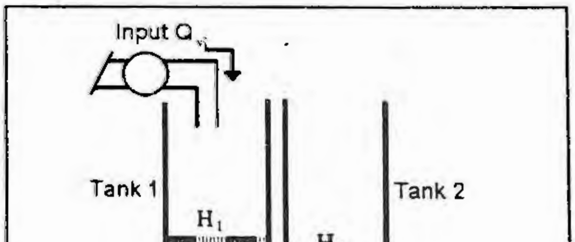


Figure 1. Schematic diagram of the coupled tanks apparatus with relevant system variables.

The system may be described by a pair of ordinary differential equations.

$$A\frac{dH_1}{dt} = Q_{v_i} - Q_{v1} \tag{3}$$

$$A\frac{dH_2}{dt} = Q_{v1} - Q_{v0} \tag{4}$$

where the variables Q_{v_i}, Q_{v1}, Q_{v_o}, H_1 and H_2 are as shown in Figure 1 and the parameter A is the cross-sectional area of each tank. The flow from tank 1 to tank 2 is given by

$$Q_{v_i} = c_{d_1} a_1 \sqrt{2g(H_1 - H_2)} \tag{5}$$

where a_1 is the cross-sectional area of the interconnecting hole, c_{d_1} is the associated coefficient of discharge and g the gravitational constant. Similarly, the outflow, v_o, is conventionally given by an equation

$$Q_{v_o} = c_{d2}\, a_2 \sqrt{2g(H_2 - x_o)} \tag{6}$$

where a_2 and c_{d_2} are parameters for the outlet pipe of tank 2 and x_o is as shown in Figure 1.

The parameters A, a_1, a_2, g and x_o are well known and the main uncertainties of the model relate to the forms of equations (5) and (6) and to the values of the parameters c_{d1} and c_{d2}.

If the forms of equations (5) and (6) are assumed correct the model uncertainties are then associated with the two discharge coefficients and the problem becomes one involving two state variables and two parameters. Since the state variables (H_1 and H_2) can both be measured the optimisation problem inherent in the model distortion approach may be solved without the use of Lagrange multipliers in this case. The problem is to minimise the cost function

$$J = (c_{d1} - c_{d10})^2 + (c_{d2} - c_{d20})^2 \tag{7}$$

(where c_{d10} and c_{d20} are the *a priori* values of the discharge coefficients), subject to the condition

$$\frac{dH_1}{dt} - \frac{1}{A}(Q_{vi} - c_{d1}\, a_1 \sqrt{2g(H_1 - H_2)}) = 0 \tag{8}$$

$$\frac{dH_2}{dt} - \frac{1}{A}\left(c_{d_1} a_1 \sqrt{2g(H_1 - H_2)} - c_{d2}\, a_2 \sqrt{2g(H_2 - x_o)}\right) = 0 \tag{9}$$

where H_1, H_2 and $\frac{dH_2}{dt}$ are all derived from system measurements.

Before applying the model distortion approach to the real system preliminary results were obtained from a simulation-based investigation. In this case the "measurements" were generated from a simulation model in which the coefficient c_{d_2} had a value which varied in a specific way with the liquid level H_2. Figure 2 shows step responses for this "system" and for a model based on equations (3) to (6) with fixed values of c_{d1} and c_{d2} chosen to give an exact match to the "system" response time histories during the initial steady state prior to the application of the step change of input flow. These values for the discharge coefficients of the model were also the values adopted for c_{d10} and c_{d20}.

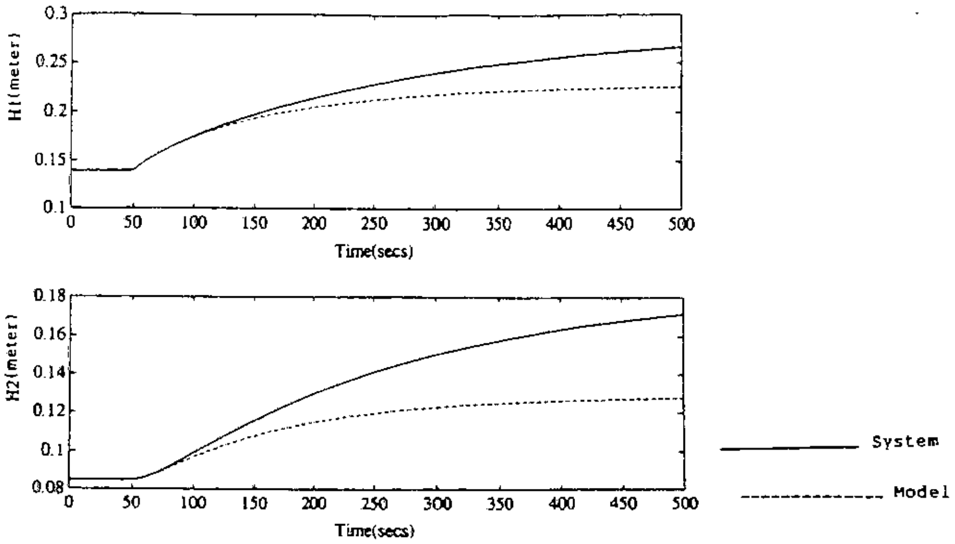


Figure 2. H_1 and H_2 responses to a step change of input flow for the simulated coupled tanks system and the nominal model with fixed parameters.

Figure 3 shows the time histories of c_{d_1} and c_{d_2} obtained from application of the model distortion method. The results show clearly that there has to be a significant distortion of parameter c_{d_2} in order to make the model responses match the "measured" output time histories. Apart from transients of very short duration which appear at the time of application of the step input these plots are exactly as expected from the known properties of the model which was used to generate the "measured" variables.

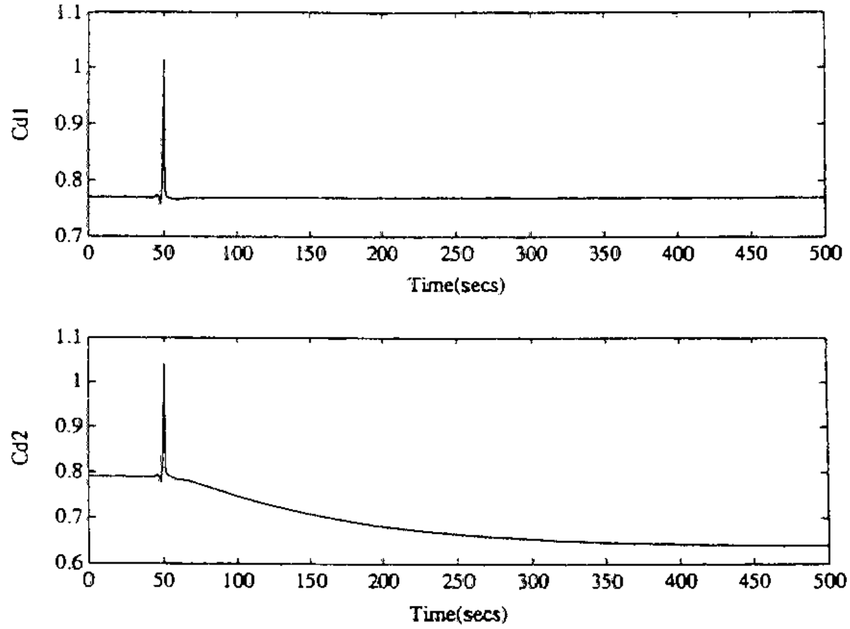


Figure 3. Time histories of c_{d_1} and c_{d_2} by the model distortion method for the simulated coupled-tanks system.

Figure 4 shows similar results for a case in which coloured noise (bandlimited to 1 Hz) of amplitude 1% of the total output variation was added to the "measured" data. It may be seen that the effects of the added noise are considerably amplified in the time histories of the distorted parameters.

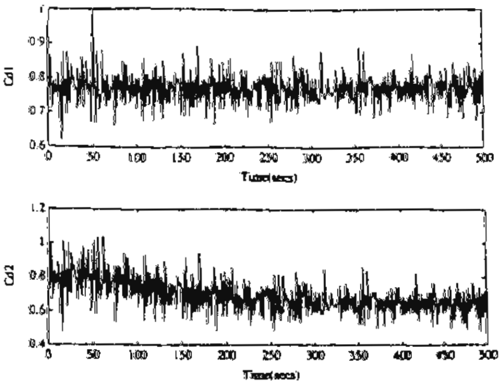


Figure 4. Time histories of c_{d_1} and c_{d_2} for the simulated coupled-tanks system with 1% added noise on measurements.

Results from the application of the distortion method to step response data gathered from the real coupled tank hardware are shown in Figures 5 and 6. In order to minimise the effects of measurement noise all data channels were prefiltered using a fifth-order Butterworth low-pass filter with a cut-off frequency of 0.1 Hz. The parameter distortion time histories suggest that the relationship for the outflow from the second tank is not appropriate since the coefficient of discharge $c_{d_2}$is not a constant. On the other hand the results do suggest that the model structure associated with the first tank is satisfactory since the distortion of c_{d_1} is negligible. These results are consistent with findings from other studies involving the coupled tanks system [7,8].

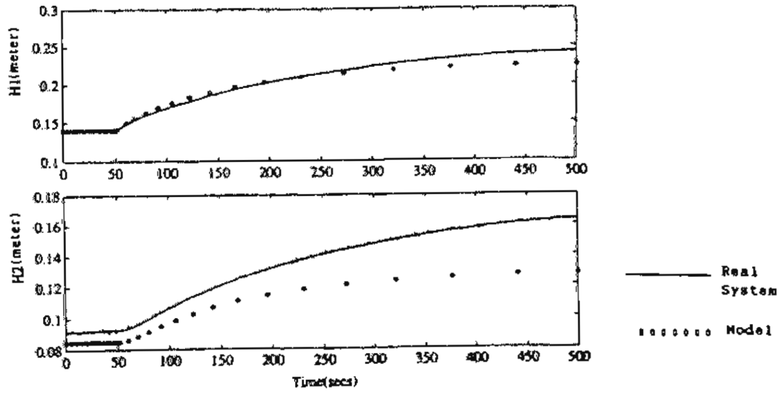


Figure 5. Step responses of the coupled tank system and the mathematical model for nominal values of c_{d_1} and c_{d_2}.

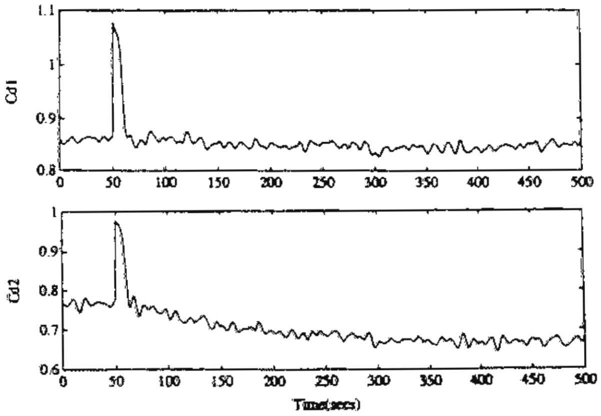


Figure 6. Time histories of c_{d_1} and c_{d_2} to ensure that the model exactly matches the system output variables (after pre-filtering of measured response data).

Discussion and Conclusions

The optimisation process inherent in the model distortion approach is c.p.u. intensive. The implementation used for the work described in this paper was based on MATLAB and run times of several hours on Pentium P5/166 processors are typical. This is a disadvantage in comparison with other methods for external validation, such as those based on standard system identification and parameter estimation tools. Experiments with simulated response data for the coupled-tanks system, and also for simpler linear models, have shown that the variance of the parameter time histories provides a potentially useful measure of model adequacy. However, this type of information does not appear to provide a very clear indication of sources of model structure inadequacy although errors in parameter values can be estimated from the distortion information in some cases.

Measurement noise presents a major problem with the model distortion approach. This arises because the constraint equations use the time derivatives of the measured state variables. Problems are particularly severe in a steady state condition where the noise component may dwarf the derivative of the data and this will be reflected in the parameter distortion time histories. Cameron in his paper on linear system approach [4] concluded that model distortion methods are useful only with noise free data and the results found in this investigation for the more general nonlinear case agree with his findings.

References

1. Butterfield, M.H. and Thomas, P.J.: Methods of quantitative validation for dynamic system models, *Trans. Inst. Meauremenr and Control*, 8 (1986), 182-219.
2. Butterfield, M.H.: A method of quantitative validation based on model distortion, *Trans. Inst. Measured and Control*, 12 (1990), 167-173.
3. Li, C.L.R.: Application of distortion technique for model validation to nuclear power plant - AGR scatter plug, *Trans. Inst. Measurement and Control*, 8 (1986), 220-232.
4. Cameron, R.G.: Model validation by the distortion method: linear state space systems, *IEE Proc.-D*, 139 (1992), 296-300.
5. G.J. Gray: Development and Validation of Nonlinear Models for Helicopter Dynamics, Ph.D. Thesis, University of Glasgow, 1992.
6. Wellstead, P.E.: Coupled Tanks Apparatus: Manual, TecQuipment Ltd., 1981.
7. Murray-Smith, D.J. and Gong, Mingrui: A practical Exercise in simulation model validation, *Proceedings 1. MATHMOD VIENNA*, Vienna, February 1994, 231-234.
8. Gray, G.J., Murray-Smith, D.J., Li, Y. and Sharman, K.C.: Nonlinear system modelling using output error estimation of a local model network, *Proceedings Summer Computer Simulation Conference*, Portland, Oregon, July 1996, 460-465.

THE TEARING PROBLEM: DEFINITION, ALGORITHM AND APPLICATION TO GENERATE EFFICIENT COMPUTATIONAL CODE FROM DAE SYSTEMS

E. Carpanzano and R. Girelli
Dipartimento di Elettronica ed Informazione, Politecnico di Milano
Piazza Leonardo Da Vinci 32, 20133 Milano, Italy
E-mail: carpanza@elet.polimi.it

Abstract. The numerical solution of DAE systems can be executed much more efficiently if a preliminary symbolic manipulation is performed. A very important step to achieve an efficient computational code for DAE systems is to solve the equations for as many algebraic variables as possible; this way the variables and equations are divided into two sets, so that it is easy to solve for the variables in the first set if the variables of the other set are known. This kind of partitioning is called tearing. The aim is to hide the variables of the first set and let the DAE-solver treat only the variables of the second set.

Though numerous tearing algorithms have been proposed, there are no clear winners.

In the present work, first, the tearing operation is defined. Second, the NP-completness of the tearing problem is proved by restriction. Then a simple formulation of the tearing problem is given by means of a bipartite graph. and a flexible algorithm is proposed, which allows to easily implement both general heuristic rules and domain specific heuristic rules. Finally an application of the algorithm in MOSES (Modular Object oriented Software Environment for Simulation) is shown.

1. Introduction

In object oriented modelling, models are described as closely as possible to the corresponding physical system. Models are defined in a declarative form, so that one software module is associated to one physical component indipendently of the context in which it is used. However this form of model representation cannot efficently be used for simulation, it is convenient to generate the procedural form.

If models are written based on first principles equations, their simulation often requires solving a system of differential-algebraic equations (DAE). Generally a DAE system has the form (1):

$$\mathbf{F}\,(t, y, \dot{y}, u, p) = 0 \tag{1}$$

where y is the unknown variables vector, u is the input variables vector, p is the parameters vector and t is time.

Especially for complex plants, the DAE system is of very large dimensions, so its solution would require excessively long computation times. The numerical solution of a DAE system can be executed much more efficiently if a preliminary symbolic manipulation is performed; this can be done by means of various expedients. A very important step to achieve an efficient computational code for DAE systems is to solve the equations for as many algebraic variables as possible. This way the variables and equations are divided into two sets, one of assignments and one of implicit equations, so that it is easy to solve for the variables in the first set if the variables of the other set are known. This kind of partitioning is called tearing. The aim is to hide the variables of the first set and let the DAE-solver treat only the variables of the second set [1, 2, 5, 6, 12, 14].

Consider the unknown variables set y as two subsets, x the state variables (i.e. those variables which are present in (1) together with their derivatives) and z the algebraic variables. It is possible to rewrite (1) in the following form :

$$\mathbf{F}\,(t, x, \dot{x}, z) = 0 \tag{2}$$

ommitting u and p for the sake of simplicity. The above mentioned order reduction of the DAE system is obtained by choosing the vector $\tilde{z}$ of minimum dimension such that (2) can be written as:

$$\begin{aligned}
z_1 &= g_1\,(t, x, \dot{x}, \tilde{z})\\
z_2 &= g_2\,(t, x, \dot{x}, \tilde{z}, z_1)\\
&\vdots\\
z_k &= g_k\,(t, x, \dot{x}, \tilde{z}, z_1, \ldots, z_{k-1})\\
\tilde{\mathbf{G}}\,(t, x, \dot{x}, \tilde{z}, z_1, \ldots, z_k) &= 0
\end{aligned} \tag{3}$$

After substituting of variables $z_1, z_2, \dots, z_k$ into $\tilde{\mathbf{G}}$, the problem is reduced to the solution of the DAE system $\mathbf{G}(t, x, \dot{x}, \tilde{z}) = 0$. So the dimension of the DAE system is reduced to that of the function **G**.

Particularly we are interested in finding the subset $\tilde{z}$ as small as possible, i.e. to divide the equations of the given DAE system into two subsets so that the first one (assignments) is as large as possible and the second one (implicit equations) is as small as possible; in this way the DAE solver, which treats only the implicit equations[1], can compute **G** more efficiently[2].

2. Formulation of the tearing problem on the reduced incidence matrix

Let's now introduce the **reduced incidence matrix** of the DAE system: this is a matrix whose rows and columns represent the equations and the algebraic variables of the system respectively, and whose element (i,j) is not zero if the j-th algebraic variable is present in equation i and is 0 otherwise. Moreover a non zero element (i,j) of the considered matrix is bold if equation i can be solved[3] for the variable j (figure 2a).

With reference to the reduced incidence matrix our problem can be reformulated as follows: given the matrix, reorder its rows and columns in such a way as to transform it in bordered triangular form, BTF (figure 1), with the triangular part as large as possible, and with bold elements on the diagonal of the triangular part.

Infact, the equation corresponding to the k-th row of the triangular part can be solved with respect to the variable corresponding to the k-th column of the triangular part; in this way assignments are obtained.

The equations corresponding to the rows of the border are the implicit equations, that are used by the DAE solver to compute the value of the remaining variables of the system, which are the ones corresponding to the columns of the border and the state variables (not represented by any column of the reduced incidence matrix).

Consider for example figure 1. In this case we have a system of order n with n equations and n variables; the reduced incidence matrix has n rows and m ≤ n columns. By reordering rows and columns of the matrix we obtain the illustrated BTF, with triangular part of size p. Now we can obtain for our system p assignments and n-p implicit equations.

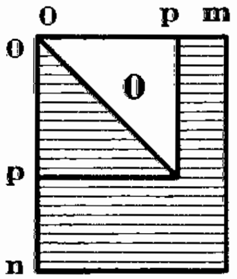


Figure 1 - Bordered Triangular Form.

3. NP-completness of the tearing problem

In this section it is shown that the tearing problem is NP-complete. This means that the optimal solution for the tearing problem can not be guaranteed in polynomial time by deterministic algorithms, but only by nondeterministic algorithms[4].

There are different techniques for proving the NP-completness of a given problem; here, the NP-completness of the tearing problem Π is proven by *restriction*, showing that the problem contains a known NP-complete problem Π' as a special case [9]. The heart of such a proof lies in the specification of the additional restrictions to be placed on the instances of Π so that the resulting restricted problem becomes identical to Π'.

Proof: suppose we've got a structurally consistent[5] system of n equations and n algebraic variables [2, 12], and that every equation can be solved with respect to all of its variables. With these additional restrictions the above mentioned reduced incidence matrix coincides with the incidence matrix M.

[1] Since DAE solvers produce a solution within the user-specified error bounds only for x, we can't guarantee any error bound for hidden variables, [12]. If accuracy of z_k is a real concern, the assignments must have the form: $z_k = g_k(t, x, z_1, \dots, z_{k-1})$.

[2] The efficiency may also be improved, if it is first possible to reorder the DAE system (2) into block lower triangular form (*BLT partitioning*), then splitting every subsystem into assignments and implicit equations, [2].

[3] In order to establish if equation i can be solved for variable j, a convenient analysis is performed. This analysis, that has to be done symbolically, is very complex, therefore it's executed in an approximate way. It is to point out that the number of non zero bold elements in the reduced incidence matrix strongly depends on the accuracy of this analysis.

[4] A nondeterministic algorithm is an abstract computational structure where the algorithm is able to test different possible choices at the same time, in order to point directly to the final solution [9].

Now, the tearing problem can be formulated as the problem of transforming the incidence matrix M to the optimal bordered triangular form, where by optimal we mean with the biggest triangular submatrix.

In order to solve our problem we shall use a graph theoretic approach. We begin by introducing the following definitions and remarks.

Since the considered system is structurally consistent, we've got that the incidence matrix M is structurally non singular; then it's possible to permute the rows to obtain a matrix J with a nonzero diagonal (usually called finding a transversal) [2, 4]. Obviously the tearing problem can be studied with reference to J instead of M.

For the matrix J with nonzero diagonal, we define an *associated digraph* G (J) as follows. If n is the dimension of the incidence matrix (the order of the system) then G (J) has n nodes and there is a directed edge from node m to node k if and only if $j_{mk} \neq 0$ for $m \neq k$, where $J = [j_{mk}]$. Note that the self-loops corresponding to the diagonal entries j_{mm} of J are not represented in the associated digraph.

A digraph is said to be acyclic if it does not have circuits. A set of nodes S of a digraph is called an essential set if the digraph obtained by removing the nodes of S is acyclic. An essential set having the minimum number of nodes is called a *minimum essential set*.

It is known that a matrix is transformable, by row and column permutations, to triangular form if and only if the associated digraph is acyclic [3].

The problem we are studing can now be reformulated with reference to the associated digraph, by observing that the optimal bordered triangular form for the incidence matrix is known once a minimum essential set for the associated digraph is known. Infact, the rows and columns corresponding to the vertices of the minimum essential set form the border, while the rows and columns corresponding to the remaining vertices of the associated digraph (which form an acyclic digraph) constitute the triangular part of the optimal BTF.

At this point the tearing problem can be considered as the problem of determining a minimum essential set of a digraph. This problem is well known in literature as the feedback vertex set (FVS) problem[6]; this is a classical NP-complete problem, and appears in Karp's seminal paper [11].

So, it has been found that the considered problem contains a known NP-complete problem: in this way the NP-completness of the tearing problem is proved by restriction. □

4. Tearing algorithm

Since the tearing problem is NP-complete, it's not possible to elaborate a deterministic algorithm that guarantees to achieve the optimal solution in polynomial time. So we've to be satisfied with an approximation algorithm.

Numerous approximation tearing algorithms have been proposed in different contexts, either to split a DAE system in assignements and implicit equations [5, 13, 14], or to transform a matrix to bordered triangular form [3, 4], or to find out the minimum essential set of a digraph [7, 10]; all of these algorithms deal with the same problem, but among them there are no clear winners. In this work an efficient and flexible algorithm is proposed. But first a simple formulation of the tearing problem is given by means of a bipartite graph.

4.1 The associated bipartite graph

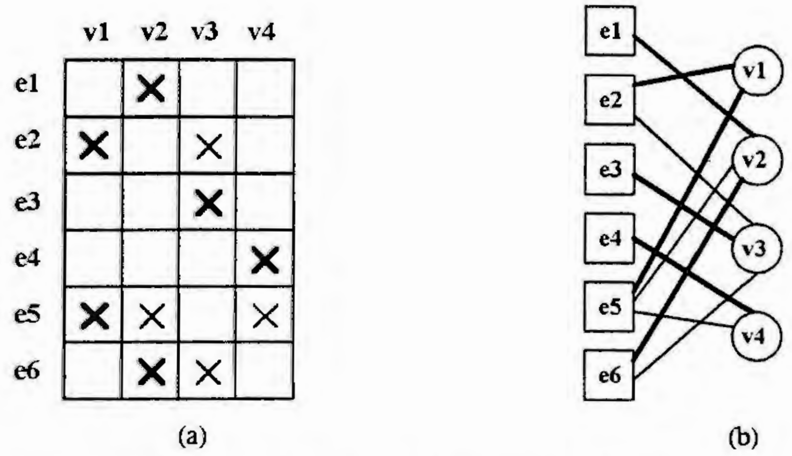
For a given DAE system we can deduce the corresponding reduced incidence matrix R as shown before; from this matrix we can then deduce the **associated bipartite graph** as follows. In the bipartite graph we have two sets of nodes: E-nodes (squares), one for each row of R, and V-nodes (circles), one for each column of R. There is an edge from an E-node e_i to a V-node v_j if and only if the element in position (i, j) in R is non zero, and the edge is bold if and only if the corresponding element is bold (figure 2).

With reference to the DAE system, an E-node stands for an equation, and a V-node stands for an algebraic variable, and there is an edge from e_i to v_j if and only if the equation associated to e_i contains the variable associated to v_j, and this edge is bold if and only if e_i can be solved for v_j.

In the following sections we will show how the use of a bipartite graph to formulate the tearing problem, allows working out an algorithm which turns out to be simple and efficient to implement directly on the incidence matrix, once an efficient data structure for the incidence matrix of the DAE system is available [7].

[5] A system is structurally consistent if it is possible to form a set of ordered pairs of variables and equations, such that each variable x_j and each equation $h_i = 0$ are only members of one pair and for each pair (x_j , h_i =0) the variable x_j appears in the expression $h_i = 0$.

[6] In the FVS problem, one is given a directed graph and is asked to find the minimum subset of vertices that intersects every directed cycle in the graph, [9].



	v1	v2	v3	v4
el		**X**		
e2	**X**		X	
e3			**X**	
e4				**X**
e5	**X**	X		X
e6		**X**	X	

Figure 2 - Reduced Incidence Matrix (a) and corrisponding Bipartite Graph (b).

4.2 Structure of the tearing algorithm

Our problem is to find out for the system as many assignments as possible. Since an assignment is obtained whenever there is an equation with only one unknown algebraic variable z_k and the equation is solvable with respect to z_k, an assignment is defined whenever, in the associated bipartite graph, an E-node e_k is found having only one incident edge and this edge is bold. So we've to operate on the graph so as to get as many E-nodes with only one incident and bold edge as possible. In particular, we can proceed by executing the following algorithm:

```
Algorithm
variables:       assVars;        (set of assigned vars)
                 compVars;       (set of vars computed by the numerical solver)
                 assEqs;         (set of assignments to compute assVars)
                 impEqs;         (set of implicit equations to compute compVars)
                 bipGr.          (data structure for the bipartite graph)
inizialization:  initialize assVars, compVars, assEqs, impEqs, bipGr;
                 given the DAE system, the associated bipartite graph is built.
begin
  put the associated bipartite graph in bipGr. put state vars in compVars.
  search the V-nodes $v_k$ with no incident bold edges in bipGr. put the variables $v_k$ in compVars.
  remove the V-nodes $v_k$ and all their incident edges from bipGr.
  repeat
    if (there is one E-node with only one incident and bold edge ($e_i$ - $v_l$) in bipGr)
       then  [ solve equation $e_i$ for variable $v_l$ . put equation $e_i$ in assEqs. put variable $v_l$ in assVars.
               remove the V-node $v_l$ , the E-node $e_i$ and all their incident edges from bipGr. ]
       else  [ select a V-node $v_r$ in bipGr. put variable $v_r$ in compVars
               remove the V-node $v_r$ and all its incident edges from bipGr. ].
  untill (there is at least one V-node in bipGr).
  put the remaining equations in impEqs.
end
```

At every cycle of the algorithm one of the following two operations is executed:

1 - V-node v_l , E-node e_i and all their incident edges are removed from the bipartite graph, (variable v_l and equation e_i are removed from the DAE system), or:

2 - only V-node v_r and its incident edges are removed from the bipartite graph, (only variable v_r is removed from the DAE system).

[7] An efficient data structure for the incidence matrix of a DAE system allows several symbolic manipulation algorithms to be efficiently performed; so it's reasonable to suppose this data structure available in our context.

In case 1 an assignment is identified, while in case 2 an implicit equation is found. Infact in case 1 equation e_i can be solved for the unknown algebraic variable v_l, to obtain an assignment. In case 2, we suppose that one of the unkown algebraic variables is known in order to find out further assignments, therefore we remove V-node v_r and its incident edges from the associated bipartite graph. The variable corresponding to v_r will be computed by the implicit system solver, which needs, consequently, one more equation. This equation is implicitly generated by removing V-node v_r from the graph, since there will be one E-node more remaining at the end of the algorithm. Thus there will be one more implicit equation.

The number of assignments we achieve, by performing the illustrated algorithm, depends on the way V-node v_r is chosen in the underlined instruction. This choice can be done by making use of both general heuristic rules and domain specific heuristic rules, as is discussed in the following sections.

4.3 General heuristic rules

Many tearing algorithms based on heuristic rules have been proposed in the least thirty years; a complete study and comparison of these algorithms is outside the scope of the present work, and the interested reader is referred to the existing literature: overviews of the most popular algorithms proposed in the 70's and 80's are presented in [4] and [13], while more recent algorithms are illustrated in [7] and [10].

In this paper we're just going to propose three simple general heuristic rules. Bearing in mind that we've to operate on the bipartite graph in such a way as to get as many E-nodes with only one incident bold edge as possible, we can formulate the following rules to select v_r in the algorithm given above:

1. select v_r among the V-nodes connected, if possible by a not bold edge, to one of the E-nodes having minimum number of incident edges;
2. give every V-node a weight equal to $p_i = [(1 + 1/n)\, n_{nb} + n_b]$, where n_{nb} and n_b are the numbers of not bold and bold incident edges, respectively, and n is the number of all the V-nodes, and then select v_r among the V-nodes with maximum weight[8];
3. first give every edge a weight equal to the inverse of the number of edges incident to its E-node, then give every V-node a weight equal to the sum of the weights of its incident edges, and then select v_r among the V-nodes with maximum weight.

Since the shown rules make use only of local information of the bipartite graph, they can be easily implemented and efficiently executed once we've an efficient data structure for the reduced incidence matrix.

4.4 Domain specific heuristic rules

Since we are interested in modelling and simulating whatever physical system, it's important to make the symbolic manipulation environment indipendent from specific domains, i.e. general algorithms have to be used. On the other side we have to consider that using rules for specific domains can significantly improve the results of the symbolic manipulation, particularly when the problem has no general optimal solution, like an NP-complete problem. So it's convenient to implement general symbolic manipulation algorithms, which may also use rules for specific domains, if these are available. These rules can be contained in the models library itself or can be generated by an interface between the models library and the symbolic manipulation environment. Let's now consider the case of object oriented modelling of multibody systems.

Heuristic rules for tree-structured multibody systems

For the sake of simplicity we first consider tree-structured multibody systems. Multibody systems are called "tree-structured", when the connection structure of bodies and joints forms a "tree". Typical examples for tree-structured systems are robots. Tree-structured systems can be solved, by using the Newton-Euler algorithm [8], in such a way that, once all the state variables are known, it's possible to obtain all the remaining kinematic and dynamic variables through assignments. Particularly this can be achieved by solving the kinematic equations for the kinematic variables from the base to the end-effector, and by solving the dynamic equations for the dynamic variables in the opposite direction. Let's now see how this rule can be implemented directly in the mechanical systems library.

One way could be to give hints to the symbolic manipulator by suggesting the variables with respect to which the equations have to be solved for. But there is a simpler way to integrate the *Newton-Euler algorithm* with the general heuristic tearing rules explained in the previous section. Suppose that in our modelling environment every component of the tree stuctured mechanical system has two mechanical terminals, one

[8] This means to select among the V-nodes with the maximum number of total incident edges, one with the minimum number of bold incident edges.

directed towards the base (MT1) and one towards the end effector (MT2). In order to allow the symbolic manipulation environment to assign all the kinematic and dynamic variables, once the state variables are known, it's sufficient to write the models of tree-structured systems components in such a way that the kinematic equations are solved for the variables of the terminal MT2 and that the dynamic equations are solved for the variables of the terminal MT1.

If multibody models are written in this way, then, just by performing the tearing algorithm with the general heurisitic rules explained before, the maximum number of assignments, i.e. the minimum number of implicit equations, is achieved, as will be shown in section 5.

Heuristic rules for multibody systems with kinematic loops

Multibody systems with kinematic loops[9] can be handled by cutting joints such that the resulting system has a tree structure. The removed cut-joints are thereby replaced by appropriate (unknown) constraint forces and torques. Furthermore, the kinematic constraint equations of the cut-joints are added as additional equations to the equations of motion of the tree-structured system.

In order to treat properly mechanical systems with kinematic loops we have to define *cut-objects* in the models library for mechanical systems. In this way the analyst, who develops the model, has to use the necessary cut-objects, when building up a model with kinematic loops, so that the model can be reduced to a tree-structured system by the symbolic manipulator, and the rules explained before can be used. If this is done, the maximum number of assignments is obtainable. More about tearing in mechanical systems with kinematic loops can be found in [5].

5. Results

In this section the results obtained by implementing the proposed algorithm in MOSES (Modular Object-oriented Software Environment for Simulation) [15] are shown. Particularly we're interested in modelling robots with 1, 2, 3, 4, 5 or 6 links, i.e. with 1, 2, 3, 4, 5 or 6 degrees of freedom. The results obtained are represented in table 1, where we have the number of total scalar equations of the considered models (row 1) and the number of implicit equations we obtain by performing the tearing algorithm; in particular we consider what happens when we use only general heuristic rules[10] (row 2) and when we use also domain specific rules (row 3).

Model:	1 link	2 link	3 link	4 link	5 link	6 link
1. total number of equations	79	167	255	343	431	519
2. only general rules	6	9	11	13	20	23
3. general and domain specific rules	2	4	6	8	10	12

Table 1 - Results achieved by performing the proposed tearing algorithm.

From the results shown in the table, we notice that the use of domain specific rules is very important, since by using these rules we manage to achieve the optimal solution, i.e. the minimum number of implicit equations; infact, in the third row of table 1 the number of equations is equal to the the number of state variables of the corresponding model. On the other side, when only general heuristic rules are used, the number of implicit equations is greater than the number of state variables: so the obtained solution is clearly sub-optimal.

6. Conclusions

In this paper the tearing problem has been discussed. First, the problem has been defined and its NP-completness has been proven by restriction. Then a simple formulation of the problem has been given by means of a bipartite graph and an efficient and flexible algorithm has been proposed. This algorithm uses general heuristic rules and allows also to use, if they are available, rules for specific domains. Moreover, the considered algorithm turns out to be easy to implement in an efficient way, once we have an efficient data structure for the incidence matrix of the DAE system in our symbolic manipulation environment. Finally the results obtained by implementing the algorithm in MOSES are illustrated.

[9] These systems have index greater than one. Since all known DAE solvers have troubles with solving problems of index greater than one, an *index reduction* has also to be performed [2, 12], though not discussed here.

[10] In this tests the general heuristic rule number 1 of section 4.3 has been used.

Acknowledgment. We would like to thank Prof. Claudio Maffezzoni for constant support and for many helpful discussions.

References

1. F.E. Cellier and H. Elmqvist. Automated Formula Manipulation Supports Object Oriented Continuous System Modelling. IEEE Control System, 1993.

2. E. Carpanzano and F. Formenti. Symbolic Manipulation of DAE Systems. Master Thesis. Politecnico di Milano. 1994.

3. L.K. Cheung and E.S. Kuh. The Bordered Triangular Matrix and Minimum Essential Sets of a Digraph. IEEE Transactions on Circuits and Systems, vol. cas.-21, n° 5, 1974.

4. I.S. Duff, A.M. Erisman and J. Reid. Direct Methods for Sparse Matrices. Clarendon Press, Oxford, 1986.

5. H. Elmqvist and M. Otter. Methods for Tearing Systems of Equations in Object Oriented Modelling. Proceedings of the Conference on Modelling and Simulation, 1994.

6. H. Elmqvist. Object Oriented Modelling and Automatic Formula Manipulation in Dymola. Scandinavian Simulation Society, 1993.

7. G. Even. J. Naor, B. Schieber and M. Sudan. Approximating Feedback Sets and Multi-Cuts in Directed Graphs. IPCO, 1995.

8. G. Ferretti. Systematic Dynamic Modelling of Mechanical Systems Containing Kinematic Loops. MMOS, volume 2, number 3, 1996.

9. M.R. Garey and D.S. Johnson. Computers and Intractability. A guide to the Theory of NP-completness. W.H. Freeman and Company, San Francisco, 1979.

10. R. Hussin and S. Rubinstein. Approximations for the maximum acyclic subgraph problem. Information Processing Letters 51, pp. 133-140, 1994.

11. R.M. Karp. Reducibility among Combinatorial Problems, in R.E. Miller and J.W. Thatcher, Complexity of Computer Computations, pp. 85-104, Plenum Press, New York, 1972.

12. S.E. Mattsson, M. Andersson and K.J. Åström. Object Oriented Modelling and Simulation. In Linkens D.A., editor, CAD for Control Systems. Marcel Decker, 1993.

13. R.S. Mah. Chemical Process Structures and Information Flow. Butterworths Series in Chemical Engineering, 1993.

14. C. Maffezzoni, R. Girelli and P. Lluka. Generating Efficient Computational Procedures from Declarative Models. Simulation Practice and Theory 4, pp. 303-317, 1996.

15. C. Maffezzoni and R. Girelli. Object Oriented Database Support For Modular Modelling And Simulation. Modelling and Simulation ESM 94, Barcellona, 1994.

TRANSFER FUNCTION MODELS FOR MULTIDIMENSIONAL SYSTEMS

Rudolf Rabenstein
Lehrstuhl für Nachrichtentechnik, Universität Erlangen-Nürnberg
Cauerstrasse 7, D-91058 Erlangen, Germany
Tel.: *9131-858717, Fax: *9131-303840, Email: rabe@nt.e-technik.uni-erlangen.de

Abstract. Multidimensional systems describe relations between signals depending on two or more independent variables, like time and space. They are also called distributed parameter systems. The only conventional model for their description are partial differential equations. This is in contrast to onedimensional (lumped parameter) systems, where a variety of different models including transfer functions is used. This paper extends the concept of transfer function models to multidimensional systems. They are useful for system analysis and as a starting point for the derivation of discrete simulation models.

Introduction

Continuous systems can be divided into systems with states depending on either one or more than one independent variables. In many physical and technical applications these are the time coordinate or the time and space coordinates. These systems are called one- and multidimensional systems, respectively.

Onedimensional systems with time as the only independent variable are also called lumped parameter systems and are mathematically described by ordinary differential equations (ODE). Furthermore, a variety of other models exists, like state space models, flow graphs, network descriptions, etc. Transfer function models based on frequency domain methods are very popular in electrical engineering. Many advances in design, control, and simulation of onedimensional systems are based on the choice of appropriate models.

Multidimensional (MD) systems describe relations between signals depending on two or more independent variables, like time and space. They are also called distributed parameter systems (DPS). Typical applications include wave propagation, heat and mass transfer, and fluid dynamics. The only model of widespread use are partial differential equations (PDEs). Other types of models are rarely considered. In many cases, the PDE model of a MD system is taken as a synonym for the system itself. Frequently, even the standard literature on DPS does not distinguish between the MD system and the PDE model (e.g. [14]). As a consequence, the algorithms for control and simulation of MD systems are almost exclusively based on the numerical solution of PDEs by Finite Element and Finite Difference methods. The computational burden associated with these methods is well known. This paper extends the concept of transfer function models based on functional transformations to multidimensional systems. It is an extension of the functional transformation approach presented in [7, 9, 10].

Transfer Function Description of Multidimensional Systems

Motivation and Previous Work

Transfer functions are an established model in linear systems theory, electrical network theory, and control theory. Not only are they indispensible for theoretical considerations, they als allow to derive effective algorithms (e.g. fast convolution). Recent advances in image coding are also based on the spectral decomposition of image sequences. The representation of spatially distributed signals by spectral decomposition into eigenfunctions, modes, characteristic frequencies, and alike is a classical method. However, it has been mainly used to provide analytical solutions for PDEs [1, 3, 5, 8, 15]. In other words, frequency domain descriptions have been applied to the output signals of MD systems, but not to the systems themselves. Therefore it seems worthwhile to represent not only MD signals in the frequency domain, but also the MD systems which process and generate such signals.

Preview

A necessary requirement for the transfer function description of MD systems is the choice of suitable functional transformations for the time and space variables. The derivation of a transfer function model from a PDE proceeds in two steps

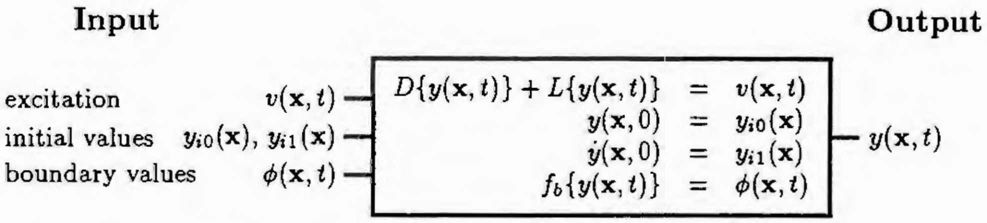

Input
Output
excitation $v(\mathbf{x},t)$
initial values $y_{i0}(\mathbf{x})$, $y_{i1}(\mathbf{x})$
boundary values $\phi(\mathbf{x},t)$
$D\{y(\mathbf{x},t)\} + L\{y(\mathbf{x},t)\} = v(\mathbf{x},t)$
$y(\mathbf{x},0) = y_{i0}(\mathbf{x})$
$\dot{y}(\mathbf{x},0) = y_{i1}(\mathbf{x})$
$f_b\{y(\mathbf{x},t)\} = \phi(\mathbf{x},t)$
$y(\mathbf{x},t)$

Figure 1: Input-Output Description of a MD System in the space-time domain

- Laplace transformation with respect to time turns an initial-boundary value problem into a pure boundary value problem for the space variable.
- A suitable transformation for the space variable converts the boundary value problem into an algebraic equation, which yields the transfer function of the system.

Problem Description

The derivation of a transfer function model will be shown for a MD system represented by the PDE

$$D\{y(\mathbf{x},t)\} + L\{y(\mathbf{x},t)\} = v(\mathbf{x},t) \qquad \mathbf{x} \in V \,, \tag{1}$$

where $\mathbf{x}$ is the vector of space coordinates defined on a domain V and t is the time coordinate with $t > 0$. $v(\mathbf{x},t)$ is an excitation function and $y(\mathbf{x},t)$ is the response of the system. The operator $D\{\cdot\}$ for the time derivation is given by

$$D\{y(\mathbf{x},t)\} = a_2\ddot{y}(\mathbf{x},t) + a_1\dot{y}(\mathbf{x},t) + a_0 y(\mathbf{x},t) \,, \tag{2}$$

where $\dot{y}$ and $\ddot{y}$ are the first and second time derivative. $L\{\cdot\}$ denotes a self-adjoint operator for spatial derivation. The initital conditions at $t = 0$ are given by

$$y(\mathbf{x},0)\} = y_{i0}(\mathbf{x}) \qquad \dot{y}(\mathbf{x},0)\} = y_{i1}(\mathbf{x}) \qquad \mathbf{x} \in V \,. \tag{3}$$

On the surface S of V, inhomogeneous boundary conditions of 1., 2., and 3. kind (Dirichlet, Neumann, Robin) are specified by

$$f_b\{y(\mathbf{x},t)\} = \phi(\mathbf{x},t) \qquad \mathbf{x} \in S \,. \tag{4}$$

Contrary to most work on PDEs, we are not primarily interested in the analytical or numerical calculation of the response $y(\mathbf{x},t)$, but in obtaining a model for the MD system described by (1)–(4). To illustrate the difference, we consider the given quantities as input signals and the system response $y(\mathbf{x},t)$ as output signal of the MD system shown in fig. 1. In the sense of linear systems theory, it is a linear and time-invariant system.

We start from the PDE model (1)–(4) and attempt to convert it into a transfer function model. The suitable mathematical tools for this purpose are functional transformations.

Transformation with Respect to Time

Laplace Transformation. Application of the Laplace transformation to the time variable turns the initial-boundary value problem (1)–(4) into a pure boundary value problem for the space variable. With the definition of the Laplace-transformation and the derivation theorem for the first and second derivative

$$\begin{aligned}
\mathcal{L}\{y(\mathbf{x},t)\} &= Y(\mathbf{x},s) = \int_0^\infty y(\mathbf{x},t)\, e^{-st}\, dt && (5)\\
\mathcal{L}\{\dot{y}(\mathbf{x},t)\} &= sY(\mathbf{x},s) - y(\mathbf{x},0) && (6)\\
\mathcal{L}\{\ddot{y}(\mathbf{x},t)\} &= s^2Y(\mathbf{x},s) - sy(\mathbf{x},0) - \dot{y}(\mathbf{x},0) \,, && (7)
\end{aligned}$$

we arrive after some manipulations at a derivation theorem for the operator $D\{\cdot\}$

$$\mathcal{L}\{D\{y(\mathbf{x},t)\}\} = \gamma^2(s)Y(\mathbf{x},s) - \mathbf{B}_i^T(s)\mathbf{y}_i(\mathbf{x}) \tag{8}$$

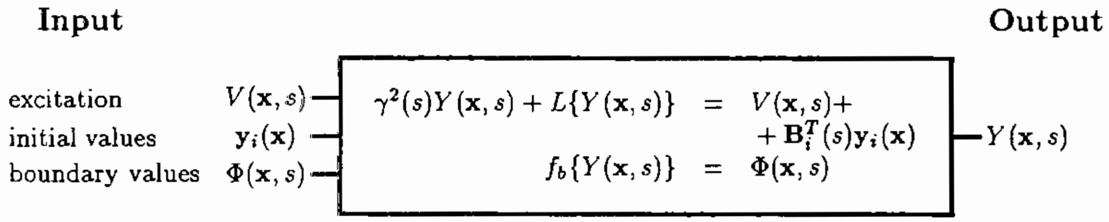

Input
Output
excitation
initial values
boundary values
$V(\mathbf{x},s)$
$\mathbf{y}_i(\mathbf{x})$
$\Phi(\mathbf{x},s)$
$\gamma^2(s)Y(\mathbf{x},s)+L\{Y(\mathbf{x},s)\} = V(\mathbf{x},s)+ +\mathbf{B}_i^T(s)\mathbf{y}_i(\mathbf{x})$
$f_b\{Y(\mathbf{x},s)\} = \Phi(\mathbf{x},s)$
$Y(\mathbf{x},s)$

Figure 2: Input-Output Description of a MD System in a space-temporal frequency domain

with

$$\gamma^2(s) = a_2s^2 + a_1s + a_0\,, \qquad \mathbf{B}_i(s) = \begin{bmatrix} a_2s + a_1 \\ a_2 \end{bmatrix}, \qquad \mathbf{y}_i(\mathbf{x}) = \begin{bmatrix} y(\mathbf{x},0) \\ \dot{y}(\mathbf{x},0) \end{bmatrix} = \begin{bmatrix} y_{i0}(\mathbf{x}) \\ y_{i1}(\mathbf{x}) \end{bmatrix}. \tag{9}$$

Boundary Value Problem. Applying (8) to the initial-boundary value problem (1)–(4) results in a boundary value problem for the corresponding Laplace transforms

$$\gamma^2(s)Y(\mathbf{x},s) + L\{Y(\mathbf{x},s)\} = V(\mathbf{x},s) + \mathbf{B}_i^T(s)\mathbf{y}_i(\mathbf{x}) \tag{10}$$

$$f_b\{Y(\mathbf{x},s)\} = \Phi(\mathbf{x},s) \tag{11}$$

Fig. 2 shows the input-output description of the system.

Note that the Laplace transformation acts in a twofold way on the initial-boundary value problem (1)–(4): First, it replaces the operator $D\{\cdot\}$ of time derivatives by an algebraic expression involving $Y(\mathbf{x},s)$ and the temporal frequency variable s. Second, it includes the initial values $\mathbf{y}_i(\mathbf{x})$ as an additive term into the resulting boundary value problem. The reasons why the differentiation theorem (8) has this special feature are given by the properties of the Laplace transformation: Its transformation kernel e^{st} is an eigenfunction of any linear and time-invariant system and the integration range of the Laplace integral (5) matches the temporal definition range of the PDE (1) $(0 < t < \infty)$.

Transformation with Respect to Space

In order to proceed towards a transfer function model, we have to convert the boundary value problem (10,11) into an algebraic equation. To this end, we need a transformation $\mathcal{T}\{Y(\mathbf{x},s)\}$ with respect to the spatial variables $\mathbf{x}$ with similar properties as the Laplace transformation $\mathcal{L}$. To be specific, $\mathcal{T}$ must satisfy a differentiation theorem similar to (8), where $\mathcal{T}\{L\{Y(\mathbf{x},s)\}\}$ can be expressed by $\mathcal{T}\{Y(\mathbf{x},s)\}$ and an addditive term which depends only on the known boundary values $\Phi(\mathbf{x},s)$.

Definition. The approach for this transformation is motivated by the properties of the Laplace transformation as observed above: We choose the transformation kernel $K(\mathbf{x},\beta)$ as an eigenfunction of the boundary value problem (10,11) and the spatial integration range according to the definition range of the PDE (10), which is the volume V.

$$\mathcal{T}\{Y(\mathbf{x},s)\} = \bar{Y}(\beta,s) = \iiint_V Y(\mathbf{x},s)K(\mathbf{x},\beta)\,dV = \langle Y(\mathbf{x},s),\, K(\mathbf{x},\beta)\rangle \tag{12}$$

The transformation kernel $K(\mathbf{x},\beta)$ contains the spatial variables $\mathbf{x}$ and the real valued spatial frequency variable β. In contrast to the transformation with respect to time, there is no unique form of the eigenfunction of the boundary value problem. It depends on the operator $L\{\cdot\}$ and on the shape of the domain V. The determination of the transformation kernel and the investigation of the properties of the resulting transformation uses results from the classical theory of boundary value problems [2, 4, 6, 16].

Greens Formula. The transformation (12) can also be written as an inner product $\langle Y,\, K\rangle$ between Y and the eigenfunction K. As a further simplification of the notation, we will drop the dependence on the temporal frequency variable s where appropriate. In order to formulate conditions for the determination of the eigenfunctions $K(\mathbf{x},\beta)$, we use the property that $L\{\cdot\}$ is a self-adjoint operator. This implies, that the Greens formula

$$\langle Y(\mathbf{x}),\, L\{K(\mathbf{x},\beta)\}\rangle - \langle L\{Y(\mathbf{x})\},\, K(\mathbf{x},\beta)\rangle = \iint\limits_S g_b\{K(\mathbf{x},\beta)\}\, f_b\{Y(\mathbf{x})\}\, dS - \iint\limits_S g_b\{Y(\mathbf{x})\}\, f_b\{K(\mathbf{x},\beta)\}\, dS \tag{13}$$

holds. The operator g_b is determined by the differential operator L and the boundary operator f_b.

Inspection of the Greens formula (13) shows, that it contains the desired differentiation theorem for $\mathcal{T}\{L\{Y(\mathbf{x})\}\}$, if the eigenfunctions satisfy a homogeneous boundary value problem of the same structure as (10,11)

$$-\beta^2 K(\mathbf{x},\beta) + L\{K(\mathbf{x},\beta)\} = 0\,, \qquad f_b\{K(\mathbf{x},\beta)\} = 0\,. \tag{14}$$

Sturm-Liouville Problem. Boundary value problems of this kind are also called Sturm-Liouville problems (SL problems). At first sight it seems, that we have gained little: In order to treat the boundary value problem (10,11), we now have to solve the SL problem (14). However (14) differs from (10,11) in three important aspects

- The SL problem does not contain the temporal frequency variable s.
- It is a homogeneous differential equation.
- SL problems are known to possess some useful properties: Nontrivial solutions exist only for discrete values β_μ, $\mu = 1, 2, 3\ldots$ of the frequency variable (eigenvalues) [2] and the corresponding solutions $K(\mathbf{x},\beta_\mu)$ are mutually orthogonal (eigenfunctions).

The functional transformation (12) is called a *Sturm-Liouville transformation* (SLT), when the transformation kernels $K(\mathbf{x},\beta_\mu)$ are eigenfunctions of a SL problem (14).

Orthogonality. The orthogonality of the eigenfunctions $K(\mathbf{x},\beta_\mu)$ is easily shown by applying (12) to (14) for two different eigenvalues β_μ and β_ν

$$\langle L\{K(\mathbf{x},\beta_\mu)\}, K(\mathbf{x},\beta_\nu)\rangle - \beta_\mu^2\langle K(\mathbf{x},\beta_\mu), K(\mathbf{x},\beta_\nu)\rangle \;=\; 0\,, \tag{15}$$

$$\langle L\{K(\mathbf{x},\beta_\nu)\}, K(\mathbf{x},\beta_\mu)\rangle - \beta_\nu^2\langle K(\mathbf{x},\beta_\nu), K(\mathbf{x},\beta_\mu)\rangle \;=\; 0\,. \tag{16}$$

Subtracting (16) from (15) gives

$$\langle L\{K(\mathbf{x},\beta_\mu)\}, K(\mathbf{x},\beta_\nu)\rangle - \langle L\{K(\mathbf{x},\beta_\nu)\}, K(\mathbf{x},\beta_\mu)\rangle = \left(\beta_\mu^2 - \beta_\nu^2\right)\langle K(\mathbf{x},\beta_\mu), K(\mathbf{x},\beta_\nu)\rangle \tag{17}$$

since $K(\mathbf{x},\beta_\mu)$ and $K(\mathbf{x},\beta_\nu)$ commute in the inner product. From the Greens formula (13) and the homogeneous boundary conditions in (14) follows, that the left hand side in (17) vanishes and thus $K(\mathbf{x},\beta_\mu)$ and $K(\mathbf{x},\beta_\nu)$ are orthogonal

$$\langle K(\mathbf{x},\beta_\mu), K(\mathbf{x},\beta_\nu)\rangle = 0 \qquad \mu \neq \nu\,. \tag{18}$$

Differentiation Theorem. When the eigenfunctions are determined to satisfy the SL problem (14), we can readily derive the desired differentiation theorem from the Greens formula (13). For $\langle L\{Y(\mathbf{x})\},\, K(\mathbf{x},\beta)\rangle$ follows

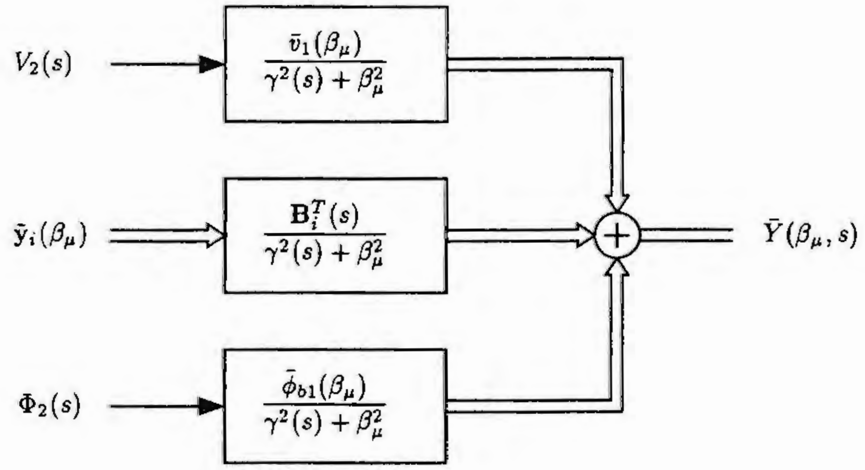
$$\langle L\{Y(\mathbf{x})\},\, K(\mathbf{x},\beta_\mu)\rangle = \beta_\mu^2\,\langle Y(\mathbf{x}),\, K(\mathbf{x},\beta_\mu)\rangle - \iint\limits_S g_b\{K(\mathbf{x},\beta_\mu)\}\, f_b\{Y(\mathbf{x})\}\, dS\,. \tag{19}$$

The surface integral depends only on the eigenfunctions and on the boundary values $\Phi(\mathbf{x})$, which are known from (4) and (11)

$$\iint\limits_S g_b\{K(\mathbf{x},\beta_\mu)\}\, f_b\{Y(\mathbf{x})\}\, dS = \iint\limits_S g_b\{K(\mathbf{x},\beta_\mu)\}\, \Phi(\mathbf{x})\, dS =: \bar{\Phi}_b(\beta_\mu)\,. \tag{20}$$

We have arrived at the differentiation theorem for the spatial transformation $\mathcal{T}$, which has the same form as the corresponding theorem for $\mathcal{L}\{D\{y(\mathbf{x},t)\}\}$ in (8)

$$\mathcal{T}\{L\{Y(\mathbf{x},s)\}\} = \langle L\{Y(\mathbf{x},s)\},\, K(\mathbf{x},\beta_\mu)\rangle = \beta_\mu^2\bar{Y}(\beta_\mu,s) - \bar{\Phi}_b(\beta_\mu,s)\,. \tag{21}$$



$V_2(s)$
$\frac{\bar{v}_1(\beta_\mu)}{\gamma^2(s)+\beta_\mu^2}$
$\bar{\mathbf{y}}_i(\beta_\mu)$
$\frac{\mathbf{B}_i^T(s)}{\gamma^2(s)+\beta_\mu^2}$
$\Phi_2(s)$
$\frac{\bar{\phi}_{b1}(\beta_\mu)}{\gamma^2(s)+\beta_\mu^2}$
+
$\bar{Y}(\beta_\mu, s)$

Figure 3: Transfer function model of a MD system in the frequency domain

This is the key result for the conversion of the boundary value problem (10,11) into an algebraic equation. It replaces the operator $L\{\cdot\}$ by an algebraic expression of the transform $\bar{Y}$ from (12) and includes the boundary value as an additive term

$$\gamma^2(s)\bar{Y}(\beta_\mu, s) + \beta_\mu^2\bar{Y}(\beta_\mu, s) = \bar{V}(\beta_\mu, s) + \mathbf{B}_i^T(s)\bar{\mathbf{y}}_i(\beta_\mu) + \bar{\Phi}_b(\beta_\mu, s) \tag{22}$$

Transfer Function Model. The final step from the algebraic equation to the transfer function model is a simple algebraic manipulation which gives an explicit expression for the transform of the output signal

$$\bar{Y}(\beta_\mu, s) = \frac{1}{\gamma^2(s)+\beta_\mu^2}\bar{V}(\beta_\mu, s) + \frac{\mathbf{B}_i^T(s)}{\gamma^2(s)+\beta_\mu^2}\bar{\mathbf{y}}_i(\beta_\mu) + \frac{\bar{\Phi}_b(\beta_\mu, s)}{\gamma^2(s)+\beta_\mu^2} \tag{23}$$

This result is rather general. The assumptions we have made are that the differentiation operator $D\{\cdot\}$ is at most of second order, which comprises most parabolic and hyperbolic PDEs of technical relevance, and that the operator $L\{\cdot\}$ is self-adjoint. There are no restrictions on the number of spatial dimensions and no special system of coordinates has been adopted. In spite of this generality, we have arrived at a compact description, which allows to separate the effects of excitation, initial and boundary values.

The transfer function description becomes even more simple under the additional assumption, that neither the sources of excititation nor the spatial distribution of the boundary values move. Then $v(\mathbf{x}, t)$ and $\phi(\mathbf{x}, t)$ are separable functions

$$v(\mathbf{x}, t) = v_1(\mathbf{x})v_2(t), \qquad \phi(\mathbf{x}, t) = \phi_1(\mathbf{x})\phi_2(t) \tag{24}$$

and the transfer function description can be decomposed into

$$\bar{Y}(\beta_\mu, s) = \frac{\bar{v}_1(\beta_\mu)}{\gamma^2(s)+\beta_\mu^2}V_2(s) + \frac{\mathbf{B}_i^T(s)}{\gamma^2(s)+\beta_\mu^2}\bar{\mathbf{y}}_i(\beta_\mu) + \frac{\bar{\phi}_{b1}(\beta_\mu)}{\gamma^2(s)+\beta_\mu^2}\Phi_2(s) \tag{25}$$

Fig. 3 shows a description of the MD system by a block diagram showing the transfer functions for the excitation, initial and boundary values, respectively. It corresponds to the input-output description of figs. 1 and 2. Those are black box systems with no visible relations between input and output signals. However, the transfer function description of fig. 3 clearly displays the internal structure of the system.

Application of Transfer Function Models

Two different directions can be taken from the transfer function description (23) or (25): One leads to a series representation of the anaytical solution of the initial-boundary value problem (1)–(4). It is based on inverse Laplace transformation and inverse SLT. The other direction leads to a discrete-time and discrete-space model suitable for computer simulation and will be shortly described in the next section.

Discrete Simulation Models

From the transfer function description follows a discrete simulation model by time and space discretization:

- Time discretization turns the system with continuous time and space coordinates into a discrete time, continuous space system (*hybrid system*). Well known analog-to-discrete transformations from onedimensional signal processing like impulse, step, ramp invariant or bilinear transformation can be applied.
- Space discretization turns the hybrid system into a discrete time, discrete space system or simply a *discrete system*. The discrete system is derived either with useful convolution properties of the SLT or by performing the inverse SLT numerically.

The resulting discrete models are well suited for computer implementation since they require only addition, multiplication and delay elements and are free of implicit loops [11, 12, 13]. The inherent stability of the physical process is preserved in the discrete model. Numerical results for the heat flow equation show that the discrete model is far more effective than standard finite difference methods (Crank-Nicolson discretization with LU-factorization) [11].

Summary

The mathematical description of a multidimensional system by transfer function models offers several advantages: It allows to separate the effects of excitation functions, initial and boundary values. It is applicable to general spatial domains and to a wide range of technically relevant boundary conditions and it does not depend on the adoption of a certain system of coordinates. Stability issues can be discussed in the framework of Laplace transfer functions. The transfer function model serves as a starting point for the derivation of discrete simulation models.

References

[1] H.S. Carslaw and J.C. Jaeger. *Conduction of Heat in Solids*. Oxford University Press, Oxford, 2. edition, 1978.

[2] R.V. Churchill. *Operational Mathematics*. McGraw-Hill, New York, 2. edition, 1958.

[3] R.M. Cotta. *Integral Transforms in Computational Heat and Fluid Flow*. CRC Press, Boca Raton, 1993.

[4] R. Courant and D. Hilbert. *Methoden der Mathematischen Physik I*. Springer-Verlag, Berlin, 1968.

[5] D.G. Duffy. *Transform Methods for Solving Partial Differential Equations*. CRC Press, Boca Raton, 1994.

[6] E. Kamke. *Differentialgleichungen I*. Akademische Verlagsgesellschaft, Geest & Portig K.-G., Leipzig, 1964.

[7] H. Krauß, R. Rabenstein, and M. Gerken. Simulation of wave propagation by multidimensional digital filters. *Simulation Practice and Theory*, 1996. to appear.

[8] M.D. Mikhailov and M.N. Özisik. *Unified Analysis and Solutions of Heat and Mass Diffusion*. John Wiley & Sons, 1984.

[9] R. Rabenstein. Simulation of linear continuous systems with distributed parameters. *Simulation Practice and Theory*, 1:93–107, 1993.

[10] R. Rabenstein. Discrete simulation of dynamical boundary value problems. In F. Breitenecker and I. Husinsky, editors, *Proc. of the EUROSIM Simulation Congress'95*, pages 177–182, Amsterdam, 1995. Elsevier.

[11] R. Rabenstein. Discrete models for multidimensional system simulation. In G. Ramponi et al., editor, *Proc. of VIII European Signal Proc. Conf. (EUSIPCO 96)*, pages 2125–2128. EURASIP, Sept. 1996.

[12] R. Rabenstein. Multidimensional system simulation with functional transformations. In *Proc. Int. Conf. Acoustics, Speech, and Signal Proc. (ICASSP 96)*. IEEE, May 1996.

[13] R. Rabenstein and H. Krauß. Discrete simulation of uniform transmission lines by multidimensional digital filters. *International Journal of Numerical Modelling, Electronic Networks, Devices, and Fields*, 9:271–294, 1996.

[14] M.D. Singh. *Systems and Control Encyclopedia*. Pergamon Press, Oxford, 1. edition, 1987.

[15] I.N. Sneddon. *The Use of Integral Transforms*. Tata McGraw-Hill, Neu Delhi, 1974.

[16] A. Tychonoff and A. Samarski. *Differentialgleichungen der mathematischen Physik*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1959.

MODELING OF STATE EVENTS

J. Plank and F. Breitenecker
Technical University Vienna, Wiedner Hauptstr. 8–10, A–1040 Wien
jplank@osiris.tuwien.ac.at

Introduction

Though state events are difficult to describe in simulation models and complicated to handle they are in a way inevitable in doing continuous system modelling and simulation. In this contribution we will present different methods for modelling state events that we worked out. Finally we will give considerations for a new concept of modelling state events.

State events become used in the modelling of continuous systems due to different reasons. The question, whether they really exist cannot be easily answered. We think that they are more or less a matter of modelling and have their origin in the difficulties of describing real systems. State events are used when

- the real system cannot be described in a single model (state space changes, different forms of motion, ...)
- there are numerical problems to be avoided (e.g. replacing of "small", quick (relative to the rest of the system) processes by State Events)

State Event Description

The description of state events is divided into two stages:

- the description of the condition that triggers the corresponding event and
- the description of the actions that have to be carried out

The condition gets usually formulated as an algebraic equation of time and state variables, that has its zeros when the event has to be triggered.

$$\Phi_i[t, x(t)] \quad i = 1, \ldots, m$$

If $i > 1$ then we could also combine these functions by mulitplying them. This product can lead to a rapidly oscillating function and therefore the algorithms will get into troubles when searching for the roots. So we should leave them separated.

Then we have to differ between level triggered and edge triggered events. This difference can be subsumed in the question whether an event condition has to be reactivated after triggering, or not. More concrete: level triggering means that when the event condition passes a threshold (zero), then the event gets permanently triggered. This makes sense for e.g. parameter changes. The correct parameter is selected depending on the actual value of the function.

On the other hand we have the edge triggered events: the only interesting point here is the time (and obviously the state) when (if) the event condition function is zero. And only in this case the event becomes triggered, e.g. when a state variable is reset. In contrast to the example above, this change of the state makes sense when it is done only once, when the condition is fulfilled.

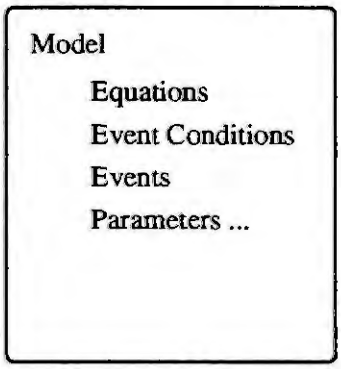
Sometimes it is also necessary to take into account the direction of the zero crossing. So it has also to be checked whether the zero crossing is from negative to positive or vice versa.

The description of the actions can differ a lot, where and how it is done in a simulation model. Different simulation languages offer different statements or other possibilities for modelling and furthermore there are differences in the applied algorithms.

In the next chapter we present different methods that will illustrate this statement.

Methods of Modelling State Events

The simplest method of modelling state events is to put the conditions and the events in the model description without using special statements.



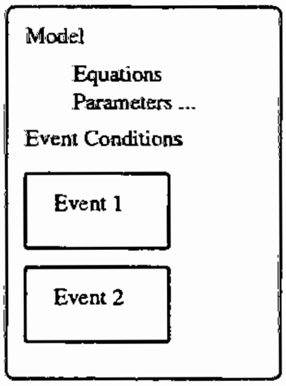
From the methodological point of view this seems to be the most "natural" method. We start from one system and build up its representation in one single model. But in practice we will come across several troubles: When trying to model structural variable systems we then would have to start the description with at least two different sets of differential equations. During simulation we would have to switch from one set to another when necessary and to force the others to be "silent".

On the other side it is almost not possible to implement algorithms for handling the events. Only the usual error control can be applied.

In addition, the model is not really easy to read and to maintain. Furthermore it is only valid for one special system. Adapting it for other systems would be a very costly task.

The next method is to use structural elements, like discrete sections. They are mostly coupled with a condition statement so that the event can be handled properly by the numerical algorithms.

The statement for the conditions of the events here can be synchronised with the numerical algorithms. The description of the actions of the events is made in separated structural elements. So this method is good for dealing with parameter changes or reinitialising state variables. For structural variable systems and for component exchanges we will run into the same troubles as above.



Due to the description of the actions in separate structural elements we gain easier maintainability. When we want to update or exchange the actions in our model we only have to exchange the discrete sections.

Languages that apply this method are for example ACSL and MOSIS.

Another method is to use the model sequentially; when the condition is fulfilled, the model becomes reinitialised and is started again.

So we make our experiments with concatenating different runs. In between of the runs the model gets reinitialised and is then started again. Therefore these sequential experiments sum up to the big experiment.

The event condition is here reduced to a single statement, the conditional termination. The action is then carried out at runtime level. The actions are therefore also separated from the model.

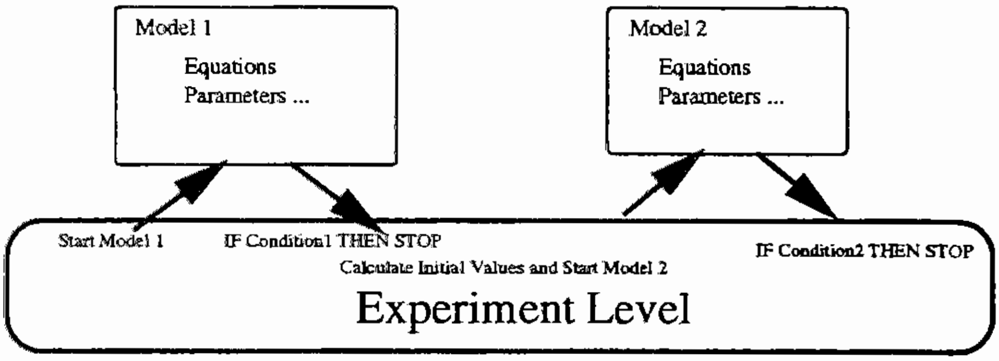
For structural variable systems this method can be easily extended. In this case we do not use only one model but more models. When a simulation run is stopped the runtime level decides which model to take next. So the parameters and the initial values are calculated as before but additionally the model has to be selected.

As an example we will discuss a simplified pendulum. The corresponding equations will be sufficiently known. When the pendulum reaches - when looping - a point where the centrifugal force does not tighten the string any more the motion gets the form of the free fall.

This example consists now of two different model description. The first model describes the swinging motion of it and the second the free fall. The first condition for swapping between the two models is the centrifugal force. When it is not able to tighten the string we swap to the fall model. The second condition is the length of the string: When the distance between the mass and the point of rotation gets the value of the length of the string, we come back to the swinging motion.

So when making an experiment we have to start the right model, depending on the initial values. When the simulation is stopped, the new initial values are calculated depending on the final values of the previous run.

At last we want to present a method that serves as a basis for the considerations for a new concept at the end of this contribution. The event can also be separated from the model and put on the experiment level. If the condition becomes true, the actual simulation run is stopped and the experiment starts a different model after calculating the initial values. This method requires a very capable experiment level that supervises the experiment and acts if a condition is fulfilled. This method is especially fit for structural changes.



The advantage of this method is that the condition and the description of the event is separated completely from the model description. Therefore it supports the reusability of the model.

Considerations for a new Concept

The considerations for a new concept base on the Model Interconnecton Concept developed by Schuster [2]. So we use generalised "models" to build up the "model", the system description. Here "models" are functions, parameter, constants, ... that are linked together dynamically at runtime dependent on the actual state of the system to the "big" model that represents the simulated system.

The events could be transferred to a "meta model" that consists only of the state event condition descriptions. The event actions are then the creating or removing of links between these models. The main advantage of this concept is the modularity that offers a modular concept, easy maintainability and reusability of model components and also possibilities for parallelisation Therefore the new concept covers:

- The simulation of parallel model blocks
- The simulation of sequential models
- A combination of sequential and parallel models or model blocks
- Describing the exchanging of numerical algorithms as state events

References

1. Breitenecker F. and Solar, D., Models, Methods, Experiments - Modern aspect of simulation languages. In: Proc. 2nd European Simulation Conference, Antwerpen, 1986, SCS, San Diego, 1986, 195-199

2. Schuster, G. - Definition and Implementation of a Model Interconnection Concept in Continuous Simulation, Dissertation, TU-Wien, 1994

A unified theory for automation systems

Dipl.-Ing. A. Mircescu Prof. Dr.-Ing. E. Schnieder

Matter and *energy* are the *substances* of the physical world. *space* and *time* their *existence forms* [6]. Automation systems additional contain *information* as a third substance with *causality* as its existence form. An exact and integrated description of the casual connection in automation systems is won in this paper by transferring and generalizing physical and mathematical reflections of the *special theory of relativity* and of *quantum theory*. Description of automation systems in *generalized spacetime* by the *generalized continuity equation* opens the possibility of computing *balance equations* for the whole automation system.

1 Introduction

The experience won in decades in automation theory points out the existence of the substances matter. energy and information with their existence forms space, time and causality (causal dependences). This three substances and their interactions determine the spatial, temporal and causal behavior of the automation system. Only an integrated description of all substances in their existence forms guarantees a correct and exact modelling and prediction of system behavior.

The general spatial and temporal structural and behavioral description of the substances matter and energy is realized in physics by the theory of relativity and by quantum theory. These two theories are the most general physical theories because they do not describe only physical aspects of the system (like the electromagnetic theory or thermodynamics) but the general behavior of the substances in space and time.

For describing the third substance of the automation system, information, in its existence form, causality, Carl Adam Petri developed 1962 in his doctorate the class of *causal nets* which years later were named *Petri nets*. Petri noticed in his doctorate that the causal description of automation systems must be embedded in the spatial and temporal connections which are determided by physics.

Petri nets have the excellent property of permitting a geometric derivation of causality. Thus they define a measure for causality like clocks for time. The structure of automation engineering is described and quantized by spatial measures in space (measured in meters), temporal measures (clocks) in time (measured in seconds) and causal measures in causality (measured in petri).

Like for spatial movements velocity of changes between causal states is limited by the speed of light. This similar behavior of causal and space achses allow the transfer of the spacetime geometry developed by Albert Einstein and Hermann Minkowski in the special theory of relativity to the five dimensional *generalized spacetime* of automation engineering.

Behavior of automation systems is determined of the possible system states and of the allowed transitions between this states. Petri nets encode both informations in a simple graphical and a precise mathematical form. Selection rules for the automation system can be directly obtainded analysing the Petri nets.

In quantum theory the eigenstates of the microsystems and the transitions between eigenstates are described by *observables* and *matrix transition elements*. These reflections introduced and developped by Werner Heisenberg, Erwin Schrödinger, Max Planck, Niels Bohr, Paul Dirac and others can be transferred to causal transitions by creating operators using the encoded information of the Petri net representation of automation systems.

Like in the theory of relativity and in quantum theory a *generalized continuity equation* of automation engineering can be derived which on her part leads to *balance equations* in automation systems.

2 Geometric derivation of automation engineering

The key for deriving a geometric derivation of automation engineering is obtaining geometric structures of causality and unifing the geometric structures of space, time and causality to a generalized spacetime which describes the world of automation engineering. The first step will be realized by Petris causal nets, the second by transferring Einsteins considerations in the special theory of relativity.

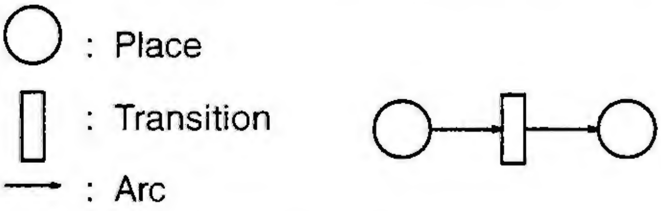


Figure 1: Basic structure of Petri nets

Petri nets describe the causal dependences in automation systems by the use of *places*, *arcs* and *transitions* [4] as shown in picture 1. The places represent possible states of the system whereas the arcs encode the selection rules for the changes of system states. The transitions cause then the changes (transitions) between different states. The elementary unit of the Petri net consists of two places connected by arcs and one transition, see picture 1. These two states are direct causal neighbors because the second state can be directly reached beginning from state one. **This minimal causal distance of two states is choosen as causal measure, is afflicted with the dimension Petri and is set per definition to** $1Petri = 1P$.

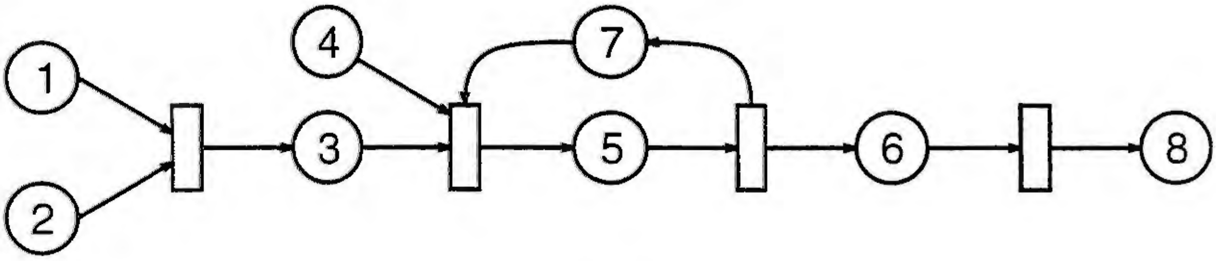


Figure 2: Causal structure

Picture 2 shows how this defined causal measure can be applied to a random causal structure. For example the causal distance between state 1 and state 3 is equal to 1 P, between state 2 and 3 1 P. Between state 5 and 8 the distance is 2 P because state 8 cannot be reached directly from state 5 but only passing state 6. Table 1 summarizes all causal distances of the reguarded causal structure.

The rows represent the starting causal states, the columns represent the target causal states. Row i and column j indicate the causal distance between state i and state j. If $i = j$, the distance is 0 P; if state j cannot be reached beginning from state i the causal distance is ∞ like between state $i = 8$ and state $j = 6$.

The definition of causal measure completes the necessary measurements for a geometric derivation of the sturcture of automation engineering. The causal measures allow specification of causal ordering relations. Setting a time reference mark and measuring time referring to this mark sequences of events and temporal distances between them can be estimated. Analogous causal sequences and causal distances between states can be obtainded by setting a causal reference mark and applicating the causal measures. Spatial, temporal and causal attributes of processes can be quantized by this method.

	1	2	3	4	5	6	7	8
1	0	∞	1	∞	2	3	3	4
2	∞	0	1	∞	2	3	3	4
3	∞	∞	0	∞	1	2	2	3
4	∞	∞	∞	0	1	2	2	3
5	∞	∞	∞	∞	0	1	1	2
6	∞	∞	∞	∞	∞	0	∞	1
7	∞	∞	∞	∞	1	2	0	3
8	∞	∞	∞	∞	∞	∞	∞	0

Table 1: Causal distances measured in *Petri*

Processes in automation systems include spatial, temporal and causal properties. For example each translation in space is coupled with a translation in time. The junction of the spatial and temporal coordinates is realized by the definition of velocity $v = x/t$. $x = vt$ is then the space which is covered after the time t by translation with the veolcity v. Thus time t can be mapped to a spatial value x. An integrated computation of the physical world in spacetime is now possible.

For the integration of causality in an unified geometric derivation a mapping for the causal axsis to the spatial or temporal axsis has to be found. The idea for the solution is to take the time for the translation between two states into consideration. This time depends on the distance in spacetime and on causal dependences between the system states. As a charakteristic time t_k specificates the time in seconds which is necessary for a causal translation of one *Petri*. t_k is a causal normalized time:

$$t_k = \frac{t}{1P}. \tag{1}$$

A multiplication of t_k with the causal distance measured in *Petri* leads to the time in seconds which is necessary for the causal transition. By this method the causal axis can be mapped to the temporal axis or with $t = x/c$ to one spatial axis. Every system state (eigenstate) can be now charakterized with an 5-toupel $(x_0, x_1, x_2, x_3, x_4)$ where (x_0) describes the temporal, (x_1, x_2, x_3) the three spatial and (x_4) the causal position in a *five dimensional generalized spacetime.*

Like in physical spacetime no spatial translations can take place with a velocity greater than c (light velocity in vakuum). This law is valid for causal translations too because of their embedding in the physical world (t_k depends on the distance in spacetime). *The causal axis acts from this point of view like an additionary spatial axis!*

In the theory of special relativity the fact of limited propagation velocity is considered by defining spacetime and light cones for the attainability of events [6]. The geometry of spacetime defined by Albert Einstein and Hermann Minkowski is a four dimensional vector mannifold with one negative (temporal) and three positive (spatial) dimensions. This geometry is described by the quadratic form:

$$Q(\vec{x}) = -x_0^2 + x_1^2 + x_2^2 + x_3^2. \tag{2}$$

In the five dimensional generalized spacetime of automation engineering this geometry can be generalized to [5], [3]:

$$Q(\vec{x}) = -x_0^2 + x_1^2 + x_2^2 + x_3^2 + x_4^2 \tag{3}$$

with x_4 representing the causal axis which behaves like an additional spatial axis. Only events within this five dimensional cone can be attained. Equation [6]

$$\vec{z} = \xi\vec{e_x} + \vec{y} + \kappa\vec{e_k} \tag{4}$$

shows a split representation of the orthogonal temporal $(\xi \vec{e_x})$, spatial $(\vec{y})$ and causal $(\kappa \vec{e_k})$ coordinates. The quadratic form is then:

$$(\vec{\Xi} \cdot \vec{\Xi}) = -\xi^2 + |\vec{y}|^2 + \kappa^2. \tag{5}$$

This equation is the generalized Pythagoras formula for the word of automation engineering. The splitted representation leads twards another result: In automation engineering the 4 axioms structure. decomposition. causality and temporality define automation systems. The spatial coordinates of the generalized Pythagoras formula define the spatial structure of the system (first axiom). The time coordinate defines the temporality (fourth axiom) and the causal coordinate the causality (third axiom) of the automation system. Transformation of coordinates and reference frames are invariant in the quadratic form. This describes the decomposition axiom. The 4 system axioms are no longer axioms but properties of generalized spacetime. The geometry of generalized spacetime can be described with a *metric tensor* g

$$g = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

like in special theory of relativity.

3 Transitions and selection rules in automation systems

Generalized spacetime describes the structure of automation engineering. Dynamic behavior of the substances matter and energy is specified in physics by quantum theory. The spatial eigenstates of the substances are encoded in the wave function Ψ, transitions between different eigenstates including selection rules are specified by matrix transition elements $\langle\varphi|M|\psi\rangle$. $\langle\varphi|$ is a vector of dimension $(1, n)$ and is representing the starting state. $|\psi\rangle$ has the dimension $(n, 1)$ and represents the target state. The quadratic (n, n) matrix M encodes the selection rules. $\langle\varphi|M|\psi\rangle$ is equal to the transition probability between state $\langle\varphi|$ and $|\psi\rangle$ [1].

Geometric derivation of causality defines ordering relation between causal states. The state representation and ordering relations of causality one one hand and the existence of selection rules encoded in Petri nets on the other hand allows thus a transfer of quantum theoretical specification to automation engineering. Next a matrix transition element for automation systems has to derieved from the Petri net representation of the system.

Following definition is made for M: **M is a quadratic matrix which contains as many rows and columns as the number of system eigenstates. If the causal distance between state i and state j is one Petri the element $M(i, j) = 1$ else $M(i, j) = 0$.** For the causal structure of picture 2 M is:

$$M = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{7}$$

Let $\langle\varphi|$ be the start and $|\psi\rangle$ the target state then $\langle\varphi|M|\psi\rangle = 1$ if the causal distance is one Petri (allowed transition) else $\langle\varphi|M|\psi\rangle = 0$ [3].

The product $\langle\varphi_i|M|\varphi_j\rangle \cdot \langle\varphi_j|M|\varphi_k\rangle$ specifies the transition sequence from state i to state j and then to state k. If the product is equal to one the transition sequence is allowed else not [2].

3.1 Generalized continuity equation

In four dimensional spacetime the continuity equation

$$\frac{\partial\varrho}{\partial t} + \mathrm{div}J = 0 \tag{8}$$

can be derieved from theory of relativity (quadratic form is invariant to translations). ϱ is the density of substace and J the density of substance current. In generalized spacetime an invariant quadratic form exists too: therefore the continuity equation must have an additional term which describes the causal transitions. The quantum theoretical specification of causal transitions by matrix transition elements allow the estimation of this term. The five dimensional generalized continuity equation can be written as:

$$\frac{\partial\varrho}{\partial t} + \mathrm{div}J + \sum_{i=1}^{n}\left(\frac{\langle\varphi_i|M|\psi_i\rangle}{dV\,dt}\right) = 0. \tag{9}$$

The third term indicates all causal transitions of the automation system in space dV and in time dt.

4 Balance equations

Behavior of the three substances matter, energy and information is specified in generalized spacetime by the generalized continuity equation using causal matrix transition elements. For the realization of transitions in generalized spacetime technical ressources are required. The ressources can be divided into three classes: material, energy and information ressources. Each class of ressource can reguarding to generalized Pythagoras formula again be divided into three functional basic elements: causal-processing (P), spatial-communication (C) and temporal-memory (M) functions.

In an information system P functions are represented by processors, C functions by communication systems and M functions by the memory. Energy systems own turbines which act like P functions and rural subscriber lines and accumulators representing C or M functions respectively. Production machines are the P functions of a production system, transportation systems their C functions. The M functions are specified by the storage units.

In every automation or information system translations in generalized spacetime consider the generalized continuity equation. In many applications the temporal behavior of the system is of great interest, for example in applications with real time requirements. For this cases the continuity equation can be integrated in the three space coordinates. The result is:

$$\frac{\Delta Q}{\Delta t} + I + \sum_{i=1}^{n}\frac{\langle\varphi_i|M|\psi_i\rangle}{\Delta t} = \mathrm{const.} \tag{10}$$

This equation is a conservation law for the substances in generalized spacetime. In an information system the first term specifies the number of Bits ΔQ leaving the memory in the interval Δt (temporal translations); I is the current measured in Bit/s circulating in the communication system (spatial translations per time unit). The third therm describes the number of causal translations per time executed by the processor (unit: operations per time unit).

In an automation system consisting of some subsystems the conservation law can be applied to each subsystem. The result is a system of balance equations specifiing the occupation of ressources. Evaluating the equations the scheduler realizes the task distribution.

5 Summary and conclusions

The world of automation engineering is the five dimensional generalized spacetime consisting of one temporal, three spatial and one causal dimension. Petri nets achieve the geometric derivation of causality. Velocity of causal translations is limited by the speed of light like velocity for spatial translations. This condition allows the transfer of specifications in the special theory of relativity building the generalized spacetime.

System behavior of automation systems consists of causal states. Like in quantum theory translations between states are limited by selection rules which are encoded in Petri nets for automation systems. This quality leads to the possibility of transferring the matrix transition elements of quantum theory to causality and specifiing a generalized continuity equation.

Interpretation of the generalized continuity equation in generalized spacetime sets up a system of balance equations. These equations can be evaluated by a scheduling system for the distribution of the tasks.

References

[1] O. Hittmair. *Lehrbuch der Quantentheorie.* Verlag Karl Thiemig, München, 1972.

[2] B. Huppert. *Angewandte Lineare Algebra.* de Gruyter Verlag, Berlin, New York, 1990.

[3] A. Mircescu. *Eine allgemeine Theorie verteilter Automatisierungssysteme.* Interner Bericht, TU Braunschweig, Institut für Regelungs- und Automatisierungstechnik, 1996.

[4] E. Schnieder. *Petrinetze in der Automatisierungstechnik.* Oldenbourg Verlag, München, Wien, 1992.

[5] D. Werner. *Funktionalanalysis.* Springer Verlag, 1995.

[6] H. Weyl. *Raum-Zeit-Materie.* Springer Verlag, achte Auflage, 1993.

Address
Prof. Dr.-Ing. Eckehard Schnieder
Dipl.-Ing. Alexander Mircescu
Technische Universität Braunschweig
Institut für Regelungs- und Automatisierungstechnik
Langer Kamp 8
38106 Braunschweig
Germany
Tel.: ++49/(0)531/391-7667
Fax.: ++49/(0)531/391-5197
email: mircescu@ifra.ing.tu-bs.de
www: http://www.ifra.ing.tu-bs.de/

CONTROLLED QUASICONTINUOUS ORBITS

V. Gontar, M. Gutman

The International Group for Scientific and Technological Chaos Studies (IGCS) Ben-Gurion
University of the Negev, Beer-Sheva
P.O. Box 653, Beer-Sheva, 84105, Israel

A technique for obtaining iterations of piecewise continuous one-dimensional maps in the special form of quasicontinuous (QC) orbits has been proposed [1]. We use the term "quasicontinuous" to indicate that long segments of the orbits resemble numerical solutions of differential equations. As has been shown, a variety of chaotic and complex periodic QC-orbits may be generated by a map for which discontinuity point x_d coincides with the selected fixed (or periodic) point x_f (or x^*_i, i=1,2...). Here we consider this approach for the case where the right continuous branch of the map is a horizontal line (HL-map). For $x_d = x^*_i$, the monotonic variation of the location of the HL branch results in higher-level inverse and direct cascades of nonconventional quasicontinuous bifurcations (QCB's) which arithmetically change the period of successive orbits [2]. Fig.1 gives us an example of a 4-level cascade of QCB's which is generated by the following logistic-like HL-map:

$$x_{n+1} = f(x_n) \equiv \begin{cases} ax_n(1-x_n) & \text{if } x_n < x_d \\ \varepsilon & \text{if } x_n \geq x_d \end{cases} \qquad (1)$$

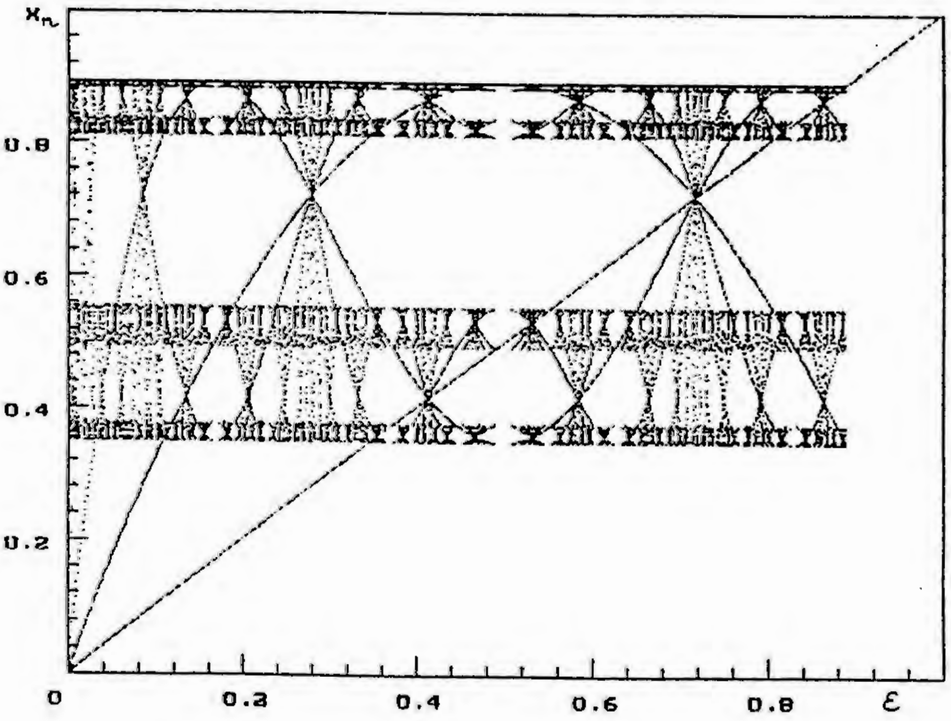


Fig.1 Bifurcation diagram of the map (1), $x_d = x^*_4$

The hierarchical organization of the levels is clearly visible: the 4-level cascades are embedded in corresponding 3-level cascades, each of which is embedded in 2-level cascades, and so forth. Moreover, every QCB-point in a cascade of m-level accumulates QCB-points belonging to the cascade of next, m+1-level (m = 1, 2, ...). This fractal structure leads to the appearance of complex orbits with long laminar segments (QC-orbits of different orders) in the vicinity of the QCB-points. (QC-orbit of i-th ofder contains those laminar segments where every 2^{i-1}-*th* periodic point lies on a smooth monotonic curve.) All such orbits are periodic. Thus, when an HL-branch is located near a selected QCB-point, we obtain a periodic QC-orbit of the desired form, as is shown in Fig. 2. These controlled QC-orbits, whose characteristics vary over a wide range, may be used to simulate real oscillating systems, e.g. chemical systems [3]. Further, HL-maps allow new types of orbits and new bifurcation phenomena to be studied.

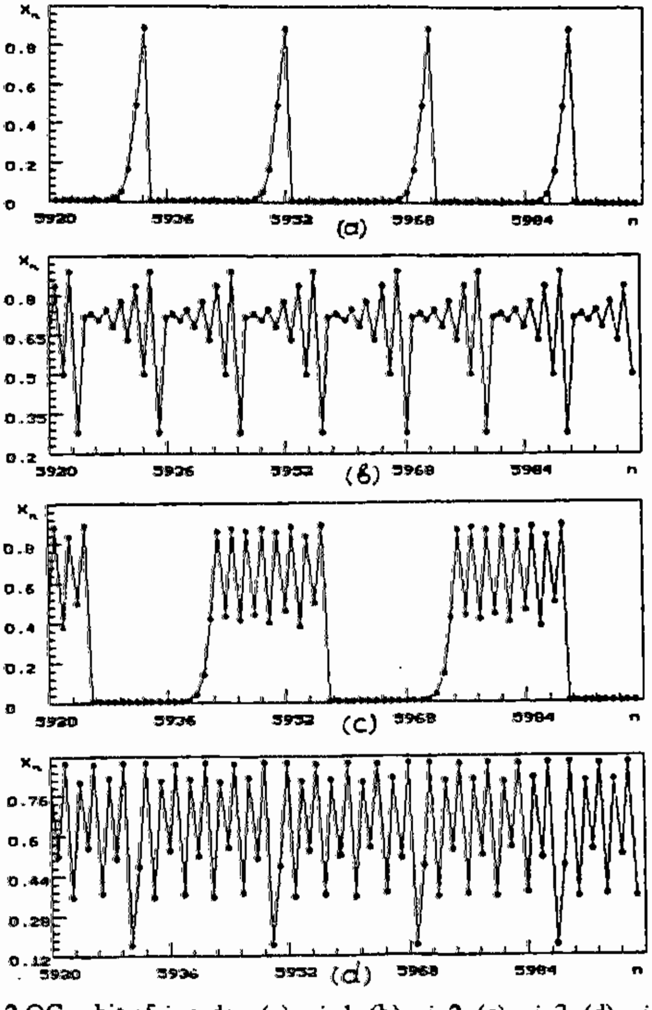


Fig.2 QC-orbit of i -order: (a) - i=1, (b) - i=2, (c) - i=3, (d) - i=4

References

1. Gontar, V. and Gutman, M. (1994) "Controlled bifurcation transitions for simulations of chaotic and complex periodic oscillations" Proc. 1. MATHMOD Vienna, LP volume, p. 946.
2. Gutman, M. and Gontar, V. (1995) "Route to chaos via inverse cascade of continuous bifurcations" Int. J. Bifurcation & Chaos, vol. 5, n. 1, pp. 123-132.
3. Gontar, V. (1993) Chaos in Chemisry and Biochemistry. World Scientific, London, pp. 215-232.

Identification of a Nonlinear Compressor Model

C. J. Rivera and J. V. R. Prasad
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150

Abstract. In this article, a technique for the identification of a nonlinear compressor performance model exhibiting bifurcation phenomena is presented. The approach consists of employing the modal form of the compressor flow equations along with derived linear and nonlinear stability results. When the identified steady-state model is used in the system, bifurcation analysis tools predict stable solutions in accord with experimental steady-state data.

Introduction

In a number of applications, the time evolution of the state in dynamic models depends strongly on variable parameters and nonlinear functions of the state. Such nonlinear models often exhibit bifurcations, or changes in the qualitative structure of the flow as the parameters are varied. The nonlinear analysis of such models provides a wider picture of the behavior of the steady-state solutions in terms of the parameters. This picture can only be complete if the nonlinear functions describing the steady-state performance are known. However, the presence of bifurcations introduces difficulties in the identification of steady-state performance models from simple steady-state experiments. Over a certain parameter range, the stable steady states are those from a bifurcated solution, which in turn may depend on the unstable and unmeasurable locus of steady states. Given the underlying physical and system limitations, the global nonlinear dynamic behavior can provide insights which could allow the mapping of the characteristic performance model of the system.

An approach to the identification process consists of characterizing the qualitative nonlinear behavior of the model as a function of the features of the steady-state performance model. Then, the qualitative nonlinear behavior is observed in the physical system. Finally, one combines the results to obtain the nonlinear performance model which allows quantitative agreement between the analytical and experimental observations.

In this article, the methodology described above is employed to determine the steady-state performance model, or pumping characteristic of a compressor in the flow range where stable operation is not attainable. The compressor exhibits, in addition to uniform flow operation, the less desirable mode of operation attributed to rotating stall. Analysis results available for a model for rotating stall in compression systems are utilized to formulate a nonlinear programming problem. The solution of this problem provides a steady-state performance model from available compressor data.

In the following section, the generic model and basic stability results are presented. The last section shows how the model equations and stability information are used to formulate the steady-state model optimization problem. The computed performance model is then tested to demonstrate the agreement between analytical and experimental results.

Compression System Modeling

Rotating stall is a type of aerodynamic instability attributable to flow separation in the compressor blading. The desirable mode of operation of a compressor involves the generation of a desired pressure rise with a uniform flow through the device. However, when rotating stall is present, the flow is no longer uniform, a rotating flow blockage is set up around the annulus and the performance decreases, often considerably.

Several models for analyzing rotating stall have been proposed, all of which possess similar behaviors. In this respect, the same problems arise at the time of identifying the performance model out of the dynamic model. Among those developed for low-speed compressors, the Moore–Greitzer model in [1] will be utilized in this work to demonstrate the qualitative behavior of a realistic model for rotating stall. The model involves the description of the rotating disturbance through its spatial harmonic content. After a reduction through the Galerkin method one obtains a system of equations for the average flow, Φ, and pressure rise, Ψ, through the compressor, and the amplitude, A, of the first harmonic of the rotating flow disturbance. The resulting system of equations is

$$\begin{aligned}\frac{dA}{d\tau} &= \frac{1}{k_A}\left(\sum_{n=1,n\ odd}^{N_p} \frac{\psi_c^{(n)}(\Phi)}{n!\, r_n} A^n\right) \\ \frac{d\Phi}{d\tau} &= \frac{1}{L_c}\left[\left(\sum_{n=1,n\ even}^{N_p} \frac{\psi_c^{(n)}(\Phi)}{n!\, s_n} A^n\right) - \Psi\right] \\ \frac{d\Psi}{d\tau} &= \frac{1}{4B^2 L_c}\left(\Phi - K_T\sqrt{\Psi}\right)\end{aligned} \tag{1}$$

The parameters k_A, μ, m, L_c and B represent flow inertia and capacitance in the model. For a fixed compressor speed and system geometry, these parameters are constant. The parameter K_T represents the area of the throttle valve. K_T is considered as the free parameter in the model, since its value sets the operating point of the system. The function $\psi_c(\Phi) = \psi_c^{(0)}(\Phi)$, and its derivatives $\psi_c^{(n)}(\Phi)$ up to order N_p characterize the steady pressure rise in the absence of rotating stall, *i.e*, when $A = 0$. Thus, it is often called the compressor axisymmetric performance. The numbers r_n and s_n are rational numbers arising in the expansion of the N_p derivatives.

As part of the research efforts to develop rotating stall controllers for the axial compressor rig at the Georgia Tech LICCHUS [1] [2,3,4,5], a steady-state and dynamic mode validation of this model was undertaken. The analysis presented in [3] plays an important role in the steady-state model identification developed in this article. Here, we refer to steady-states as the limit sets of system trajectories from all possible initial conditions in the phase space. In this discussion, attention is focused on equilibrium or time-invariant solutions as these are the relevant steady-state solutions of the model in Equation 1 in the parameter range of interest.

According to Equation 1 the case $A = 0$ corresponds to a steady-state solution of the model. This situation characterizes the uniform, or axisymmetric flow solutions of the model, in which rotating stall is absent. A linear analysis of the model equations reveals that the axisymmetric steady-state solution, defined by

$$\Psi = \psi_c(\Phi)\ ,\quad \Phi = K_T\sqrt{\Psi}\ ,\quad A = 0\ \ ,$$

is locally stable if the slope of the map, $\psi_c^{(1)}(\Phi) = \psi_c'(\Phi)$ is negative. At axisymmetric equilibrium points for which $\psi_c'(\Phi) > 0$, the local solution is unstable. The stability of points for which $\psi_c'(\Phi)$ is zero cannot be determined from the linearization, since the eigenvalue of the Jacobian matrix of the system in 1 corresponding to the amplitude equation is zero. In particular, the local maximum of $\psi_c(\Phi)$ is called the stall inception point in [3]. As explained below, in the neighborhood of this point, a variety of steady-state solutions is possible. As the throttle parameter varies around the value

$$K_{T_c} = \Phi_p/\sqrt{\psi_c(\Phi_p)}\ ,\quad \{\Phi_p :\ \psi_c'(\Phi_p) = 0\ , \psi_c''(\Phi_p) < 0\}$$

a nonzero initial condition simulation of the model out of the plane $A = 0$ may settle into the axisymmetric solution, or the system may enter fully developed stall, in which case, the stall amplitude A is nonzero. In addition, the model may exhibit hysteresis with respect to the onset and cessation of rotating stall, which is a characteristic behavior of highly loaded axial flow compressors. Such steady-state multiplicity indicates that bifurcations have a strong effect on the long-term behavior of the system. Therefore, results in bifurcation theory [6] find direct application in the analysis of the model.

The analysis of local bifurcations of nonlinear equations is accomplished by reducing the model vector field to a single scalar equation–the bifurcation equation–and determining singularity conditions by computing the derivatives of this equation. In general this approach is difficult to carry out and numerical methods are applied to the analysis [7]. The method employed for the results shown in this article is described by Doedel in [8].

Since the linearization of the model at the critical point is singular, the stability of the stall inception point must be determined from a nonlinear/bifurcation analysis. In [3], a relation for the stability of the peak point based on the center manifold theorem and the reduction principle is developed. The

[1] Acronym for Laboratory for Identification and Control of Complex, Highly Uncertain Systems

condition states that the stability of the stall inception point is determined by the sign definiteness of $q(\Phi_p)$, where

$$q(\Phi_p) = \psi_c'''(\Phi_p) + \frac{\Phi_p}{\psi_c(\Phi_p)}[\psi_c''(\Phi_p)]^2 \ , \quad \{\Phi_p : \psi_c'(\Phi_p) = 0 \ , \psi_c''(\Phi_p) < 0\} \tag{2}$$

If $q(\Phi_p) < 0$, then the stall inception point is a stable steady-state solution. In that case, the bifurcating solution emerging from the peak, which corresponds to rotating stall, is also stable in the vicinity of the critical point. If $q(\Phi_p) > 0$, the stall inception point is an unstable equilibrium solution, in which case the bifurcating solutions are also unstable. A third stable solution corresponding to rotating stall is predicted in this case. Thus hysteresis is present whenever $q(\Phi_p)$ is sufficiently positive. The important result is that the stability of the stall inception point can be determined from the derivatives of the compressor characteristic at that point.

Performance Model Validation

To follow the nonlinear viewpoint, it is necessary to explore the long-term behavior measured in experiment. For the facility at Georgia Tech, extensive tests [4] reveal the typical behaviors mentioned above. A rising performance at large throttle values away from stalled operation is measured. At lower values, hysteresis with respect to the onset and cessation of stall is observed. At yet lower throttle settings, rotating stall is the only mode of operation in steady-state.

Employing the measured steady-state experimental data, one can formulate a mathematical programming problem for the determination of the axisymmetric steady-state performance model of the system. The model steady-state relations, which are linear in the coefficients of the polynomial, define the function to be minimized. In this regard, we employ only the amplitude and mean flow equations in the system of Equation 1. The third equation does not depend explicitly on the unknown map and is omitted.

The steady-state relations for flow and rotating stall amplitude are cast in terms of the unknown coefficients $\{c_k\}_{k=0}^{N_p}$ of the polynomial $\psi_c(\Phi) = \sum_{k=0}^{N_p} c_k \Phi^k$, as

$$\begin{aligned} \frac{dA}{dt} &= \frac{1}{k_A}\sum_{k=0}^{N_p} c_k f_k(A, \Phi) = 0 \\ \frac{d\Phi}{dt} &= \frac{1}{L_c}\left[\sum_{k=0}^{N_p} c_k g_k(A, \Phi) - \Psi\right] = 0 \end{aligned} \tag{3}$$

where N_p is the order of the polynomial and the functions f_k and g_k arise in the collection process. The experimental data is inserted into Equation 3 as follows. For N_a data points at which stall is not present (axisymmetric equilibria), and N_s stalled data points, the algebraic equations in steady state can be put in matrix form as $Ac = b$, where

$$A = \left[\begin{array}{ccccc} (1 & g_1 & g_2 & \cdots & g_{N_p})_1 \\ \vdots & & & & \vdots \\ (1 & g_1 & g_2 & \cdots & g_{N_p})_{N_a} \\ \hline (1 & g_1 & g_2 & \cdots & g_{N_p})_1 \\ (0 & f_1 & f_2 & \cdots & f_{N_p})_1 \\ \vdots & & & & \vdots \\ (1 & g_1 & g_2 & \cdots & g_{N_p})_{N_s} \\ (0 & f_1 & f_2 & \cdots & f_{N_p})_{N_s} \end{array}\right] \ ; \ b = \left[\begin{array}{c} (\Psi)_1 \\ \vdots \\ (\Psi)_{N_a} \\ \hline (\Psi)_1 \\ 0 \\ \vdots \\ (\Psi)_{N_s} \\ 0 \end{array}\right] \ . \tag{4}$$

A is a general rectangular matrix whose rows are made up of the functions f_i and g_i evaluated at each data point. The load vector b contains the pressure for the flow equation entries and zeros for the amplitude equation. The vector c holds the unknown polynomial coefficients. Since the system is not necessarily square, the coefficients must be found in an optimal sense. The traditional objective function in such case is

$$J_0(c) = ||Ac - b||^2 = c^T A^T Ac - 2b^T Ac + b^T b \quad . \tag{5}$$

The unconstrained optimal solution, given by $c = A^+ b$, where A^+ is the pseudoinverse of A, may yield a set of coefficients which could fail to predict the global properties of the steady state solutions in terms of stability and bifurcation results. As pointed out above, the model should give only one set of equilibria for the $A = 0$ branch, in a flow region away from hysteresis where the axisymmetric map has negative slope. The data for this case is contained in the partitions corresponding to zero-stall data of A and b. In addition, the model should predict two steady-state solutions when the slope of the axisymmetric map is positive. One of the solutions is the unstable axisymmetric performance model itself, which is the unknown of the problem. The other is the stable stalled solution, which has been incorporated as experimental data in A and b in the stalled-data partitions. Furthermore, if the data demonstrates the presence of hysteresis, then the resulting peak stability criterion of Equation 2 should be relevant to the optimization process. Hence, in order to obtain a reasonable set of coefficients, the stability information should be supplied as constraints in the search of an optimal set of coefficients which minimizes $J_0(c)$.

The general programming problem for the model consists of minimizing $J_0(c)$ subject to constraints which depend on the stability of the equilibrium solutions in different flow regimes. Since the majority of the constraints depends on the slope of the map,

$$\psi_c'(\Phi) = \sum_{k=1}^{N_p} \frac{k!\; c_k \Phi^{k-1}}{(k-1)!} \quad ,$$

the problem is almost a quadratic programming problem. In effect, it is known that at axisymmetric values of Φ and Ψ, the slope should be negative. When the data shows that $A \neq 0$, the slope must be positive, except in the hysteresis region. Therefore, for N_a axisymmetric data points and N_l stalled data points which do not include the hysteresis region, the coefficients satisfy the linear inequality

$$Ec < 0 \; ; \quad E = \begin{bmatrix} E_a \\ --- \\ -E_l \end{bmatrix} \tag{6}$$

The nonlinear hysteresis information can be incorporated into the steady-state model by identifying or estimating a peak pressure value from the data. Given that the peak point is unstable when hysteresis is present, the recourse is to make an estimate of the peak point. If the peak point is thus selected, one can form a nonlinear inequality based on Equation 2 with a selected Φ_p and $\psi_c(\Phi_p)$. In this case, the model consisting of Equations 5, 6, and the nonlinear constraint, represents a nonlinear programming problem in the unknown coefficients.

Efficient numerical tools for the solution of nonlinear programming problems exist. In particular, the DONLP code by Spelluci [9] serves this purpose quite well. Employing the experimental data from the compressor, setting $\Phi_p = 0.35$, $\psi_c(\Phi_p) = 0.173$, and placing a bound on $q(\Phi_p) \geq 10$, one obtains, for $N_p = 5$,

$$\psi_c(\Phi) = 73.34\Phi^5 - 127.787\Phi^4 + 68.1558\Phi^3 - 13.2766\Phi^2 + 0.6546\Phi + 0.1802 \quad .$$

A comment is in order regarding the selection of the peak flow and axisymmetric performance values. In general, one can allow for an extra degree of freedom if such values are not imposed. The peak flow, Φ_p will depend on the coefficients in a nonlinear, implicit manner. Although this can be incorporated with extra coding effort, range constraints on Φ_p and $\psi_c(\Phi_p)$ must still be imposed, given the expectation of finding a bifurcation point close to the point at which the highest pressure is recorded. In this regard, the values chosen for the peak flow and pressure realistically reflect the influence of the range of experimental data on the expected solution.

Once the compressor characteristic model is determined, a validity test is performed with the bifurcation analysis code. Figure 1 shows a comparison between experimental data and the output of the bifurcation analysis code. As seen, the optimal solution has a rising part in accord with experiment and bifurcates at the imposed peak point. Notice that the highest values of the measured pressure are below the imposed peak value. This is in accord to the expectation that, if the peak point is an unstable

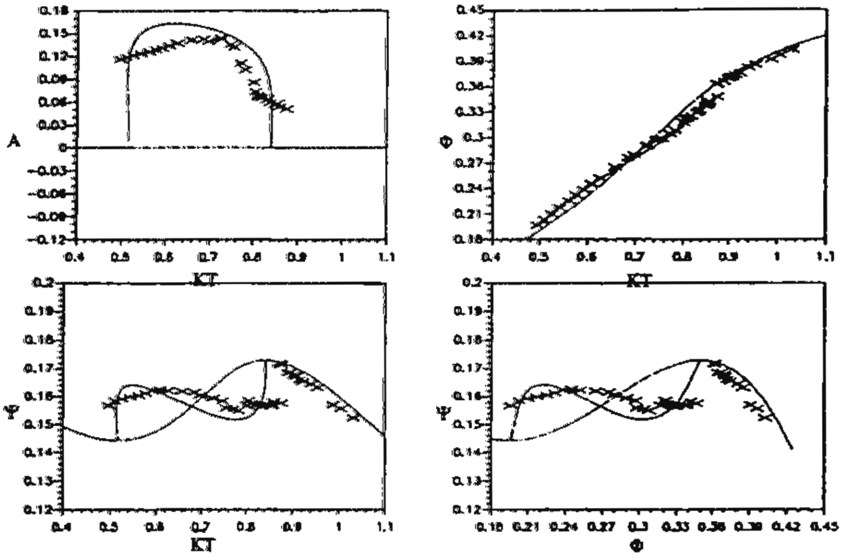


Figure 1: Bifurcation Diagrams for Steady-State Optimal Axisymmetric Map and Comparison with Experiment (x), Single Harmonic Model

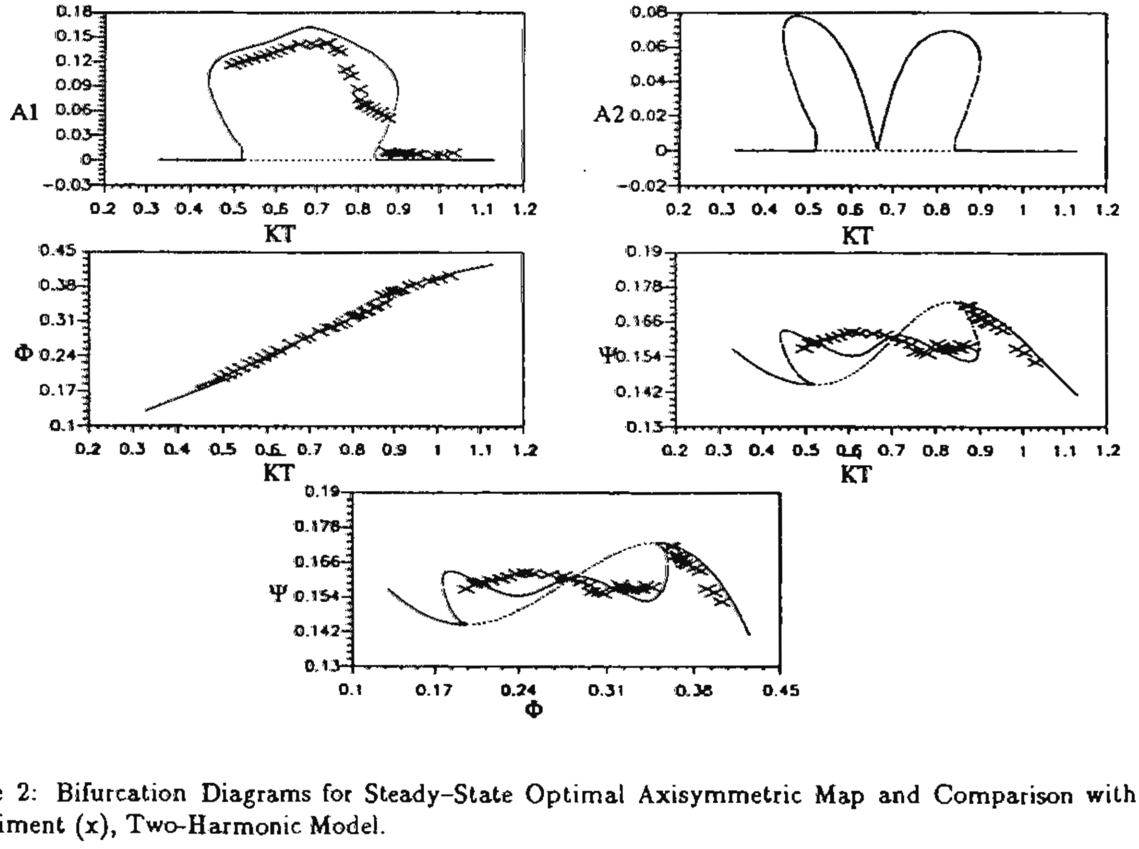


Figure 2: Bifurcation Diagrams for Steady-State Optimal Axisymmetric Map and Comparison with Experiment (x), Two-Harmonic Model.

equilibrium solution and hysteresis is present, then it will be difficult to measure the "peak" point state experimentally. However, hysteresis is minimal in the single-harmonic model. The results in [3] demonstrate that, for a given steady-state model, hysteresis is more pronounced when more harmonics are employed to capture the stall pattern. Figure 2 shows the results obtained from a bifurcation analysis of a two–harmonic model after computing an optimal solution for the axisymmetric polynomial coefficients based on the first–harmonic formulation presented here. This plot shows a reasonable agreement of the steady-state solutions, especially in the vicinity of the peak or stall inception point.

Conclusions

In this article, the authors demonstrate bow the qualitative nonlinear behavior of a compressor model can be used to obtain the relevant steady-state model. In the future, the authors hope to reformulate the problem of steady-state model identification to allow for more harmonics in the description of the compressor flow.

Acknowledgements This work has been supported under the ONR AASERT grant N00014-92-J-1794 and the ARO MURI Project DAAH049610008.

References

1. Moore, F. K. and Greitzer E. M., A Theory of Post–Stall Transients in Axial Compression Systems: Part 1, Development of Equations and Part 2, Application. ASME Journal of Turbomachinery, 108 (1996), 68 – 76 and 231 – 239.

2. Eveker, K. M., Model Development for Active Control of Stall Phenomena in Aircraft Gas Turbine Engines. PhD thesis, Georgia Institute of Technology, 1993.

3. Badmus, O. O., Nonlinear Dynamic Analysis and Control of Surge and Rotating Stall in Axial Compression System Models. PhD thesis, Georgia Institute of Technology, 1994.

4. Cbowdhury, S., An Experimental Ivestigation of Active Stall Control in Compression Systems. PhD thesis, Georgia Institute of Technology, 1995.

5. Rivera, C. J., Numerical Simulation of Dynamic Stall in Axial Compressor Blade Rows. PhD thesis, Georgia Institute of Technology, 1996.

6. Hale, J. and Koçak, H., Dynamics and Bifurcations, volume 3 of Texts in Applied Mathematics. Springer–Verlag, 1991.

7. Keller, H. B., Lectures on Numerical Methods in Bifurcation Theory. Springer–Verlag, 1987.

8. Doedel, E., AUTO: A Program for the Automatic Bifurcation Analysis of Autonomous Systems. Congressus Numerantium, 30 (1981) 265 – 284.

9. Spelluci, P., DONLP: A Program for Solving Dense Nonlinear Programming Problems. NETLIB, 1993.

MODELLING OF DISTRIBUTED PARAMETER SYSTEM FOR THE STATE OBSERVATION PURPOSES

W. Byrski and P. Kubik

Institute of Automatics
University of Mining & Metallurgy
Al.Mickiewicza 30, 30059 Kraków, Poland

Abstract. The progress in computer technique makes it possible to apply new sophisticated algorithms in digital feedback control application. One of these problems is the reconstruction of inaccessible initial-state vector of the system from the measurement data by the use of the observers. The structure of the observer is given not by the differential equation but originates directly from the definition of the exact observability. In the papers [1],[3],[4] general approach to exact reconstruction (observation) of finite state vector in linear time invariant system by the use of integral deterministic observers in Hilbert spaces was presented. The problem of optimal state reconstruction under assumption that errors not only in measurements of the output but also in measurements of the control may occur was also solved. These results one can apply to distributed parameter models. In this paper we present the modelling problems of distributed parameter system for such observation purposes.

Introduction

For the proper description of some physical phenomena the distributed parameter model should be used. Observability of such system means that an initial (or final) state can be uniquely determined from the output perfect measurement data. Generally for the distributed parameter system the initial state space is infinite-dimensional, hence a few questions related to system observability may arise, e.g. how many sensors we should use for guarantee the uniqueness of the observed state, where should these sensors be located and does the initial state depends continuously on measurement data. Especially the last one is very important if the measurement data are not perfect and small observation error may induce a large error to the initial state reconstruction. Generally the observability of the distributed parameter system does not assure continuous dependence of the reconstructed state on output measurements, hence in some cases such problem are not well-posed. All these problems were investigated in many works [5],[6],[8]-[13], however in these approaches the systems without inputs were considered. It was motivated by the fact that both distributed and boundary inputs are assumed to be known. Hence one can calculate that part of the solution which depends on these inputs and subtract it from observed data. This assumption specially in industrial applications is not always proper. Theoretically the control signal is known but it represents only the information which is sending to actuator and control valve (for instance the number of impulses). What is the real e.g. flow of heating steam which is the real input signal to process one can check only by measurement. This measurement can be also affected by disturbances. Hence measurement errors concern both output and input signal. This fact motivates the more general statement of the optimal state observation problem. It was stated by Byrski and Fuksa in [1], [7].

To omit the unwell-posed problem of continuity of infinite-dimensional initial state to measurement data the reconstruction of this state was transformed to reconstruction of finite dimensional vector of unknown parameters $x \in R^n$ (e.g. unknown amplitudes x_i of finite number of sinusoids which create the initial state) or in case of ordinary differential equation ODE this vector will represent standard initial or finite state $x(T) \in R^n$. The relations were formulated generally in Hilbert function spaces. The structure of the observer was given by the inner products of the output $y \in Y$ and input $u \in U$ measurements and special filtering functions $G_1(\tau)$, $G_2(\tau)$ on interval $[0,T]$. After the first observation interval $[0,T]$ the observer reconstructs the exact value of $x(T)$. The optimal functions $G_{1,2}(\tau)$ were chosen in such a way that they fulfilled observability requirements and minimized the norm of the observer. The observer with minimal norm guarantees minimal state reconstruction error for the disturbed measurements of y and u affected by the worst disturbances which belong to the unit balls in Y and U (disturbances with bounded norm). If the spaces Y and U are chosen as L^2 $[0,T]$ the inner product is represented by an integral operator. Some new extensions of the on-line exact observation were presented in [2] and [3]. In [2], the integral observers with Expanding and Moving Observation Window and their differential versions were given. In [3] a generalization to disturbances from an ellipsoid was derived.
The application to ODE system was showed in [4].

The most important properties of the proposed integral observers are:

- deterministic approach to exact reconstruction of the finite state,
- integral description of the on-line observer,
- fixed finite observation time interval T,
- noisy measurements of both input u(t) and output y(t),
- optimal properties resulting from the special form of the performance index of observation,
- application to systems with delay and distributed parameters.

The aim of this paper is to discuss the modeling of one-dimensional heat equation for such observation purposes and computation problems for the output measurements given by the ideal point observation and by integral sensor which represent spatial average of physical quantity over some effective sensing region located in different position. In the next section we will briefly recall the main formulas describing integral observers theory.

Statement of the observation problem

Consider a linear time invariant LTI system which output is given in a general operator form as the sum

$$y = \mathbf{H}_1 x + \mathbf{H}_2 u \qquad (1)$$

where the output y and control u belong to Hilbert function spaces Y and U, respectively and x is the unknown parameter: $x \in X = \mathbf{R}^n$. The maps are linear and continuous and the map $\mathbf{H}_1$ is defined as

$$\mathbf{H}_1 : X \rightarrow Y, \quad \text{where} \quad \mathbf{H}_1 x = \sum_{i=1}^{n} h_1^i x_i = \left[h_1^1, \cdots, h_1^n\right]\left[x_1, \cdots x_n\right]^T, \quad h_1^i \in Y, \qquad \text{and} \qquad \mathbf{H}_2 : U \rightarrow Y$$

The continuous measurements of function y and u on interval [0,T] are given. The observer for system (1) should reconstruct $x \in X = \mathbb{R}^n$ hence, in general it should be determined by n-dimensional linear continuous functionals on Y and U. By the Riesz Theorem every linear continuous functional in Hilbert spaces can be expressed as inner product (which we will denote by $<\cdot|\cdot>$ or $<\cdot,\cdot>$). Hence the observer is assumed as:

$$x = \mathcal{G}_1 y + \mathcal{G}_2 u = < \mathbf{G}_1 | \, y >_Y + < \mathbf{G}_2 | \, u >_U \qquad (2)$$

where the maps $\mathcal{G}_1 : Y \rightarrow X$, $\mathcal{G}_2 : U \rightarrow X$ are linear, continuous and the new operator $H_1 \in Y^n$ is defined as:

$$G_1 = \begin{bmatrix} g_1^1 \\ \vdots \\ g_1^n \end{bmatrix} \in Y^n, \quad G_2 = \begin{bmatrix} g_2^1 \\ \vdots \\ g_2^n \end{bmatrix} \in U^n, \quad H_1 = \begin{bmatrix} h_1^1 \\ \vdots \\ h_1^n \end{bmatrix} \in Y^n$$

In order to obtain the necessary and sufficient conditions for the relation (2) to be an observer for system (1) we substitute (1) to (2) and use the adjoint operator to $\mathbb{H}_2$

$$x = \left\langle \mathbf{G}_1 \middle| \mathbf{H}_1 x \right\rangle_Y + \left\langle \begin{bmatrix} \mathbf{H}_2^* g_1^1 \\ \vdots \\ \mathbf{H}_2^* g_1^n \end{bmatrix} \middle| u \right\rangle_U + \left\langle \mathbf{G}_2 \middle| u \right\rangle_U$$

Hence we have the following conditions for obserwation operators $\mathbf{G}_1$ and $\mathbf{G}_2$:

- observability condition - $\ker \mathbf{H}_1 = 0$,
- identity matrix constrain - $<\mathbf{G}_1 | \mathbf{H}_1> = \mathbf{I}$, (3)
- the formula for $\mathbf{G}_2$ $\qquad \mathbf{G}_2 = -\mathbf{H}_2^* \mathbf{G}_1$. (4)

Condition (3) takes the form of a matrix inner product in Y^n and represents the constraint for $\mathbf{G}_1$

$$\left\langle \begin{bmatrix} g_1^1 \\ \vdots \\ g_1^n \end{bmatrix} \middle| \begin{bmatrix} h_1^1, \ldots, h_1^n \end{bmatrix} \right\rangle = \begin{bmatrix} \langle g_1^1, h_1^1 \rangle, \ldots, \langle g_1^1, h_1^n \rangle \\ \vdots \\ \langle g_1^n, h_1^1 \rangle, \ldots, \langle g_1^n, h_1^n \rangle \end{bmatrix} = \mathbf{I}$$

The other form of the condition (4) may be given by the operators G_1 and G_2 which transpositions are defined by: $G_2' = -\mathbf{H}_2^* G_1'$; or $\left[g_2^1,\ldots,g_2^n\right] = -\mathbf{H}_2^*\left[g_1^1,\ldots,g_1^n\right]$

There is an infinite number of pairs (G_1, G_2) which fulfill formula (2), constraint (3) and formula (4).

Optimal observation problem

Assume we have an LTI observable system (1) and the observer (2) of the unknown parameter x. In the space S of all observer pairs (G_1, G_2), $S = Y^n \times U^n$ we define the norm of the observer

$$\|(G_1,G_2)\|_S^2 \overset{df}{=} \sum_{i=1}^{n}\left\langle g_1^i, g_1^i\right\rangle + \sum_{i=1}^{n}\left\langle g_2^i, g_2^i\right\rangle = J$$

which represents also the performance index of observation (if the measurement disturbances are unknown but bounded we can use the worst case approach assuming that the worst disturbances with unity norm will active. Such approach is called sometimes as guaranteed estimation [10]). The optimization task is:

$$J^o = \min_{(G_1,G_2)} J$$

The constraint (3) and the performance index J give the Lagrangian functional

$$L = J + 2\sum_{i=1}^{n}\left(e_i' - [\langle g_1^i, h_1^1\rangle,\ldots,\langle g_1^i, h_1^n\rangle]\right)\lambda_i$$

where vectors $\lambda_i \in R^n$ are Lagrange multipliers and e'_i are transpositions of the basis vectors in R^n. The necessary condition of minimum together with (4) give the relation for each optimal element g_1^i,

$$g_1^i + \mathbf{H}_2\mathbf{H}_2^*\, g_1^i - \sum_{j=1}^{n} h_1^j\,\lambda_i^j = 0,$$

or in matrix form: $\left[g_1^1,\ldots,g_1^n\right] = \mathbf{H}_2\left[g_2^1,\ldots,g_2^n\right] + \mathbf{H}_1\,\lambda, \quad \Rightarrow \quad G_1' = \mathbf{H}_2\, G_2' + \mathbf{H}_1\,\lambda,$ (5)

where λ is a matrix of columns λ_i. The form of the final solution from (3) and (5) for optimal $G_1'^o$, $G_2'^o$ is

$$G_1' = \mathbf{F}^{-1}\mathbf{H}_1\left\langle H_1 \middle| \mathbf{F}^{-1}\mathbf{H}_1\right\rangle^{-1}, \qquad G_2' = -\mathbf{H}_2^* G_1' \tag{6}$$

where the scalar operator F is: $\mathbf{F} = 1 + \mathbf{H}_2\mathbf{H}_2^*$.

Although the reconstructed state is the vector of finite dimension the general final formulas (6) can be applied also to some observation problems for the systems with time delay and with distributed parameters.

Modeling of the heat process

One-dimensional parabolic heat equation (called sometimes diffusion equation) is assumed for heat transfer by conduction in a homogeneous rod or thin wire of L length under assumption that the surface of the rod is insulated.

$$k^2 \cdot \frac{\partial^2 T(t,z)}{\partial z^2} = \frac{\partial T(t,z)}{\partial t}, \tag{7}$$

where: $T(t,z)$ is temperature at the time t in the point z, and constant $k^2 = K/c\rho$ is called thermal diffusivity (K- thermal conductivity, c - specific heat, ρ - mass per unit volume), z - denotes a spatial dimension. .

The initial condition ; $T(0,z) = \varphi(z)$, for $0 < z < L$,

and Dirichlet boundary conditions $T(t, 0) = \Psi_1(t)$; $:T(t, L) = \Psi_2(t)$. are given.

The observations will be generated by two type of sensors located in point α, $0<\alpha<L$ ideal point observation and spatially averaged observation, what gives different equations of the output::

1) $y(t,\alpha) = T(t, \alpha)$,

2) $y(t,\alpha) = \frac{1}{2\varepsilon}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon} T(t,\xi)d\xi$, where 2ε is diameter of thermocouple sensor located in α, $0<\alpha<L$.

Hence, different form of the operators $\mathbf{H_1}, \mathbf{H_2}$ in the equation (1): $\mathrm{y} = \mathbf{H_1}\mathrm{x} + \mathbf{H_2}\mathrm{u}$ will be obtained. The vector $\mathrm{x} \in \mathrm{R}^n$ is a finite dimension parameter which characterize unknown initial temperature $T(0,z) = \varphi(z)$.
We assume that the initial condition $T(0,z)$ is given as the sum of finite number of sinusoid (eigenfunctions).

$$T(0,z) = \varphi(z) = \sum_{j=1}^{n} x_j \cdot \sin(\frac{j \cdot \pi}{L} z) \tag{8}$$

For the Dirichlet boundary conditions: $u(t,0) = \psi_1(t)$; $u(t,L) = \psi_2(t)$ the solution of (7) is given by

$$T(t,z) = \sum_{i=1}^{\infty} T_i(t) \cdot \sin(\frac{i \cdot \pi}{L} z) \tag{9}$$

where

$$T_i(t) = e^{-(\frac{i\pi k}{L})^2 t} \left[C_i + \frac{2i\pi k^2}{L^2} \int_0^t e^{(\frac{i\pi k}{L})^2 \tau} \left[\psi_1(\tau) - (-1)^i \psi_2(\tau) \right] d\tau \right] \tag{10}$$

and C_n represents influence of initial condition

$$C_i = T_i(0) = \frac{2}{L} \int_0^L \varphi(\xi) \sin \frac{i\pi\xi}{L} d\xi \tag{11}$$

The final formula will be derived for exemplary control which is given by the boundary condition in left hand side of the rod $u(t) = T(t,0) = \Psi_1(t)$,.whereas right hand side end is held at temperature zero $T(t,L) = \Psi(t)=0$ for all the time t>0. Then we have

$$T(t,z) = \sum_{i=1}^{\infty} e^{-(\frac{i\pi k}{L})^2 t} \cdot T_i(0) \cdot \sin(\frac{i \cdot \pi}{L} z) + \sum_{i=1}^{\infty} \frac{2i\pi k^2}{L^2} \int_0^t e^{-(\frac{i\pi k}{L})^2 (t-\tau)} \psi_1(\tau)\, d\tau \cdot \sin(\frac{i \cdot \pi}{L} z)$$

We can write this equation in the form

$$T(t,z) = T_0(t,z) + T_u(t,z) \tag{12}$$

where T_0 depends on initial condition and T_u on the boundary control.
Substituting initial condition (8) to (11) and to $T_0(t,z)$ and taking into account that

$$\int_0^L \sin(j\pi \frac{\xi}{L}) \sin(i\pi \frac{\xi}{L}) d\xi = 0, \quad \text{for} \quad \forall i \neq j \quad \text{and} \quad \int_0^L \sin(j\pi \frac{\xi}{L}) \sin(j\pi \frac{\xi}{L}) d\xi = \frac{L}{2},$$

we will have only sum of finite number of elements in $T_0(t,z)$

$$T_0(t,z) = \sum_{j=1}^{n} x_j \cdot e^{-(\frac{j \cdot \pi k}{L})^2 t} \cdot \sin(\frac{j \cdot \pi}{L} z) \tag{13}$$

and sum of infinite series in $T_u(t,z)$

$$T_u(t,z) = \sum_{i=1}^{\infty} \left[\frac{2i\pi k^2}{L^2} \int_0^t e^{-(\frac{i\pi k}{L})^2 (t-\tau)} \psi_1(\tau) d\tau \quad \sin(\frac{i \cdot \pi}{L} z) \right] \tag{14}$$

Observation problem 1

The ideal point observation sensors is located at point α, $0<\alpha<L$, then from (1), (13), (14) the operators $\mathbf{H_1}$ and $\mathbf{H_2}$ are given by formula:

$$y_1(t,\alpha) = T_0(t,\alpha) = \sum_{j=1}^{n} h_1^j x_j = \left[h_1^1, \quad h_1^2, \quad \ldots \quad h_1^n \right] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{H_1}\, \mathbf{x}, \quad \text{where} \quad h_1^j = e^{-(\frac{j \cdot \pi k}{L})^2 t} \sin(\frac{j \cdot \pi}{L} \alpha)$$

$$y_2(t,\alpha) = T_u(t,\alpha) = \int_0^t \left[\frac{2\pi k^2}{L^2} \sum_{i=1}^{\infty} i e^{-(\frac{i\pi k}{L})^2 (t-\tau)} \sin(\frac{i \cdot \pi}{L} \alpha) \right] \psi_1(\tau) d\tau = \mathbf{H_2}\ \mathbf{u} \cdot \tag{15}$$

It is easy to see that the condition of observability $\ker \mathbf{H}_1 = 0$, is fulfilled for $\forall\ \alpha$, $0<\alpha<L$.

Remarks on calculation method for integral in formula (14)

Numerical tests with different methods of integration have given insufficient accuracy of results. To improve the calculation correctness one can notice that the coefficients $T_i(t)$ in (14)

$$T_i(t) = \int_0^t e^{-(\frac{i\pi k}{L})^2(t-\tau)} \frac{2i\pi k^2}{L^2} \psi_1(\tau) d\tau$$

may be treated as the elements of vector solution of matrix differential equation with zero initial condition

$$\begin{aligned} \dot{\tilde{T}}(t) &= A\tilde{T}(t) + Bu(t) \\ \tilde{T}_u(t) &= \quad C\tilde{T}(t) \end{aligned} \tag{16}$$

where $\tilde{T}(t) = [T_1(t), T_2(t), \dots\dots T_N(t)]^{\mathrm{T}}$, and matrix $A = diag[-\frac{(\pi k)^2}{L^2}, -\frac{(2\pi k)^2}{L^2}, \cdots, -\frac{(N\pi k)^2}{L^2}]_{N\times N}$,

$$B = \left[\frac{2\pi k^2}{L^2}, \frac{4\pi k^2}{L^2}, \dots, \frac{2N\pi k^2}{L^2}\right]^T_{N\times 1}, \qquad C = [1, 1, \cdots, 1]_{N\times N}$$

where N is assumed finite number of terms in series (14). In this approach very good results by the use of the procedure ***lsim*** in Matlab package for solution of equation (16) were obtained.

Observation problem 2

For spatially averaged observation the integral sensor is located at point α, $0<\alpha<L$. The output is given by:

$$y(t,\alpha) = \frac{1}{2\varepsilon}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon} T(t,\xi)d\xi = \frac{1}{2\varepsilon}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon} T_0(t,\xi)d\xi + \frac{1}{2\varepsilon}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon} T_u(t,\xi)d\xi \tag{17}$$

The first integral from (17) and (13) is:

$$y_1(t,\alpha) = \frac{1}{2\varepsilon}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon} T_0(t,\xi)d\xi = \frac{1}{2\varepsilon}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon}\sum_{j=1}^{n} x_j \cdot e^{-(\frac{j\cdot\pi k}{L})^2 t} \sin(\frac{j\cdot\pi}{L}\xi)\, d\xi = \frac{1}{2\varepsilon}\sum_{j=1}^{n} x_j \cdot e^{-(\frac{j\cdot\pi k}{L})^2 t}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon} \sin(\frac{j\cdot\pi}{L}\xi)\, d\xi$$

The final solution for $y_1(t,\alpha)$ is

$$y_1(t,\alpha) = \frac{L}{\pi\varepsilon}\sum_{j=1}^{n}\frac{1}{j} x_j \sin(\frac{j\cdot\pi}{L}\alpha)\sin(\frac{j\cdot\pi}{L}\varepsilon)\cdot e^{-(\frac{n\cdot\pi k}{L})^2 t}$$

The last formula can be written as inner product of vector function $\mathbf{H}_1$ and vector of parameter $\mathbf{x} = [x_1, x_2, \dots\dots x_n]^{\mathrm{T}}$

$$y_1(t,\alpha) = \sum_{j=1}^{n} h_1^j x_j = \begin{bmatrix} h_1^1, & \cdots & , h_1^n \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{H_1}\ \mathbf{x}, \text{ where } h_1^j = \frac{L}{j\pi\varepsilon} e^{-(\frac{j\pi k}{L})^2 t} \sin(\frac{j\pi}{L}\alpha)\sin(\frac{j\pi}{L}\varepsilon)\cdot \tag{18}$$

The second integral is

$$y_2(t,\alpha) = \frac{1}{2\varepsilon}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon} T_u(t,\xi)d\xi \ = \frac{1}{2\varepsilon}\int_{\alpha-\varepsilon}^{\alpha+\varepsilon}\sum_{i=1}^{\infty}\frac{2i\pi k^2}{L^2}\int_0^t e^{-(\frac{i\pi k}{L})^2(t-\tau)}\psi_1(\tau)\, d\tau\cdot\sin(\frac{i\cdot\pi}{L}\xi)\, d\xi\cdot$$

After its transformation we will have the form of operator $\mathbf{H}_2$

$$y_2(t,\alpha) = \frac{\pi k^2}{\varepsilon L^2}\sum_{i=1}^{\infty} i\int_0^t e^{-(\frac{i\pi k}{L})^2(t-\tau)}\psi_1(\tau)d\tau\int_{\alpha-\varepsilon}^{\alpha+\varepsilon}\sin(\frac{i\pi}{L}\xi)d\xi = \frac{2k^2}{\varepsilon L}\int_0^t\left[\sum_{i=1}^{\infty} e^{-(\frac{i\pi k}{L})^2(t-\tau)}\sin(\frac{i\pi}{L}\alpha)\sin(\frac{i\pi}{L}\varepsilon)\right]\psi_1(\tau)\, d\tau$$

$$= \mathbf{H}_2\,\mathbf{u}\,. \tag{19}$$

In this type of observation the same condition of observability occurs: for $\forall\alpha$: $0<\alpha<L \Rightarrow \ker \mathbf{H}_1 = 0$.

Numerical example

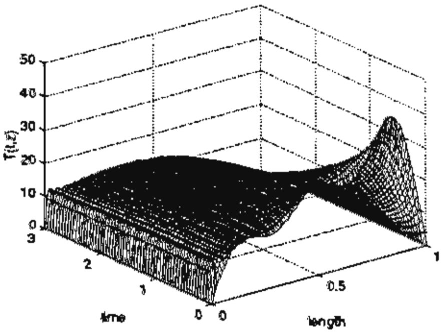


Fig.1. Observation problem 1

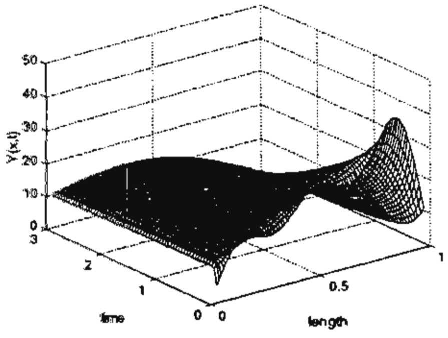


Fig.2. Observation problem 2

Conclusions

In the paper the analytical form of system output (1) observed by two kind of sensors was derived. From this form explicit formulas for operators $\mathbf{H}_1$, $\mathbf{H}_2$ result. It will enable us to design the optimal state observer for distributed parameter system based on presented theory. The conditions of observability in two cases were checked. In Fig.1 and 2 numerical results of simulation for the system with parameters: k=0.2, L=1.0, n=5, N=50, $\psi_1(t)=10\cdot 1(t)$, steps of numerical discretization $\Delta t=0.05$, $\Delta z=0.01$, and radius of sensor $\varepsilon=3\Delta z$ were presented.

Acknowledgments. This work was partly supported by the Polish Committee of Scientific Research under Grants no.18.120.447 AGH and 11.120.376/96, AGH.

References

1. Byrski W., S.Fuksa, "Optimal finite parameter observer. An application to synthesis of stabilizing feedback for a linear system", Control and Cybernetics, vol. 13, 1984, No.1-2.
2. Byrski W., "Optimal State Observers with Moving and Expanding Observation Window", Procc.of IASTED, XII Intern.Conference on Model.&Sim. Innsbruck, 1993, .68-74.
3. Byrski W. "Integral Description of the Optimal State Observers", *Procc. of II European Control Conference,ECC'93* , Groningen, vol.4, 1993, 1832-1838.
4. Byrski W. „Theory and application of the optimal integral state observers", *Procc. of III European Control Conference,ECC'95,* Roma, vol.1,1995, 526-532
5. Dolecki Sz., Observability for the one-dimensional heat equation, Studia Math., 48 (1973), 291-305.
6. Dolecki Sz. And Russell D.L., A General theory of observation and control., SIAM .J. Control and Optimization , Vol. 15, No.2, February 1977, 185-220.
7. Fuksa S., W.Byrski. "General approach to linear optimal estimator of finite number of parameters", *IEEE Trans.on Automatic Contr.*, vol.29, No.5, 1984, 470-473.
8. Goodson R.E. and Klein R.E., A definition and some results for distributed system observability., IEEE Trans. Automatic Control, AC-15 (1970) ,165-174.
9. Jai A.El and Pritchard., Sensors and Actuators in the Analysis of Distributed Systems. New York:J.Wiley, 1988.
10. Kurzhanski A.B. and Khapalov A.Yu., An observation theory for distributed-parameter systems., J.of Mathematical Systems, Estimation and Control.Vol.1.No.4, 1991, Birhaeuser, 389-440.
11. Kobayashi T, Initial state determination for distributed parameter systems. SIAM .J. Control and Optimization , Vol. 14, No.5, August 1976, 934-944.
12. Sakawa Y., Observability and related problems for partial differential equations of parabolic type. SIAM J.Control,Vol.13.No,1 January 1975, 15-27.
13. Rolewicz S., On optimal observability of linear systems with infinite-dimensional states., Studia Math.,44, 1972, 411-416.

A FAULT DETECTION FILTER FOR BILINEAR SYSTEMS WITH UNKNOWN INPUTS

Mechmeche C., Zasadzinski M., Rafaralahy H., Keller J.Y., Darouach M.

CRAN–ACS–CNRS URA 821, Université Henri Poincaré–Nancy 1, 186, rue de Lorraine, 54400 Cosnes et Romain, FRANCE
E-mail : mzasad@iut-longwy.u-nancy.fr

Abstract : An extension of White and Speyer's detection filter to bilinear systems with unknown inputs is presented. Under an appropriate transformation, the detection filter design for bilinear systems subjected to unknown inputs is solved as a special case of unknown input observers for linear systems by introducing eigenspace constraints. A simple solution for the detection filter gain and closed-loop eigenvectors is proposed.

Key words : Fault detection and isolation, bilinear systems, unknown inpnt observer.

1- Introduction

In recent years there has been a significant growth in the need for sophisticated diagnostic procedures in a variety of industrial processes. The ever increasing presence of complex electronic controls has dictated the need for accurate and timely diagnosis of the sensors and actuators that are part of these subsystems. Consequently, many of researchers have investigated the design of Fault Detection and Isolation (FDI) algorithms and demonstrated their feasibility and applicability to actual systems. Beard [2] developed the detection filter design method for linear systems. This approach was based on an observer designed so that the occurrence of a fault in the system yields a fixed direction (unidirectional) of the residual vector in the output space. More recently, White and Speyer [12] and Park and Rizzoni [11] improved this design procedure using a spectral approach that is suitable for the detection of multiple faults. They directly considered (and assigned all) the closed-loop eigenvalues of the detection filter with eigenvector constraints, yielding an algebraic algorithm for determining the detection filter gain.

The major limitation of detection filter has been its applicability only to linear time-invariant systems without unmeasured disturbance. However a wide variety of industrial systems and chemical process can be described by a bilinear system [10]. FDI for bilinear systems is seldom studied while the control of bilinear systems is well developed. The existence conditions and design procedure for residual generator for bilinear systems is investigated in [9].

In this paper we propose to extend the detection filter to bilinear systems subjected to unmeasured disturbances and faults. A design procedure using eigenvalues-eigenvectors approach is used in order to determine an appropriate direction of the residual vector in the measured output space. We propose a new algorithm to design a detection filter gain using the Kronecker product; the novelty in the approach lies in the ability to generate a closed-form expression for the detection filter gain, leading to a greatly simplified design procedure. This paper offers three major contributions : (1) it makes the design filter detection gain more simple; (2) it extends the detection filter developed in [12] to bilinear systems; (3) it includes the decoupling between residuals to be tested and unmeasured disturbances in the detection filter synthesis. The paper is organized as follows. In section 2 the objectives of the fault detection filter are stated. In section 3 the fault detection filter for bilinear systems is formulated as a special case of unknown input observer. The necessary and sufficient conditions are given. In section 4 a complete solution and a synthesis procedure of the detection filter to bilinear systems are given.

Notations : Im(A) and Ker(A) are full rank matrices which span the range and the null spaces respectively of A. A^+ is a generalized inverse of matrix A defined by $A = A A^+ A$.

2- Problem Formulation

We consider the following bilinear system described by

$$\dot{x} = A x + \sum_{i=1}^{m} N_i u_i x + B u + D d + \sum_{i=1}^{r} F_i f_i \tag{1.1}$$

$$y = C x \tag{1.2}$$

where the state vector $x(t) \in \mathbb{R}^n$, the input vector $u(t) = [\ u_1(t)\ \ \dots\ \ u_m(t)\ ]^T \in \mathbb{R}^m$, the unknown input vector $d(t) \in \mathbb{R}^q$ and the output $y(t) \in \mathbb{R}^p$. A, B, C, N_i, D are constant matrices. In the remainder of this paper we will assume that faults may be modelled by an additive known equation (1.1), where $F_i \in \mathbb{R}^n$ is defined an component fault event vector, and $f_i(t)$ is a scalar function which represents the evaluation of the fault. Without loss of generality, we assume that matrix C is surjective, i.e.

$$\operatorname{rank} C = p \tag{1.3}$$

The detection filter is given in the form of the following estimator

$$\dot{z} = H z + L y + J u + \sum_{i=1}^{m} E_i u_i y \tag{2.1}$$

$$\hat{y} = C z + P y \tag{2.2}$$

where $z(t) \in \mathbb{R}^n$ and $\hat{y}(t) \in \mathbb{R}^p$. H, L, E_i, J, and P are unknown matrices of appropriate dimensions. In nonlinear systems, specially in bilinear systems, there exist many kinds of observers, defined by a constraint on the observer error [4], [5]. In this paper, this constraint is the same than considered in [5]. Define the residual vector as follows

$$r = \hat{y} - y \tag{3}$$

The objective of fault detection in the context of this work will be to restrict the output error residual, r(t), to a certain direction while maintaining the freedom to select the dynamics of the observation error of the filter (2). This is the conventional definition of fault detectability, given in definition 1.

Definition 1 : The failure associated with $f_i(t)$ in the system described by (1) is detectable if there exists a filter gain K such that

r(t) maintains a fixed direction in the output space, (i)
the filter is stable, (ii)
r(t) is independent of d(t). (iii)□

Condition (i) forces the filter to have properties such that the output error direction r(t) can be associated with the design error direction $f_i(t)$. Condition (ii) is imposed so that the filter can be made stable. Conditions (iii) corresponds to unknown input observer theory.

3- Unknown input observer design

This section is devoted to the decoupling of the residual r(t) from the unknown input d(t) and the bilinearities, with stability constraint, the residual observer (2) must have a linear reconstruction error. Let e(t) be the reconstruction error given by

$$e = z - S\,x \tag{4}$$

where matrix $S \in \mathbb{R}^{n \times n}$.

Theorem 1 : If the following constraints are verified

$$H\,S - S\,A + L\,C = 0 \tag{5.1}$$

$$E_i\,C - S\,N_i = 0 \qquad i = 1 \ldots m \tag{5.2}$$

$$S\,D = 0 \tag{5.3}$$

$$J = S\,B \tag{5.4}$$

and H is a stability matrix, then the error dynamics is exponentially stable and can be written as

$$\dot{e} = H\,e + \sum_{i=1}^{r} S\,F_i\,f_i \tag{6.1}$$

In addition, the residual vector becomes

$$r = C\,e \tag{6.2}$$ ■

The proof is straightforward and is omitted. Condition (i) of definition 1 corresponds to an invariance problem in the output space, which can be translated into an eigenvalues-eigenvectors problem. In [3], Darouach et al. have shown that matrix H can be always written as

$$H = S\,A - K\,C \tag{7}$$

where K is the gain matrix to be determined. This eigenvalues-eigenvectors problem can be solved by constraining the design of the gain matrix K. Then, according to theorem 1, constraints (5-1)-(5-3) must be solved independently of the gain matrix design. The following theorem shows that this is possible if matrix S defined in equation (4) has a particular form.

Theorem 2 : Assume that matrix H in (2.1) is given by (7), then the constraints (5.1)-(5.3) of the theorem 1 are satisfied for any matrix $S\,A \in \mathbb{R}^{n \times n}$ and gain matrix $K \in \mathbb{R}^{n \times p}$ if and only if the matrix S in (4) can be written as

$$S = I_n - \alpha\,C \tag{8}$$

where α is a given matrix of appropriate dimension. ■

Proof : Substituting equation (7) into (5-1) gives the following equation

$$LC = \begin{bmatrix} K & SA \end{bmatrix} \begin{bmatrix} CS \\ I_n - S \end{bmatrix}$$

which have a solution L independently of matrices S A and K if and only if

$$\operatorname{rank}(C) = \operatorname{rank} \begin{bmatrix} C^T & S^T C^T & I_n - S^T \end{bmatrix}^T$$

or, since matrix C is surjective, equivalently

$$\operatorname{rank} \begin{bmatrix} I_p & 0 \end{bmatrix} = \operatorname{rank} \begin{bmatrix} I_p & 0 \\ CSC^+ & CSKer(C) \\ (I_n - S)C^+ & (I_n - S)Ker(C) \end{bmatrix} \Leftrightarrow \begin{cases} CSKer(C) = 0 \\ (I_n - S)Ker(C) = 0 \end{cases}$$

This is equivalent to the existence of a matrix α such that

$$C^T\,\alpha^T = (I_n - S^T)$$

then matrix S is given by (8). ◆

Inserting (8) into (5.1) yields

$$L = K\,(I_p - C\,\alpha) + S\,A\,\alpha \tag{9}$$

Define $N_a = [\,N_1 \;\ldots\; N_m\,]$, $E_a = [\,E_1 \;\ldots\; E_m\,]$ and $C_a = \operatorname{diag}(C)$ with $C_a \in \mathbb{R}^{(p.m)\times(n.m)}$. The existence conditions for a stable detection filter satisfying the constraints (5.1)-(5.4) of theorem 1 are given by the following theorem.

Theorem 3 : Assume that the matrix S is given by (8). There exists a detection filter given by (2) for the bilinear system (1) satisfying theorem 1 if and only if the following conditions hold

$$\operatorname{rank}(C\,\Phi) = \operatorname{rank}\Phi \tag{10.1}$$

and

$$\operatorname{rank} \begin{bmatrix} sI_n - A & \Phi \\ C & 0 \end{bmatrix} = n + \operatorname{rank}\Phi,\ \forall\, s \in \mathbb{C},\ \operatorname{Re}(s) \geq 0 \tag{10.2}$$

where

$$\Phi = [\;D \quad N_a Ker(C_a)\;] \tag{10.3}$$ ■

Proof : Inserting equation (8) into relations (5.2) and (5.3) gives

$$[\ \alpha \quad E_a\ ]\begin{bmatrix} CD & CN_a \\ 0 & C_a \end{bmatrix} = [\ D \quad N_a\ ]$$

The matrix $[\ C_a{}^+ \quad Ker(C_a)\ ]$ being nonsingular [8], the previous equation is equivalent to

$$[\ \alpha \quad E_a\ ]\begin{bmatrix} CD & CN_a \\ 0 & C_a \end{bmatrix}\begin{bmatrix} I_q & 0 & 0 \\ 0 & C_a{}^+ & Ker(C_a) \end{bmatrix} = [\ D \quad N_a\ ]\begin{bmatrix} I_q & 0 & 0 \\ 0 & C_a{}^+ & Ker(C_a) \end{bmatrix}$$

and can be written as $[\ \alpha \quad E_a\ ]\begin{bmatrix} CF & CN_aC_a{}^+ & CN_aKer(C_a) \\ 0 & I_{p.m} & 0 \end{bmatrix} = [\ F \quad N_aC_a{}^+ \quad N_aKer(C_a)\ ]$ (11)

Then, by using relations (8) and (10.3), equation (11) becomes

$$E_a = S\ N_a\ C_a{}^+ \qquad (12.1)$$

and

$$S\ \Phi = 0 \qquad (12.2)$$

Hence relations (5.2) and (5.3) are replaced by (12.2) and E_i is given by (12.1). Then (5.1), (5.4) and (12.2), with the stability of matrix H, correspond to the necessary and sufficient constraints to design an unknown input observer for the following system [8]

$$\dot{x} = A\ x + B\ u + \Phi\ d_a \qquad (13.1)$$

$$y = C\ x \qquad (13.2)$$

where $d_a(t)$ is an unknown input from which the observation error must be decoupled. The necessary and sufficient existence conditions for this unknown input observer are given by (10.1) and (10.2) [3], [6]. ◆

Then for given matrices K and α, the filter matrices J, H, S and L are given by (5.4), (7), (8) and (9) respectively. The gain matrix K being determined in section 4, then to end this section, it remains to compute α and P. Using (8), equation (12.2) can be rewritten as $\alpha\ C\ \Phi = \Phi$

and has a solution if and only if condition (10.1) is satisfied. This solution is given by

$$\alpha = \Phi\ (C\ \Phi)^+ + Y\ (I_p - (C\ \Phi)\ (C\ \Phi)^+) \qquad (14)$$

where Y is an arbitrary matrix of appropriate dimension. In [3], Darouach et al. have shown that matrix Y must be chosen such that S is of maximal rank, otherwise supplementary unobservable modes for the pair (S A, C) may arise, which can lead to an unstable matrix H. (Note that the unobservable modes of (S A, C) are given by the scalars $s \in \mathbb{C}$ generating a rank deficiency in condition (10.1) [3].) An available choice to obtain S of maximal rank is to set $Y = 0$.

The maximal rank of S is given by $\quad \text{rank } S = k \quad$ (15.1)

where $\quad \text{rank } \Phi = n - k \quad$ (15.2)

Then α in (14) becomes $\quad \alpha = \Phi\ (C\ \Phi)^+ \quad$ (16)

and, using (2.2), (6.2) and (8), P is given by $\quad P = C\ \alpha \quad$ (17)

4- Detection filter gain synthesis

This section is devoted to the computation of the gain matrix K such that r(t) verify conditions (i) and (ii) of definition 1 (note that condition (iii) is satisfied if the filter matrices are chosen as shown in section 3). Firstly, in subsection 4.1, we deal with the detection space constraints. In subsection 4.2, an algorithm is proposed to determine the gain matrix K.

4.1- Detection space constraints

4.1.1- Eigenspaces constraints

Consider (6.1)-(6.2) and define $\bar{F}$ as $\quad \bar{F} = \left[\bar{F}_1 \quad \cdots \quad \bar{F}_r\right] = \left[SF_1 \quad \cdots \quad SF_r\right]$ (18)

The failures $f_i(t)$ can be detected in the output space if they do not belong to Ker(S), i.e. if

$$Im(\bar{F}) \cap Ker(S) = \{0\} \Leftrightarrow \text{rank } \bar{F} = \text{rank}\left[F_1 \quad \cdots \quad F_r\right] = r \qquad (19)$$

Since the detection filter (2) for the bilinear system (1) can be seen as a unknown input observer for the linear system (13), the error e(t) (4) associated to the dynamic equation (6.1) can be decomposed into two invariant subspaces as follows [1] $\quad$ if $e(0) \in Im(\Phi) = Ker(S)$ and $\dot{e} = H\ e$, then $e(t) \in Ker(S)$ (a)

if $e(0) \in Im(S) = Ker(\Phi^T)$ and $\dot{e} = H\ e$, then $e(t) \in Im(S)$ (b)

Then the eigenvalues-eigenvectors problem related to the detection filter for system (1) implies the following fact.

Fact 1 : Ker(S) = Im(Φ) is the eigenspace associated to the decoupling of the unknown input d(t) and the bilinearities. The eigenspace associated to the failure directions is included in Im(S) (i.e. $Im(\bar{F}) \subset Im(S)$, see (18)). ▲

Condition (i) of definition 1 implies that each vector $\bar{F}_i$ belongs to an observable eigenspace, i.e.

each vector $\bar{F}_i$ can be written as $\quad \bar{F}_i = SF_i = \sum_{j=1}^{\delta_i} \alpha_j^i v_j^i \qquad i = 1 \ldots r \quad$ (20)

where δ_i is the dimension of the eigenspace associated to $\bar{F}_i$, α_j^i are scalars and v_j^i the eigenvector of H with $j = 1 \ldots \delta_i$.

Since the eigenspace associated to v_j^i ($i = 1 \ldots r$ and $j = 1 \ldots \delta_i$) is assumed to be observable, one obtains

$$C v_j^i \neq 0 \tag{21}$$

Denote λ_j^i the eigenvalue of H related to the eigenvector v_j^i. Then, using (7), the eigenspace problem (20) can be written as follows [12]

$$\begin{bmatrix} \lambda_j^i I_n - \overline{A} & K \\ C & 0 \end{bmatrix} \begin{bmatrix} v_j^i \\ w_i \end{bmatrix} = \begin{bmatrix} 0 \\ w_i \end{bmatrix} \qquad i = 1 \ldots r \tag{22.1}$$

with

$$w_i = C v_j^i = C\overline{F}_i \qquad j = 1 \ldots \delta_i \tag{22.2}$$

Note that λ_j^i can be arbitrarily chosen with $\mathrm{Re}(\lambda_j^i) < 0$ (use the observability obtain by (21)). Using fact 1, we have for $i = 1 \ldots r$ and $j = 1 \ldots \delta_i$

$$v_j^i \in \mathrm{Im}(S) \tag{23}$$

The invariant zeros of the triplet (A, Φ, C) are the unobservable mode of (S A, C) [3] and the eigenspace of matrix (S A) associated to these invariant zeros are the zero-directions of the triplet (A, Φ, C). This vector space is invariant for the matrix H (7) and corresponds to the unobservable space of the pair (S A, C). Let $\mathcal{V}$ be a basis of this unobservable space and $\delta_v = \mathrm{rank}\ \mathcal{V}$. By definition, we have

$$\mathrm{Im}(\mathcal{V}) \subset \mathrm{Im}(S) \tag{24}$$

Since $\mathrm{Ker}(C) \subset \mathrm{Im}(\mathcal{V})$ then, using (21), we have for $i = 1 \ldots r$ and $j = 1 \ldots \delta_i$

$$v_j^i \notin \mathrm{Im}(\mathcal{V}) \tag{25}$$

4.1.2- Construction of the detection spaces

The vector $\overline{F}_i$ is a linear combination of eigenvectors v_j^i who span his detection space, then $\overline{F}_i$ must satisfy the following equation (see (22))

$$(\overline{A} - KC)\ \overline{F}_i = \lambda_j^i\ \overline{F}_i \qquad i = 1 \ldots r \tag{26.1}$$

or, by using (20),

$$K\,C\,\overline{F}_i = \overline{A}\,\overline{F}_i - \sum_{j=1}^{\delta_i} \alpha_j^i \lambda_j^i v_j^i \qquad i = 1 \ldots r \tag{26.2}$$

The solution of (26.2) is given by [12]

$$K = (\overline{A}\,\overline{F}_i - \sum_{j=1}^{\delta_i} \alpha_j^i \lambda_j^i v_j^i)(C\overline{F}_i)^+ + K_i(I_p - (C\overline{F}_i)(C\overline{F}_i)^+) \qquad i = 1 \ldots r \tag{27}$$

where K_i ($i = 1 \ldots r$) is a parametrization of K associated to each $\overline{F}_i$. Substituting (27) in to equation (26.1) gives [12]

$$\overline{A} - KC = \overline{A}_i - K_i C_i \qquad i = 1 \ldots r \tag{28.1}$$

with

$$\overline{A}_i = \overline{A}\,(I_n - \overline{F}_i(C\overline{F}_i)^+ C) \text{ and } C_i = (I_p - (C\,\overline{F}_i)\,(C\,\overline{F}_i)^+)\,C \qquad i = 1 \ldots r \tag{28.2}$$

Note that $C_i\,\overline{F}_i = 0$, then the detection space is orthogonal on the observable space of $(\overline{A}_i, C_i)$ (see (20) and (21)). Then the detection space associated to the fault $f_i(t)$ can be defined as the space spanned by eigenvectors corresponding to the unobservable modes of the pair $(\overline{A}_i, C_i)$.

Definition 2 : [12] The dimension δ_i of the detection space associated to $f_i(t)$ is given by

$$\delta_i = n - \mathrm{rank}\ M_i \tag{29.1}$$

where

$$M_i^T = \begin{bmatrix} C_i^T & (C_i\overline{A}_i)^T & \cdots & (C_i\overline{A}_i^{n-1})^T \end{bmatrix} \tag{29.2}$$

the null space of $\mathbf{M}_i$ spans the detection space of f_i. ❐

The detection space of $f_i(t)$ is the H-invariant subspace of the error variable state which represents that part of the system affected by $f_i(t)$. This H-invariance property implies that the controllable space of $f_i(t)$ with respect to H, called W_i, is the smallest H-invariant containing $\overline{F}_i$ and is defined as follows.

Definition 3 : The smallest H-invariant containing $\overline{F}_i$ is the controllable space of $\overline{F}_i$ with respect to H and is given by

$$W_i = \begin{bmatrix} \overline{F}_i & H\overline{F}_i & \ldots & H^{n-1}\overline{F}_i \end{bmatrix} \tag{30}$$ ❐

4.1.3- Output separability and mutual detectability

In the following, we must verify if there is no overlap between the detection spaces M_i ($i = 1 \ldots r$).

Definition 4 : [12] Faults $f_i(t)$ ($i = 1 \ldots r$) are separable in the output space if

$$\mathrm{rank}\ C\ \overline{F} = \mathrm{rank}\ \overline{F} \tag{31}$$ ❐

If this definition is not satisfied then there exist some vectors w_i satisfying (22.2) which are collinear and the output spaces overlap. Then the faults can not be separated in the output space.

Lemma 1 : [12] If the failure vectors in $\overline{F}$ are output separable, then the detection space of $f_i(t)$'s are pairwise independent ■

Now we study the mutual detectability of failure modes $f_i(t)$'s which is related to condition (i) of definition 1 [12]. To do that, the detection space associated to the matrix $\bar{F}$ is determined in a manner analogous to the case of vector $\bar{F}_i$. Equation (26.2) can be rewritten as [12]

$$KC\bar{F} = Q_d \qquad (32.1)$$

where the ith column of Q_d, called Q_{di}, is given by

$$Q_{di} = \bar{A}\,\bar{F}_i - \sum_{j=1}^{\delta_i} \alpha_j^i \lambda_j^i v_j^i \qquad (32.2)$$

The solution of (32.1) is given by $\quad K = Q_d(C\bar{F})^+ + \bar{K}(I_p - (C\bar{F})(C\bar{F})^+) \qquad (33)$

and the equation (26.1) can be rewritten as $\quad \bar{A} - KC = \bar{\mathcal{A}} - \bar{K}\,\bar{C} \qquad (34.1)$

where $\quad \bar{\mathcal{A}} = \bar{A} - Q_d\,(C\bar{F})^+ C$ and $\bar{C} = (I_p - (C\bar{F})(C\bar{F})^+) \qquad (34.2)$

Then the observability matrix of the pair $(\bar{\mathcal{A}}, \bar{C})$ is given by

$$M^T = \left[\bar{C}^T \quad (\bar{C}\,\bar{\mathcal{A}})^T \quad \ldots \quad (\bar{C}\,\bar{\mathcal{A}}^{n-1})^T\right] \qquad (35)$$

Using fact 1, the dimension of the detection space associated to $(\bar{\mathcal{A}}, \bar{C})$ is

$$\delta = n - \text{rank}\, M \le k \qquad (36)$$

The following lemma gives condition of mutual detectability.

Lemma 2 : [12] The failure modes $f_i(t)$ $(i = 1 \ldots r)$ are separable in the output space, then they are mutually detectable if and only if

$$\sum_{i=1}^{r} \delta_i = \delta \qquad (37)■$$

Using fact 1, two cases may arise : (a) $\delta + \delta_v = k$ and (b) $k - \delta - \delta_v = \delta_c > 0$. In case (a), the k eigenvalues associated to Im(S) are determined : δ eigenvalues are chosen in order to satisfy condition (i) of definition 1 and the δ_v remaining eigenvalues are the invariant zeros of (A, Φ, C). In case (b), δ_c additional eigenvalues must be assigned. Now, we consider the case (b) : the problem can be solved by choosing δ_c vectors F_{ci} $(i = 1 \ldots \delta_c)$ such that, if the vectors F_{ci} are treated in the same way that the vectors F_i, one obtains $\delta + \delta_v = k$. Then, the problem of designing the filter K can be treated by considering the case (a). Then the detection of the fault component $f_i(t)$ is solved by multiplying the residual vector r(t) (3) by a projection into Im(C W_i) [12]. Note that, by construction, one has [12]

$$\text{rank}\, C\,W_i = 1$$

4.2- Detection gain design

This section is dedicated to the design of the filter gain K by considering the case (a). First consider a non singular matrix π such that

$$\Phi\,\pi = [\ \varphi \quad 0\ ] \qquad (38)$$

where the matrix φ is injective ($\varphi \in \mathbb{R}^{n\times(n-k)}$ with rank $\varphi = n - k$) and denotes φ_i the ith column of φ.

Then the problem of designing the gain K is reduced to choose the eigenvalues λ_j^i $(i = 1 \ldots n{-}k{+}r$ and $j = 1 \ldots \mu_i$ with $\mu_i = \delta_i$ if $i \le r$ and $\mu_i = 1$ if $i > r$) and to solve simultaneously equations (22) by replacing matrix $\bar{F}$ by $\bar{F}_\varphi$ where

$$\bar{F}_\varphi = \left[\bar{F} \quad \varphi\right] \qquad (39)$$

Denoting $\hat{F}_i$ the ith column of $\bar{F}_\varphi$, equation (22) is replaced by

$$\begin{bmatrix} \lambda_j^i I_n - \bar{A} & K \\ C & 0 \end{bmatrix} \begin{bmatrix} v_j^i \\ w_i \end{bmatrix} = \begin{bmatrix} 0 \\ w_i \end{bmatrix} \qquad i = 1 \ldots n{-}k{+}r \qquad (40.1)$$

with

$$w_i = C v_j^i = C\hat{F}_i \text{ and } \hat{F}_i = \sum_{j=1}^{\mu_i} \alpha_j^i v_j^i \qquad i = 1 \ldots n{-}k{+}r \qquad (40.2)$$

The set of eigenvectors v_j^i verifying equations (40.2) is given by

$$v_j^i = C^+ w_i + \bar{C}\ \beta_j^i \qquad (41)$$

with $\bar{C} = (I_n - C^+ C)$. The design parameter β_j^i is introduced in order to obtain vectors v_j^i who verify equation (40.2). In the sequel, the indices i and j varies as $i \in [1, n{-}k{+}r]$ and $j \in [1, \mu_i]$. Substituting v_j^i in equation (40.1), we obtain

$$Kw_i = -\bar{A}_j^i\ C^+ w_i - \bar{A}_j^i\ \bar{C}\ \beta_j^i \qquad (42)$$

where

$$\bar{A}_j^i = \lambda_j^i I_n - \bar{A}$$

Applying the vec operator into equation (42) [7]

$$\text{vec}(K\,w_i) = -\,\text{vec}(\bar{A}_j^i\ C^+ w_i) - \text{vec}(\bar{A}_j^i\ \bar{C}\ \beta_j^i) \qquad (43)$$

we obtain

$$(w_i^T \otimes I)\,\text{vec}(K) = -\,\text{vec}(\bar{A}_j^i\ C^+ w_i) - (\bar{A}_j^i\ \bar{C})\,\text{vec}\,(\beta_j^i) \qquad (44)$$

Let be $Y_i = w_i^T \otimes I_n,\ Z_{ij} = \mathrm{vec}(\overline{A}_j^i\ C^+ w_i),\ X_{ij} = \overline{A}_j^i\ \tilde{C}$

Then equation (44) becomes $Y_i\,\mathrm{vec}(K) + X_{ij}\,\mathrm{vec}(\beta_j^i) = -Z_{ij}$

and can be written in a matrix form as follows
$$\tilde{X}\ \overline{V} = -\overline{Z} \tag{45.1}$$

with
$$\overline{X} = \mathrm{diag}(X_{ij}),\ \overline{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_{n-k+r} \end{bmatrix},\ \tilde{X} = \begin{bmatrix} \overline{Y} & \overline{X} \end{bmatrix},\ \overline{Z} = \begin{bmatrix} \vdots \\ Z_{ij} \\ \vdots \end{bmatrix} \text{ and } \overline{V}^T = \begin{bmatrix} \mathrm{vec}(K)^T & \cdots & \mathrm{vec}(\beta_j^i)^T & \cdots \end{bmatrix} \tag{45.2}$$

(45) has a solution if and only if
$$\mathrm{rank}\ \tilde{X} = \mathrm{rank}\begin{bmatrix} \overline{X} & \overline{Y} \end{bmatrix} \tag{46}$$

If condition (46) is satisfied, e.i. if the faults are output separable and mutually detectable, then the solution of equation (45) is given by
$$\overline{V} = -\tilde{X}^+\,\overline{Z} + (I - \tilde{X}^+\,\tilde{X})\,\eta \tag{47}$$
where η is an arbitrary vector with appropriate dimension.

We will determine η and then β_j^i, in order to verify condition (40.2). In equation (47), we define

i) the matrix ρ_j^i given by the rows of the matrix $(I - \tilde{X}^+\tilde{X})$ corresponding to β_j^i in the vector $\overline{V}$,

ii) the vector γ_j^i given by the n coordinates of the vector $\tilde{X}^+\overline{Z}$ corresponding to β_j^i in the vector $\overline{V}$.

Then equation (41) becomes
$$v_j^i = C^+ w_i - \tilde{C}\gamma_j^i + \tilde{C}\rho_j^i\eta_j^i \tag{48.1}$$

where
$$\eta_i{}^T = \begin{bmatrix} \eta_1^i{}^T & \cdots & \eta_{\mu_i}^i{}^T \end{bmatrix} \text{ and } \eta^T = \begin{bmatrix} \eta_1{}^T & \cdots & \eta_{n-k+r}{}^T \end{bmatrix} \tag{48.2}$$

From (48.2), the constraint (40.2) can be written as
$$\hat{F}_i = \sum_{j=1}^{\mu_i} \alpha_j^i (C^+ w_i - \tilde{C}\gamma_j^i + \tilde{C}\rho_j^i\eta_j^i) \tag{49}$$

and the jth column of V_i is given by
$$V_i^j = C^+ w_i - \tilde{C}\gamma_j^i \tag{50}$$

If the following condition holds
$$\hat{F}_i \in \mathrm{Im}(V_i) \tag{51}$$

then we have
$$\eta_j^i = 0 \tag{52}$$

else, we decompose the vector $\hat{F}_i$ as
$$\hat{F}_i = \hat{F}_{i1} + \hat{F}_{i2} \tag{53.1}$$

where
$$\hat{F}_{i1} \in \mathrm{Im}(V_i) \text{ and } \hat{F}_{i2} \notin \mathrm{Im}(V_i) \tag{53.2}$$

To satisfy (53.2), we can choose $\hat{F}_{i1}$ and $\hat{F}_{i2}$ as
$$\hat{F}_{i1} = V_i^+ F_i \text{ and } \hat{F}_{i2} = \hat{F}_i - \hat{F}_{i1} \tag{54}$$

Define the matrix Ω as
$$\Omega = \tilde{C}\begin{bmatrix} \rho_1^i & \dots & \rho_j^i & \dots & \rho_{\mu_i}^i \end{bmatrix} \tag{55}$$

then the vectors $\hat{F}_i$ verify the constraint (40.2), or equivalently the relation (49), if the following relations hold
$$\hat{F}_{i2} \in \mathrm{Im}\ (\Omega) \tag{56}$$

The vector η_i in (47) and (48.2) is given by
$$\eta_i = \Omega^+\hat{F}_{i2} \tag{57}$$

Thus the vectors β_j^i and v_j^i and the gain K are solution to equation (41).

5- Conclusion

In this paper the detection filter for bilinear systems with unknown inputs is formulated and interpreted as an eigenspace problem, by means of an eigenvalue-eigenvector assignment technique. The detection gain is designed such that the residual is decoupled from the system perturbations.

References

1 G. Basile, G. Marro, Controlled and Conditioned Invariants in Linear System Theory, Prentice Hall, 1992.

2 R.V. Beard, Failure Accommodation in Linear Systems Through Selfreorganzation, Ph.D., Mass. Inst. Tech., Cambridge, M A, 1971.

3 M. Darouach, M. Zasadzinski, S.J. Xu, "Full-order observers for linear systems with unknown inputs", IEEE Tr. Aut. Cont., 39, 606-609, 1994.

4 Y. Funahashi, "Stable state estimator for bilinear systems", Int. J. Cont., 29, 181-188, 1979.

5 S. Hara, K. Furuta, "Minimal order state observers for bilinear systems", Int. J. Cont., 24, 705-718, 1976.

6 P. Kudva, N. Viswanadham, A. Ramakrishna, "Observers for linear systems with unknown inputs", IEEE Tr. Aut. Cont., 25, 113-115, 1980.

7 P. Lancaster, M. Tismenetsky, The Theory of Matrices, Academic Press, 1985.

8 C. Mechmeche, M. Zasadzinski, M. Darouach, H. Rafaralahy, "On unknown input observers for bilinear systems", American Control Conference, Seattle, 1995.

9 C. Mechmeche , Nowakowski S., Darouach M., "A failure detection and isolation procedure for bilinear systems based on a new formulation of unknown inputs bilinear observers", IFAC SAFEPROCESS'94, Helsinki, Finland, 1994.

10 R. Mohler, Nonlinear Systems, Application to Bilinear Control, Vol II, Prentice-Hall, 1991.

11 J. Park, G. Rizzoni, "A new interpretation of the fault detection filter (part 1)", Int. J. Cont., 60, 767-787, 1994.

12 J.E. White, J.L. Speyer, "Detection Filter Design: Spectral Theory and Algorithms", IEEE Tr. Aut. Cont., 32, 593-603, 1987.

WAVELETS IN OPTIMISATION AND APPROXIMATIONS

A. N. Fedorova and M. G. Zeitlin
Russian Academy of Sciences, Institute of Problems of Mechanical Engineering, Lab. of Mathematical Models of Mechanics, Russia, 199178, St. Petersburg, V.O., Bolshoj pr. 61.
E-mail: zeitlin@math.ipme.ru, anton@math.ipme.ru

Abstract. We give the explicit time description of four the following problems: dynamics and optimal dynamics for some important electromechanical system, Galerkin approximation for beam equation, computations of Melnikov function for perturbed Hamiltonian systems. All these problems are reduced to the problem of the solving of the systems of differential equations with polynomial nonlinearities and with or without some constraints. The first main part of our construction is some variational approach to this problem, which reduces initial problem to the problem of the solution of functional equations at the first stage and some algebraical problems at the second stage. We consider also two private cases of our general construction. In the first case (particular) we have the solution as a series on shifted Legendre polynomials, which is parameterized by the solution of reduced algebraical system of equations. In the second case (general) we have the solution in a compactly supported wavelet basis. Multiresolution expansion is the second main part of our construction. The solution is parameterized by solutions of two reduced algebraical problems, the first one is the same as in the first case and the second one is some linear problem, which is obtained from anyone of the next wavelet construction: Fast Wavelet Transform, Stationary Subdivision Schemes, the method of Connection Coefficients.

We give the explicit time description of the following problems: dynamics and optimal dynamics for nonlinear dynamical systems and Galerkin approximation for some class of partial differential equations, computations of Melnikov function for perturbed Hamiltonian systems. All these problems are reduced to the problem of the solving of the systems of differential equations with polynomial nonlinearities with or without some constraints. The first main part of our construction is some variational approach to this problem, which reduces initial problem to the problem of the solution of functional equations at the first stage and some algebraical problems at the second stage. We consider also two private cases of our general construction.

In the first case (particular) we have for Riccati type equations the solution as a series on shifted Legendre polynomials, which is parametrized by the solution of reduced algebraical (also Riccati) system of equations [1]–[5].

In the second case (general polynomial systems) we have the solution in a compactly supported wavelet basis [6]–[8]. Multiresolution expansion is the second main part of our construction. In this case the solution is parametrized by solutions of two reduced algebraic problems, one as in the first case and the second is some linear problem, which is obtained from one of the next wavelet construction: Fast Wavelet Transform (FWT) [9], Stationary Subdivision Schemes (SSS) [10], the method of Connection Coefficients (CC) [11].

We use our general construction for solution of important technical problems: minimization of energy and detecting signals from oscillations of a submarine.

Our initial problem comes from very important technical problem – minimization of energy in electromechanical system with enormous expense of energy. That is synchronous drive of the mill–the electrical machine with the mill as load. It is described by Park system of equations [1]–[2]:

$$\begin{aligned}
\frac{di_1}{dt} &= A_{11}i_1 + A_{12}i_2i_6 + A_{13}i_3 + A_{14}i_4 + A_{15}i_5i_6 + A_1(t)\\
\frac{di_2}{dt} &= A_{21}i_1i_6 + A_{22}i_2 + A_{23}i_3i_6 + A_{24}i_4i_6 + A_{25}i_5 + A_2(t)\\
\frac{di_3}{dt} &= A_{31}i_1 + A_{32}i_2i_6 + A_{33}i_3 + A_{34}i_4 + A_{35}i_5i_6 + A_3(t)\\
\frac{di_4}{dt} &= A_{41}i_1 + A_{42}i_2i_6 + A_{43}i_3 + A_{44}i_4 + A_{45}i_5i_6 + A_4(t)\\
\frac{di_5}{dt} &= A_{51}i_1i_6 + A_{52}i_2 + A_{53}i_3i_6 + A_{54}i_4i_6 + A_{55}i_5 + A_5(t)
\end{aligned}$$

$$\frac{di_6}{dt} = A_{61}i_1i_2 + A_{62}i_1i_5 + A_{63}i_2i_3 + A_{64}i_2i_4 + A_6(i_6,t),$$

where $A_{ij}, (i,j = \overline{1,6})$ are constants, $A_i(t)$ explicit functions of time,
$A_6(i_6,t) = a + di_6 + bi_6^2$ is analytical approximation for the mechanical moment of the mill. In our case we consider i_1, i_2 as the controlling variables. Because we consider the energy optimization, we use the next general
form of energy functional in our electromechanical system

$$Q = \int_{t_0}^{t} [K_1(i_1,i_2) + K_2(\dot{i}_1,\dot{i}_2)]dt,$$

where K_1, K_2 are quadratic forms. Moreover, we consider the optimization problem with some constraints which are motivated by technical reasons. After the manipulations from the theory of optimal control, we reduce the problem of energy minimization to the some nonlinear system of equations [1]. Thus for the Lagrangian optimization we have the system of 13 equations (12 - differential equations, 1-functional one). For the Hamiltonian optimization we have the system of 12 equations (10-differential equations, 2-algebraic ones). In both cases obtained systems of equations are the systems of Riccati type. As result of solution of equations of optimal dynamics we have:

1. the explicit time dependence of the controlling variables
   $u(t) = \{i_1(t), i_2(t)\}$ which give
2. the optimum of corresponding functional of energy and
3. explicit time dynamics of the controllable variables $\{i_3, i_4, i_5, i_6\}(t)$.

Next we consider the construction of explicit time solution. The obtained solutions are given in the next form:

$$i_k(t) = i_k(0) + \sum_{i=1}^{N} \lambda_k^i X_i(t),$$

where in our first case we have $X_i(t) = Q_i(t)$, where $Q_i(t)$ are shifted Legendre polynomials [12] and λ_k^i are roots of reduced algebraic system of equations. In wavelet case $X_i(t)$ correspond to multiresolution expansions in the base of compactly supported wavelets and λ_k^i are the roots of corresponding algebraic Riccati systems with coefficients, which are given by FWT, SSS or CC constructions. According to the variational method of [12] to give the reduction from differential to algebraical system of equations we need compute the objects $\gamma_a^j(i_b)$ and μ_{ji}, where in Lagrangian case $a = \overline{1,13}, b = \overline{1,13}$. We compute it by the formulae:

$$\begin{aligned}\gamma_a^j(i_b) &= t_f \int_0^1 \phi_a(i_b,\tau)X_j(\tau)d\tau \\ \mu_{ji} &= \int_0^1 X_i'(\tau)X_j(\tau)d\tau,\end{aligned}$$

where ϕ_a is RHS of initial equations. Then the reduced algebraical system has the form:

$$\sum_{i=1}^{N} \mu_{ji}\lambda_a^i - \gamma_a^j(\lambda_b) = 0$$

where coefficients of algebraical systems are constructed from objects:

$$\begin{aligned}\sigma_i &\equiv \int_0^1 X_i(\tau)d\tau = (-1)^{i+1}, \\ \nu_{ij} &\equiv \int_0^1 X_i(\tau)X_j(\tau)d\tau = \sigma_i\sigma_j + \frac{\delta_{ij}}{(2j+1)},\end{aligned}$$

$$\begin{aligned}
\beta_{klj} &\equiv \int_0^1 X_k(\tau)X_l(\tau)X_j(\tau)d\tau = \sigma_k\sigma_l\sigma_j + \alpha_{klj} + \frac{\sigma_k\delta_{jl}}{2j+1} + \frac{\sigma_l\delta_{kj}}{2k+1} + \frac{\sigma_j\delta_{kl}}{2l+1}, \\
\alpha_{klj} &\equiv \int_0^1 X_k^* X_l^* X_j^* d\tau = \frac{1}{(j+k+l+1)R\big(1/2(i+j+k)\big)} \times \\
&\quad R\big(1/2(j+k-l)\big)R\big(1/2(j-k+l)\big) \times R\big(1/2(-j+k+l)\big),
\end{aligned}$$

if $j+k+l=2m, m \in Z$,and $\alpha_{klj}=0$ if $j+k+l=2m+1$; where $R(i)=(2i)!/(2^i i!)^2$, $X_i=\sigma_i+X_i^*$, where the second equality in the formulae for $\sigma, \nu, \beta, \alpha$ hold for the first case.

Now we give construction for their computations in the wavelet case.

We use compactly supported wavelet basis: orthonormal basis for functions in $L^2(\mathbf{R})$ [13]. As usually $\varphi(x)$ is a scaling function, $\psi(x)$ is a wavelet function, where $\varphi_i(x)=\varphi(x-i)$. Scaling relation that defines φ, ψ are

$$\begin{aligned}
\varphi(x) &= \sum_{k=0}^{N-1} a_k \varphi(2x-k) = \sum_{k=0}^{N-1} a_k \varphi_k(2x), \\
\psi(x) &= \sum_{k=-1}^{N-2} (-1)^k a_{k+1} \varphi(2x+k)
\end{aligned}$$

Let be $f: \mathbf{R} \longrightarrow \mathbf{C}$ and the wavelet expansion is

$$f(x) = \sum_{\ell \in \mathbf{Z}} c_\ell \varphi_\ell(x) + \sum_{j=0}^{\infty} \sum_{k \in \mathbf{Z}} c_{jk} \psi_{jk}(x)$$

The indices k, j represent translation and scaling

$$\begin{aligned}
\varphi_{jl}(x) &= 2^{j/2}\varphi(2^j x - \ell) \\
\psi_{jk}(x) &= 2^{j/2}\psi(2^j x - k)
\end{aligned}$$

If $c_{jk}=0$ for $j \geq J$, then $f(x)$ has an alternative expansion in terms of dilated scaling functions only

$$f(x) = \sum_{\ell \in \mathbf{Z}} c_{J\ell} \varphi_{J\ell}(x)$$

This is a finite wavelet expansion, it can be written solely in terms of translated scaling functions. We use wavelet $\psi(x)$, which has k vanishing moments

$$\int x^k \psi(x) d(x) = 0, \qquad 0 \leq k \leq K$$

or, equivalently

$$x^k = \sum c_\ell \varphi_\ell(x) \text{for each } k, \quad 0 \leq k \leq K$$

Also we have the shortest possible support: scaling function DN (where N is even integer) will have support $[0, N-1]$ and $N/2$ vanishing moments.

There exists $\lambda > 0$ such that DN has λN continuous derivatives; for small $N, \lambda \geq 0.55$. To solve our second associated linear problem we need to evaluate derivatives of $f(x)$ in terms of $\varphi(x)$.

Let be $\varphi_\ell^n = d^n \varphi_\ell(x)/dx^n$. We derive the wavelet - Galerkin approximation of a differentiated $f(x)$ as $f^d(x) = \sum_\ell c_l \varphi_\ell^d(x)$ and values $\varphi_\ell^d(x)$ can be expanded in terms of $\varphi(x)$

$$\begin{aligned}
\phi_\ell^d(x) &= \sum_m \lambda_m \varphi_m(x), \qquad \text{where} \\
\lambda_m &= \int\limits_{-\infty}^{\infty} \varphi_\ell^d(x) \varphi_m(x) dx
\end{aligned}$$

The coefficients λ_m are 2-term connection coefficients [11]. In general we need to find

$$\Lambda(\ell_1, \ell_2, ..., \ell_n, d_1, d_2, ..., d_n) = \Lambda^{d_1 d_2 \ldots d_n}_{\ell_1 \ell_2 \ldots \ell_n} = \int\limits_{-\infty}^{\infty} \prod \varphi^{d_i}_{\ell_i}(x)dx$$

For our Riccati case we need to evaluate two and three connection coefficients

$$\begin{aligned} \Lambda^{d_1 d_2}_{\ell} &= \int_{-\infty}^{\infty} \varphi^{d_1}(x)\varphi^{d_2}_{\ell}(x)dx, \quad d_i \geq 0, \\ \Lambda^{d_1 d_2 d_3} &= \int\limits_{-\infty}^{\infty} \varphi^{d_1}(x)\varphi^{d_2}_{\ell}(x)\varphi^{d_3}_{m}(x)dx \end{aligned}$$

According to [11] we use the next construction. When N in scaling equation is a finite even positive integer the function $\varphi(x)$ has compact support contained in $[0, N-1]$. For a fixed triple (d_1, d_2, d_3) only some $\Lambda^{d_1 d_2 d_3}_{\ell m}$ are nonzero : $2-N \leq \ell \leq N-2, \quad 2-N \leq m \leq N-2, \quad |\ell - m| \leq N-2$. There are $M = 3N^2 - 9N + 7$ such pairs (ℓ, m). Let $\Lambda^{d_1 d_2 d_3}$ be an M-vector, whose components are numbers $\Lambda^{d_1 d_2 d_3}_{\ell m}$. Then we have the first key result: Λ satisfy the system of equations

$$\begin{aligned} A\Lambda^{d_1 d_2 d_3} &= 2^{1-d}\Lambda^{d_1 d_2 d_3}, \ d = d_1 + d_2 + d_3, \\ A_{\ell,m;q,r} &= \sum_p a_p a_{q-2\ell+p} a_{r-2m+p} \end{aligned}$$

By moment equations we have created a system of $M + d + 1$ equations in M unknowns. It has rank M and we can obtain unique solution by combination of LU decomposition and QR algorithm. The second key result gives us the 2-term connection coefficients:

$$\begin{aligned} A\Lambda^{d_1 d_2} &= 2^{1-d}\Lambda^{d_1 d_2}, \quad d = d_1 + d_2, \\ A_{\ell,q} &= \sum_p a_p a_{q-2\ell+p} \end{aligned}$$

Also, we use FWT and SSS for computing coefficients of reduced algebraic systems. We use for modelling D6,D8,D10 functions and programs RADAU and DOPRI for testing [14].

As a result we obtained the explicit time solution of optimal control problem. The generalization to polynomial systems is evidently. Analogously we consider in wavelet approach related problems: computations in Galerkin approximations and routes to chaos in Melnikov approach [6], [15]. These problems are related to a problem of detecting signals from an oscillating submarine.

In 2-mode Galerkin approximation for beam contacting with ideal compressible liquid in a channel we have the next system of equations [16]:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -ax_1 - b[\cos(x_5) + \cos(x_6)]x_1 - dx_1^3 - mdx_1x_3^2 - px_2 - \varphi(x_5) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= ex_3 - f[\cos(x_5) + \cos(x_6)] - gx_3^3 - kx_1^2x_3 - gx_4 - \psi(x_5) \\ \dot{x}_5 &= r \\ \dot{x}_6 &= s \end{aligned}$$

or in Hamiltonian form

$$\begin{aligned} \dot{x} &= J \cdot \nabla H(x) + \varepsilon g(x, \Theta), \\ \dot{\Theta} &= \omega, \quad (x, \Theta) \in R^4 \times T^2, \quad T^2 = S^1 \times S^1, \end{aligned}$$

for $\varepsilon = 0$ we have:

$$\dot{x} = J \cdot \nabla H(x), \quad \dot{\Theta} = \omega$$

We solve two problems related with these systems of equations. We need to compute explicit time solution for perturbed and unperturbed systems. The solution for unperturbed system we use next for computing Melnikov functions

$$M(\Theta) = \int_{-\infty}^{\infty} \nabla H(\bar{x}_0(t)) \wedge g(\bar{x}_0(t), \omega t + \Theta) dt$$

$$M^{m/n}(t_0) = \int_{0}^{mT} DH(x_\alpha(t)) \wedge (x_\alpha(t), t + t_0) dt$$

which we use for detecting chaotic and quasiperiodic regimes of oscillations [6],[15],[16]. In comparison with wavelet expansion on the real line which we use in optimal control problem and in calculation of Galerkin approximation, in Melnikov function approach we need to use periodized wavelet expansion, i.e. wavelet expansion on finite interval [17]. Also in the solution of perturbed system we have some problem with variable coefficients. For solving last problem we need to consider one more refinement equation for scaling function $\phi_2(x)$:

$$\phi_2(x) = \sum_{k=0}^{N-1} a_k^2 \phi_2(2x - k)$$

and corresponding wavelet expansion for variable coefficients $b(t)$:

$$\sum_k B_k^j(b) \phi_2(2^j x - k),$$

where $B_k^j(b)$ are functionals supported in a small neighborhood of $2^{-j}k$ [18].

The solution of the first problem consists in periodizing. In this case we expand homoclinic orbit or periodic orbit into periodized wavelets defined by [17]:

$$\phi^{per}_{-j,k}(x) = 2^{j/2} \sum_Z \phi(2^j x + 2^j \ell - k)$$

All these modifications lead only to transformations of coefficients of reduced algebraic system, but general scheme remains the same [15].

References

[1] Fedorova, A.N., Zeitlin, M.G. (1993) *Preprint IPME*, no. 96.

[2] Fedorova, A.N., Rubashev, G.M. and Zeitlin, M.G. (1990), *Electrichestvo (Electricity)*, in Russian, no. 6, p. 40.

[3] Fedorova, A.N, Zeitlin, M.G (1993), *Int. Congress on Computer Systems and Applied Mathematics CSAM'93*, p. 277, St. Petersburg.

[4] Fedorova, A.N., Zeitlin, M.G. (1994), *Asymptotic Methods in Mechanics*, p. 9, St. Petersburg.

[5] Fedorova, A.N., Zeitlin, M.G., (1995), *Proc. of 22 Summer School'Nonlinear Oscillations in Mechanical Systems'*, p. 89, St. Petersburg.

[6] Fedorova, A.N., Zeitlin, M.G. *Proc. of 23 Summer School 'Nonlinear Oscillations in Mechanical Systems'*, St. Petersburg, in press.

[7] Fedorova, A.N., Zeitlin ,M.G. (1995) *Optimisation of Finite Element Approximations*, p. 254, St. Petersburg.

[8] Fedorova, A.N., Zeitlin, M.G. (1995) *2-nd Russian- Swedish Control Conference*, p. 204, St. Petersburg.

[9] Beylkin, G., Coifman, R. and Rokhlin, V. (1991), *Comm. Pure Appl.Math.*, **no. 44**, p. 141.

[10] Dahlke, S., Weinreich, I. (1993), *Constructive approximation*, no. **9**, p. 237.

[11] Latto, A., Resnikoff, H.L. and Tenenbaum, E. (1991, July), Aware Technical Report AD910708.

[12] Hitzl, D.L., Huynh , T.V, Zele, F. (1984), *Physics Letters*, **Vol. 104, no. 9**, p. 447.
Hitzl, D.L. (1980), *J. of Computational Physics*, **Vol. 38, no. 2**, p. 185.

[13] Daubechies, I. (1988), *Comm.Pure Appl.Math.*, **no. 41**, p. 906.

[14] Hairer, E., Lubich, C., Roche, M. (1989), *Lecture Notes in Mechanics*, **Vol. 1409**.

[15] Fedorova, A.N., Zeitlin M.G., in press.

[16] Abramian, A.K., Fedorova, A.N., Zeitlin, M.G. (1995), *Proc.of 22 Summer School 'Nonlinear Oscillations in Mechanical Systems'*, p. 103, St. Petersburg.

[17] Cohen, A., Daubechies, I., Vial, P. (1993), *Wavelets on the interval and fast wavelet transforms*, preprint.

[18] Dahmen, W., Micchelli, C.A. (1993), *SIAM J. Numer. Anal.*, **Vol. 30, no. 2**, p.507.

MATHEMATICAL MODELS OF DISCRETE SELF-SIMILARITY

F.M. Borodich
Department of Mathematics, Glasgow Caledonian University,
Glasgow G4 0BA, United Kingdom

Abstract. Natural phenomena which exhibit discrete self-similarity are under consideration. We argue that a new concept of parametric-homogeneity can be helpful in the modelling of similarity of such non-smooth phenomena. Although some discrete self-similar phenomena were studied earlier there was however a gap in this field. Recently we tried to fill this gap by concentrating on the study of parametric-homogeneous (PH) and parametric-quasi-homogeneous (PQH) functions based on the use of discrete group of coordinate dilations. Here we consider some models of natural phenomena which have PH-features and discuss some properties of PH-functions. As an example of practical usage of these functions we consider the phenomenon of seismic activation prior to a major earthquake.

Introduction

The group of coordinate dilations is a mathematical tool of especial importance for describing the symmetry of natural phenomena. Dimensional analysis and the concept of self-similarity in the usual sense, which are used in all branches of physics and mechanics, are based on the use of this group. The concept of group of coordinate dilations is closely connected to the concepts of homogeneous and quasi-homogeneous functions. These functions are often called self-similar because they make it possible to reduce searching for a function of n-variables to searching for a function of $n-1$-variables. It follows from the definition that these functions are smooth functions of arguments. To model some types of non-smooth or discrete self-similarity it is necessary to use other kinds of functions.

Various attempts were made to find some universal parameters characterising self-similar properties of a phenomenon under consideration and use these parameters to model its behaviour. In particular, this tendency appeared in the use of the fractal geometry approach. However, it was proved that the fractal dimension alone was not sufficient to model the main physical properties of the phenomenon. Another approach was based on the use of smooth log-periodic functions for modelling the global trend of a self-similar process. Parametric-homogeneity, which includes parametric-homogeneous (PH) and parametric-quasi-homogeneous (PQH) functions, PH-sets and corresponding transformations, is a kind of discrete self-similarity which allows us to model both the long-term global trend of a self-similar process and its complex fractal structure. Indeed, log-periodicity is a particular kind of parametric-homogeneity and PH-functions often have fractal graphs.

In this paper we recall some basic mathematical properties of PH-functions and discuss several ways of constructing PH-functions of arbitrary degree. It is shown that such basic functions used in fractal geometry as the von Koch curve, the Weierstrass-Mandelbrot function and the Cantor staircase are examples of parametric-homogeneity. However, in general the fractal properties of objects and the property of parametric-homogeneity are independent of each other, i.e., the scaling law (PH-law) considered is not a kind of fractal scaling.

We consider some applications of the concept to problems of mechanics, physics, biology and some models often used for describing natural phenomena (logarithmic spiral, Sierpiński triangle, etc.) which have PH-features. As an example of practical usage of PH-functions we consider the phenomenon of seismic activation prior to a major earthquake.

Thus, we argue that the generalisation of classical self-similarity to the discrete case is very important for studies of various natural phenomena exhibiting threshold behaviour or phenomena with scaling near the critical point.

Some basic properties of parametric-homogeneity

We have introduced PH-functions and PQH-functions as natural generalisations of concepts of homogeneous and quasi-homogeneous functions and instead of the continuous group we consider the discrete group of coordinate dilations [4, 6]. The mathematical background of parametric-homogeneity was discussed in detail [6]. Here we discuss only some basic properties of parametric-homogeneity.

Let us recall the definition of PQH–functions [5, 6]. The function $B_d : \mathbb{R}^n \to \mathbb{R}$ is called a parametric–quasi–homogeneous function of degree d and parameter p with weights $\alpha = (\alpha_1, \dots, \alpha_n)$ if there exists a positive parameter p, $p \neq 1$ such that it satisfies the following identity

$$B_d(p^{k\alpha_1}x_1, ..., p^{k\alpha_n}x_n; p) = p^{kd}B_d(\mathrm{x}; p), \quad k \in \mathbb{Z}.$$

Parametric-homogeneous functions [4, 6] are a particular case of PQH–functions when $\alpha_1 = ... = \alpha_n$. To avoid a non unique definition we will take as the parameter the least p, $p > 1$. We say that a set is a PH–set if its symmetry, i.e., a group of transformations of coordinates which conserves the set, is a discrete one–parameter group.

Evidently, the PH–functions of zero degree are the automorphic functions invariant with respect to the discrete group of coordinate dilations. Some examples of PH–functions are well-known, for example the classical Cantor function $C_{1/p}$ (Cantor staircase) on the interval $I \equiv [0, 1]$ with similarity factor $r = 1/p$. This function is also often called the devil's staircase. The $C_{1/p}$ on $[0, 1]$ exhibits the following parametric–homogeneous properties

$$C_{1/p}(p^k x) = 2^k C_{1/p}(x) = p^{kd}C_{1/p}(x), \quad k = 0, -1, -2, \dots, \quad d = \ln 2/\ln p.$$

To construct PH–functions we use the following algebraic concepts: orbit and fundamental domain of a group [6]. An orbit of an element $m \in M$ under a transformation group Γ of the set M is the set Γm, consisting of the elements of the form $\gamma(m)$, where γ runs through all elements of Γ. A fundamental domain for a group Γ is the set $\Omega(\Gamma) \subset M$ such that the orbit of every point $m \in M$ meets $\Omega(\Gamma)$ and the orbit of every interior point $m \in \Omega(\Gamma)$ only meets $\Omega(\Gamma)$ in m itself. Then different points of one orbit belonging to the closure of $\Omega(\Gamma)$ can only lie on the boundary of $\Omega(\Gamma)$.

Evidently, the necessary condition for building the fundamental domain of a group is the condition that the group is discrete (discontinuous), i.e., that the orbital equivalent points are discrete.

Let $h_f : (a, b] \to \mathbb{R}$ be an arbitrary function, $(a, b] \subset \mathbb{R}_+$. Then it is extended uniquely on all $\mathbb{R}_+$ as an automorphic function which is invariant with respect to the discrete group of coordinate dilations. Indeed, we can take the half–interval $(a, b]$ as the fundamental domain of discrete group Γ_{p^n}. Then, for any $x \in \mathbb{R}_+$ there exists a corresponding number $x^0 \in (a, b]$ such that x and x^0 belong to the same orbit. It follows from the definition of the fundamental domain that such an x^0 is unique.

Let us denote by $b_d(\mathrm{x}; p)$ a PH–function of degree d, which depends on a parameter p, and by $H_d(x)$ a homogeneous function of degree d. It is easy to show that for each $b_d : \mathbb{R}^n_+ \to \mathbb{R}$ there exist $H_d : \mathbb{R}^n_+ \to \mathbb{R}$ and $b_0 : \mathbb{R}^n_+ \to \mathbb{R}$ such that

$$b_d(\mathrm{x}; p) = H_d(\mathrm{x})b_0(\mathrm{x}; p).$$

If $n = 1$ then the above representation is unique up to a constant [4, 6].

Using the above representation, the PH–function $b_d : \mathbb{R}_+ \to \mathbb{R}$ is constructed in the following way

$$b_d(x; p) = Ax^d b_0(x; p), \quad b_0(x; p) = h_f(x^0),$$

where $\quad x^0 \in \gamma(x), \quad \Omega(\Gamma_{p^n}) = (a, b], \quad p = b/a, \quad A = const.$

We can also construct some PH–functions using another method. If h_p is a periodic function with period $\ln p$, then

$$b_d(x; p) = Ax^d h_p(\ln x), \quad x \in \mathbb{R}_+,$$

is a PH–function of degree d. This is the so-called log–periodicity.

The Weierstrass–type functions f are also well-known examples of PH–functions

$$f(x) = \sum_{n=-\infty}^{\infty} p^{-\beta n}h(p^n x),$$

especially the Weierstrass–Mandelbrot function [1] when $h(p^n x) = (1 - \cos p^n x)$ and the Takagi-Hopson function when $h(p^n x) = 2\left|p^n x - \left[p^n x + \frac{1}{2}\right]\right|$, where $[a]$ denotes the integer part of the number a. These functions are often used in applications. Evidently, the Weierstrass–type functions are PH–functions of degree $d = \beta$. It is known that the box dimension of the graph of f is $2 - \beta$. It is believed that the Hausdorff dimension of these functions is equal to their box dimension [8].

It is easy to show that if f is a Weierstrass–type function and h is a bounded function. Then the PH-function $b_d(x;p)$

$$b_d(x;p) = Ax^{d-\beta}f(x;p)$$

has the same Hausdorff dimension as the Weierstrass–type function f.

This is a consequence of the fact that the Hausdorff dimension of a set does not change under the action of the Lipschitz homeomorphism and any continuously differentiable function with a bounded derivative is necessarily a Lipschitz function [4, 6].

Discrete self-similarity of some physical phenomena

Clearly, real natural phenomena do not exhibit the pure mathematical PH-properties. However, PH-features can be exhibited by some processes on their intermediate stage when the behaviour of the processes has ceased to depend on the details of the boundary conditions or initial conditions. We can expect that some self–organised processes possess PH–features. It was often supposed in papers concerning applications of log–periodicity to critical phenomena (see, e.g., [10]) that if a system has a preferable scale factor (parameter) and exhibits a discrete dilation symmetry then it can only be described by log–periodic functions. However, as we have seen there are various PH–functions of other kinds.

Examples of PH-sets can be found everywhere because the discrete coordinate dilation symmetry is quite common in nature. An example of a PH–set is the logarithmic spiral. Indeed, magnifying the spiral by a factor λ shows the same spiral rotated by some angle. Clearly, we can choose such a λ so that this angle is 2π. Thus, the logarithmic spirals

$$r(\varphi) = r_0 \exp c(\varphi - \varphi_0), \quad -\infty < \varphi < \infty$$

where r_0, φ_0 and c are constants, possess PH–features, namely

$$r(\varphi + 2\pi n) = p^n r(\varphi)$$

where the parameter p is equal to $\exp(2\pi c)$. A ray with centre at the origin (polar coordinates) intersects the curve at discrete points r_n such that $r_{n+1}/r_n = p$. Therefore, if we consider a part of the logarithmic spiral

$$r(\varphi)/r_0 = e^{c\varphi}, \quad 0 \le \varphi < 2\pi$$

then we can obtain the whole spiral by PH–transformation.

The logarithmic spiral often arises in biology. The spiral structure can be found in numerous species of shellfish. Of course, there are other biological objects which exhibit PH-features. Another application of logarithmic spirals can be found in astronomy. Indeed, it is well-known that there are a number of spiral galaxies whose spiral arms can often be well fitted by logarithmic spirals.

The Sierpiński carpet and triangle, the von Koch curve and the Cantor set are examples of PH–sets. Indeed, if we consider a part of the Sierpiński carpet with a size equal to 3 without a square $0 \le x < 1,\ \ 0 \le y < 1$, then the whole carpet can be obtained by a PH–transformation. Similar constructions can be applied to the von Koch curve and the Cantor set.

The Liesegang rings can also be put within PH–sets. It is known that the so-called Liesegang ring experiment consists of diffusion of silver nitrate which is placed as a drop of aqueous solution onto a gel matrix containing potassium bichromate. During the experiment there arises a fragmented pattern consisting of bands (rings) of different widths. The formation of these Liesegang rings are explained by self–organisation. It was experimentally shown that the distances b_n between the successive n-th and $n+1$-th Liesegang rings increase following the law of geometric series $b_{n+1}/b_n = p, \quad p > 1$, in other words, the Liesegang ring patterns are PH–sets.

Let us now consider a particular case of PH-functions, namely log–periodic functions. Log–periodic functions are often called functions with complex scaling exponent (see, e.g., [12]) because the following sine log–periodic functions are usually used in applications

$$b_0(x;p) = A\sin\left(\frac{2\pi \ln x}{\ln p} + \Phi\right) + C, \qquad b_d(x;p) = Ax^d b_0(x;p),$$

where A, C and Φ are constants. For the complex exponent $d + i\eta$, $\eta = 2\pi n/\ln p$ we obtain

$$x^{d+i\eta} = x^d \exp(\ln x^{i\eta}) = x^d \left[\cos\left(\frac{2\pi n \ln x}{\ln p}\right) + i \sin\left(\frac{2\pi n \ln x}{\ln p}\right)\right].$$

It is clear that the real and imaginary parts of the function in square brackets are particular cases of the sine log-periodic functions.

The first area of applications of log-periodic functions to mechanics was in fluid dynamics [9, 10]. It seems to us that some recent results concerning turbulent flows [13] could also be explained using the concept of parametric-homogeneity.

The first area of the application of parametric-homogeneity was in contact mechanics [3]. Here the following result was obtained [3–6].

Let the distance between surfaces of contacting solids be determined by a positive PH-function of degree $d > 0$ and parameter p. In addition, let the operator of the constitutive relations F of body material be homogeneous functions of degree μ with respect to the components of the strain tensor ϵ_{ij}, i.e., for each positive λ we have $F(\lambda\epsilon_{ij}) = \lambda^\mu F(\epsilon_{ij})$. Then the three-dimensional Hertz boundary-value problem of contact the bodies loaded by the increasing force P is satisfied by the PQH-functions of variables (x_1, x_2, x_3, P) with weights $(1, 1, 1, \mu(d-1)+2)$.

Thus, the solution near any load P_0 is repeated in scaling form near all loads $p^{k[2+\mu(d-1)]}P_0$, $k \in \mathbb{Z}$. It is possible to get the whole solution using the results of numerical simulation of the problem on a finite interval of external load only.

Similar result and scaling can be obtained in the two-dimensional model of quasi-static discontinuous crack propagation when the extension of main fracture consists of a sequence of finite growth steps (stick-slip regime) [6].

Finally, let us consider some recent renormalization schemes of earthquake prediction. These schemes are based on long-distance correlations between earthquakes and seismic activity prior to a major earthquake. The seismic activity is attributed to an increase in the regional cumulative Benioff strain release (square root of energy) [7, 12].

The hypothesis that earthquakes are similar to critical points of phase transition phenomena is in common use. The critical point hypothesis supposes that a function f of some argument T diverges in the neighbourhood of critical value T_c as $f \sim |T - T_c|^{-\alpha}$, with $\alpha > 0$ [2]. The number α, which is called a critical exponent, describes the singular behaviour of interesting quantities. In application to earthquakes the hypothesis means that the regional cumulative Benioff strain release ϵ near the critical point (predicted time of failure) T_c is described by the following equation [7]

$$\frac{d\epsilon}{dT} = k|T - T_c|^{-\alpha} \qquad \text{or} \qquad \epsilon = K + A|T - T_c|^d,$$

where k, K, A are constants and $d = 1 - \alpha$ is some critical exponent.

Using the idea of a complex critical exponent, it has been shown [12] that there is an excellent fit to log-periodic increase in seismic activity and that experimental data [7] for the regional strain ϵ can be fitted by the following log-periodic function

$$\epsilon - K = A(T_c - T)^d \left[1 + C\cos\left(2\pi\frac{\log(T_c - T)}{\log p} + \Psi\right)\right],$$

better than the above simple power-law does. Here C, Ψ are constants. Figure 1 shows experimental data of cumulative Benioff strain released by earthquakes of magnitude 5 and greater in the San Francisco Bay region prior to the 1989 Loma Prieta earthquake [7]. Fitting these data by the simple power-law [7] it was obtained $T_c = 1990.0$.

Using the above smooth log-periodic approximation [12] it was obtained $T_c = 1989.9 \pm 0.8$ years which is closer to the actual value 1989.8. Figure 1 shows the graph of the above log-periodic approximation for the following parameters

$$p = 3.13; \quad \Psi = 1.45; \quad T_c = 1989.9; \quad K = 8.49; \quad A = -0.3; \quad d = 0.34; \quad C = -0.05.$$

It is unlikely that a process such as seismic activity has smooth character. Hence, the above smooth log-periodic function $\epsilon - K$ of the argument $(T_c - T)$ is only a particular case of PH-functions of degree

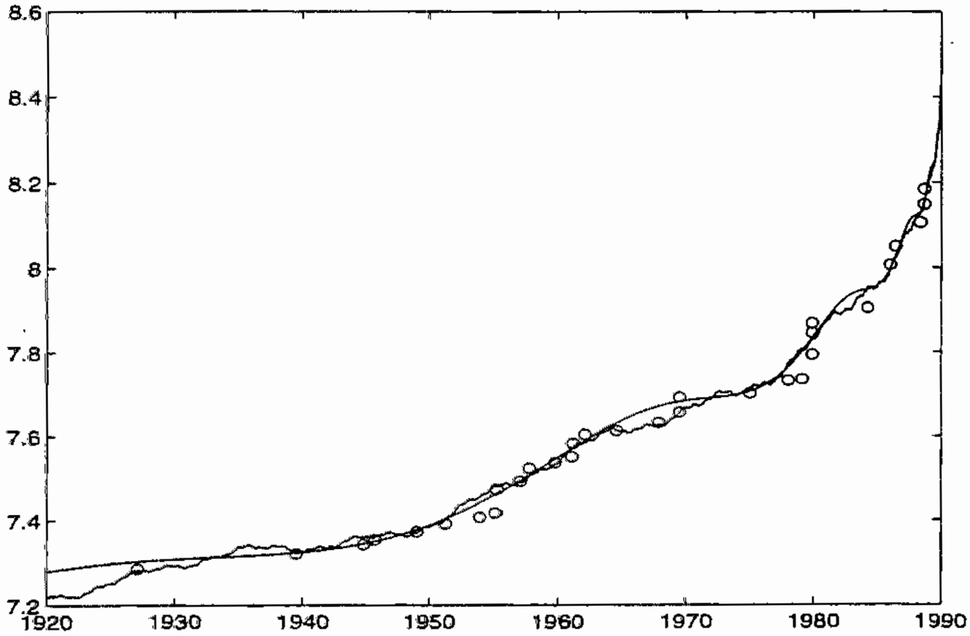


Figure 1: Comparison of log-periodic and PH-approximations of cumulative Benioff strain (in $10^8 (Nm)^{1/2}$) released by earthquakes of magnitude 5 and greater in the San Francisco Bay region.

d and parameter p which are suitable for an approximate description of the main trend of a PH-process. We can approximately model the process by another non-smooth PH-function of appropriate degree and parameter.

It is known that the spatial distributions of faults in the region of the San Andreas fault system and other regions of seismic activity have fractal structure (see, e.g., [11]). If fractal dimension is one of the universal parameters characterising the regional complexity then it could be expected that this dimension can be connected with the process's critical exponent. Relations between the process's critical exponents are known in renormalization group theory as scaling laws [2]. If we suppose that the process has fractal character then it could be expected that there is a scaling law between the fractal dimension of the regional fault system and the fractal dimension of the seismic process. It seems to us that in order to model both the long-term global trend of the fractal process and its complex fine structure it is worthwhile to use a Weierstrass type PH-function with the same parameter as in log-periodic modelling and the fractal dimension $2 - \beta$ concordant to the process's fractal dimension

$$\epsilon - K = A(T_c - T)^d \left[1 + C b_0(T;p)\right], \qquad b_0 = -(b_0^{(0)} - M)/A,$$

where $A = (\max b_0^{(0)} - \min b_0^{(0)})/2, \quad M = (\max b_0^{(0)} + \min b_0^{(0)})/2$ and

$$b_0^{(0)}(T;p) = (C_1(T_c - T))^{-\beta} \sum_{n=-\infty}^{\infty} p^{-\beta n}(1 - \cos(C_1 p^n (T_c - T))).$$

Here C_1 is a fitting constant.

Fitting the PH-function to experimental data and determining the moment T such that $\epsilon = K$ we obtain a prediction of the critical time T_c. Figure 1 shows the graph of the above fractal PH-approximation for the following parameters

$$p = 3.13; \quad T_c = 1989.9; \quad K = 8.48; \quad A = -0.3; \quad d = 0.34; \quad C = -0.05; \quad C_1 = 0.7; \quad \beta = 0.7.$$

Thus, if the physical hypothesis of long-distance correlations between earthquakes and seismic activity prior to a major earthquake, used in [7, 12] is valid then other kinds of PH-functions can predict the critical time to failure T_c and describe the experimental data at least with the same accuracy as log–periodic approximation does. In addition, fractal PH–functions allow us to take into account the fractal features of the process.

Conclusion

We have seen that the concept of parametric–homogeneity can be helpful in the modelling of discrete self–similarity of various cases of very interesting phenomena of solid mechanics, physics, biology, etc. We have seen that if we are interested in the main self–similar features of a process, namely in PH–properties, and the fine structure is not of interest, then we can approximately model the process by any PH–function of appropriate degree and parameter. In particular, we can describe it by the sine log–periodic function which is suitable for certain applications.

To model the fine structure we can use other PH–functions. In particular, PH–functions with fractal graphs because it is known that fractality gives a good description of statistical self–similarity of local structure of the phenomenon. However, in this case we lose the differentiability of the modelling. The above consideration showed that PH–functions of Weierstrass–type can lead to more accurate power laws and a better understanding of the seismic activity.

Acknowledgment. I am grateful to Mr. C.A. MacDonald (Glasgow) for technical help.

References

1. Berry, M.V. and Lewis, Z.V., On the Weierstrass-Mandelbrot fractal functions. Proc. R. Soc. Lond., A370 (1980), 519–521.

2. Binney, J.J., Dowrick, N.J., Fisher, A.J. and Newman, M.E.J., The Theory of Critical Phenomena. An Introduction to the Renormalization Group. Clarendon Press, Oxford, 1992.

3. Borodich, F.M., Similarity properties of discrete contact between a fractal punch and an elastic medium. C. r. Ac. Sc. (Paris), Ser. 2, 316 (1993), 281–286.

4. Borodich, F.M., Some applications of the fractal parametric–homogeneous functions. Fractals, 2 (1994), 311–314.

5. Borodich, F.M., Similarity in the problem of discrete contact between fractal surfaces. In: Fractal Reviews in the Natural and Applied Sciences, (Ed.: Novak, M.M.) Chapman and Hall, London, 1995, 113–120.

6. Borodich, F.M., Parametric–homogeneous functions, similarity, and fractal function graphs. Technical Report TR/MAT/FMB/95-35, Glasgow Caledonian University, Glasgow, 1995, 1–55.

7. Bufe, C. & Varnes, D.J. Predictive modeling of the seismic cycle of the greater San Francisco bay region, J. Geophys. Res. 98 (1993), 9871–9883.

8. Falconer, K.J., Fractal Geometry: Mathematical Foundations and Applications. Wiley, Chichester, New York, 1990.

9. Novikov, E.A., Mathematical model for the intermittence of turbulent flow. Sov. Phys. Dokl., 11 (1966), 497–499.

10. Novikov, E.A., The effects of intermittency on statistical characteristics of turbulence and scale similarity of breakdown coefficients. Phys. Fluids A, 2 (1990), 814–820.

11. Okubo, P.G. and Aki, K., Fractal geometry in the San Andreas fault system. J. Geophys. Res., 92 (1987), 345–355.

12. Sornette, D. and Sammis, C.G., Complex critical exponents from renormalization group theory of earthquakes: implications for earthquake predictions. J. Phys. I. France, 5 (1995), 607–619.

13. Ugalde, E., Self-similarity and finite-time intermittent effects in turbulent sequences. J. Phys. A. Math. Gen., 29 (1996), 4425–4443.

Computers as Living Systems

V. Gontar

International Group for Scientific and Technological Chaos Studies,
Ben-Gurion University of the Negev, P.O.B. 653, Beer-Sheva 84105, Israel

Abstract

The ability of computer programs to use non-linear equations with chaotic regimes to produce unpredictable information is discussed. Considering computers as artificially made "living systems" should open new ways of organization of computer systems by analogy with human societies. A numerical example for producing ornamental patterns from discrete chemical reaction dynamics is presented

Introduction

The new concept of "artificial life" is currently the object of intensive interest and study [1]. To find the means to combine modern knowledge of biological systems with that of artificially made systems—which have much in common in terms of functions but are very different in origin—constitutes one of the most interesting scientific problems facing the modern scientific thinker. The ultimate goal of much current research is to construct a theory that will finally give us answers to the age-old questions: Where do we come from? How can life arise despite the extremely small probability that such an event could take place? What is the difference between thinking organic and non-organic matter? Before we can answer these questions, we must ask—and answer— three others. What are the main general scientific principles and mathematical models, if any, on which we should base such a theory? What experimental data do we need for a better understanding of the dynamic processes of living systems, from viruses and bacteria to individuals and societies? What will result from the interaction between artificially made "living systems" and ourselves as representatives of natural biological objects?

It is obvious that we have a long way to go before we have the answers to these three questions. But we can certainly make a start by studying artificially made systems—computers—which, at this modern stage of development, have many features that we could previously detect only in natural living systems. In addition, today we are much more advanced in our understanding of the operational rules of computers. In many cases can use these rules as a model for the simulation of real living systems, and, as we intend to demonstrate here, we can conversely apply some of the natural biological rules to artificial living systems.

Background

Computers can be considered to be living systems: they develop from one generation to another (in this case we know the creator); they consume energy for their operation; they can think (in the sense that they can play, and even win at, chess, act as not bad consultants for stock exchange operations, and can control of aircraft and other complex systems, etc.); they can get sick(with computer viruses): they are capable of exchanging information; they can be organized in groups and are already being organized into a community (via networks and the internet); they can learn and teach; and finally, they can design and produce new computers. But what is more important for this publication, modern computers can produce information not only under control of programs made by programmers, according to clear mathematical and logical rules, computers can also produce unpredictable information, opening a new page in the history of a computers. This has directed us to view computers as "active thinking systems".

How can we assimilate the idea that a computer can produce information of some new value that is outside the framework of our initial intentions and our instructions to the computer in the form of the computer programs and what should be done to exploit such cases of "computer creativity" for good for their creators? On the basis of some new scientific results, we intend to demonstrate that the whole field of computers is much more complicated than was expected at the beginning of the computer era. To understand better the new problems that we face with modern computer systems, it is preferable to view them as living systems, with special internal laws that we are only in the beginning stages of defining.

The idea of considering the computer as an instrument to aid us to accelerate our analytical and numerical calculations and even to help us in decision making was based on the assumption that computer errors can be made as small as necessary. In this case, there is no difference between analytical and computer solutions of, say, differential equations or any other mathematical models conventionally used for programming. The situation has, however, been drastically changed since chaotic solutions have been obtained and discussion has been opened about the meaning of this kind of solutions [2]. Chaos theory forces us to ask the questions: How can we control numerical solutions when infinitesimal changes of initial conditions or parameters can strongly affect the final results? How we can check the agreement of our mathematical model expressed by differential equations with its numerical solution? The complexity of the initial equations do not give us the possibility to obtain an analytical solution even in the simplest case of chaotic systems.

The traditional chain mathematical model —> numerical solution is broken by this finding. Computers (i.e., numerical methods & computer errors) have become an essential part of the final results and should therefore be considered as an active component of decision making.

The concept of computer systems as "living systems" in a sense that they can produce some "free choice behavior" appeared when an awareness of chaotic systems emerged. We intend to use this concept for producing a new generation of artificial living intellectual systems. This approach can explain many unpredictable situations in modeling complex systems and will open a new horizon in the use of computers. As an a example, which we will use for practical demonstration of the idea presented above, let us consider a mathematical model for chemical transformations, based on a new discrete chaotic dynamics [3]. The use the chemical reaction dynamics as a model for the behavior of complex systems is based on the fact that many processes can be represented by very the effective language of stoichiometry of a chemical reaction:

$$\sum_{i=1}^{N} \nu_{li} A_i = 0 \qquad i = 1,2,\ldots,N \quad l = 1.2,\ldots,L \qquad (1)$$

where A_i - any element or constituent that can be combined by a matrix of stoichiometric coefficients (ν_{li}) in the form of a chemical reactions. Using this approach, we can use the equations of the law of mass conservation:

$$\sum_{i=1}^{N} a_{ij} X_i(t_q r) = b_j \qquad j = 1, 2, \ldots, M \qquad (2)$$

where a_{ij} - "molecular" matrix defining the quantity of the j^{th} component in the i^{th}constituent and $X_i(t_q,r)$ - quantitative measure of the constituent A_i, in which t_q,r - discrete time and space coordinate.

According to classical chemical rules, the dynamics of chemical mass transformations can be described by system of differential equations of the kinetic mass action law or by new discrete chemical reaction dynamics:

$$\prod_{i=1}^{N} X_{li}^{v_{li}}(t_q, r) = \pi_l \qquad l = 1, 2, .\,.\,.\,, L = N - M, \qquad (3)$$

where $(\pi_l) = K_l \exp [-W_l / t_q]$ $\qquad l = 1, 2, .\,.\,.\,, L = N - M,$

$$W_l(X(t_{q-s}, r) = \sum_{i=1}^{N} \left(\alpha_{li} X_i(t_{q-s}) + \beta_{li} X_i(r) \right) \qquad (4)$$

and α_{li}, β_{li} are empirical parameters. In an open system, t_q changes from 0 to the constant - t_c, the time at which the steady state is reached; t_c should also be considered as an empirical parameter, defined from the experimental data.
In the latter case, we can take into consideration in our equations not only the dynamics of mass transformation but also the so-called "information exchange". For example:

$$\begin{tikzcd} A_1 \arrow[rr, bend left] & + \; A_2 \arrow[r] & A_3 \end{tikzcd}$$

represents a simple bimolecular reaction, but the error indicated on the top of this reaction means that the concentration of A_1- X_1 will influence the rate of chemical transformation and can thus be considered as what we have called "information exchange".

Based on a system of difference equations of discrete chemical reaction dynamics [3], we have obtained a method of construction the equations, which by their very nature contained different chaotic solutions and hence could produce many regimes including chaotic ones and therefore strongly affecting by the accuracy of the computer and numerical methods.

Results of numerical calculations

Here we will present some results that we obtained using discrete dynamics equations to represent the spatio-temporal behavior of chemical reactions . Our intention was to simulate patterns that have been observed in experiments on a B-Z reaction [3]. The results we are looking for, is presented in Picture 1. However, the simulation patterns obtained for regimes very close to those that we were looking at previously have nothing in common with experiments on the B-Z reaction. These latter very decorative patterns are presented in Picture 2.

The production of ornaments and decorations is one of the oldest creative activities man. It is a complicated procedure requiring intellectual efforts and special talents. The fact that the patterns we can get from mathematical formulas of the same level of complexity are very similar to hand-made ornamentation reflects the situation described in the introduction, in which computers can produce unpredictable information of some value, but being unpredictable, it demands understanding and further control. To realize such a program, we require a computer system that can make this program real, but again we are faced with the same situation of unpredictability on the next level of computer organization. This process has no limits, but we have no choice other than to produce a computer system of the next level of complexity for the control such a type of solutions. This computer system should be able: to analyze all the solutions that are obtained by changing the parameters of the mathematical model, to analyze the stability of these solutions; and to provide us with recommendations about the further use and changes of the initial mathematical model. The creation of such a system would bring us to a new level of image processing and automatic generation of images.

Conclusions

We have considered here only one example of the generation of much more information than we expect from the beginning from a computer algorithm based on a mathematical model. Such results are absolutely normal for human beings—natural living systems. Natural intellectual systems are intended to produce innovations, and the fact that this role can be played by a computer should bring us to the idea of considering computers as artificially made "living systems". If so, we can look at an evolutionary process of computer "life" and organization in the similar way that we view our own history in order not to repeat the same mistakes we have made in a past trying to organize our own future; we need coexist with the new computer civilization in the best possible way!

References

1. A. Michailov, Artificial Life: An Engineering Perspective, Springer Proceedings in Physics, Vol. 69. *Evolution of Dynamical Structures in Complex Systems.* Editors: R. Friedrich, A. Wunderlin, Springer Verlag, Berlin, Heidelberg, 1992.
2. J.H.E. Cartwright and O. Piro, The Dynamics of Runge-Kutta Methods, *International Journal of Bifurcation and Chaos*, V. 2, N. 3, 1992.
3. V. Gontar, Calculus of Iterations and Dynamics of Physicochemical Reactions, *Mathematics and Computers in Simulation*, V. 39, 1995.

Pic.1 Spiral waves in B-Z reaction

Pic.2 Ornaments have been produced by the same mathematical model, with a small changes of parameters.

Poster Abstracts

A TEST BASED ON TRAINING SAMPLES FOR THE OBSERVATION CLASSIFICATION PROBLEM

ISMIHAN G. BAIRAMOV
ANKARA UNIVERSITY. FACULTY OF SCIENCE,DEP. OF STATISTICS
06100, TANDOGAN ANKARA, E-MAIL:BAYRAMOV@SCIENCE.ANKARA.EDU.TR

Abstract. Consider the following model of classification of observations. Let G_0 and G_1 be two populations with unknown distribution functions. Suppose that the training samples $\{X_1^k, X_2^k, ..., X_n^k\}$ and $\{Y_1^k, Y_2^k, ..., Y_n^k\}$, $k = 1, 2, ..., m$ from G_0 and G_1 , respectively, are given. Consider a random sample $\{Z_1, Z_2, ..., Z_n\}$ from one of the two populations. Denote the hypothesis asserting that $\{Z_1, Z_2, ..., Z_n\} \in G_i$ $(i = 0, 1)$ by H_i $(i = 0, 1)$. Let G_0 and G_1 have unknown distribution functions $F(u_1, u_2, ..., u_n) = \prod_{i=1}^n F_i(u_i)$ and $F_Y(u_1, u_2, ..., u_n) = \prod_{i=1}^n F_i(u_i)$, respectively. In terms of statistical theory of pattern recognition, $\{X_1^k, X_2^k, ..., X_n^k\}$, $k = 1, 2, ..., m$ are measuring values of n signs $X_1, X_2, ..., X_n$,and number of observations equal to m.

This model can be used in many practical applications, for example, in medical diagnostics. Let G_0 and G_1 denote two groups of patients. Suppose that the group G_0 has a known disease A and group G_1 has a known disease B. (For example, A can be "diabetes mellitus" and B can be "diabetes insipidus"). $X_1, X_2, ..., X_n$ denote the set of symptoms of illness. Let $Z_1, Z_2, ..., Z_n$ be the set of symptoms of a new patient. If it is known that the patient is suffering from one of these illnesses, it is required to classify the patient into one of the group G_0 or G_1. In [1] and [2] , the set of criteria based on training samples are given , the error probabilities of first and second types have been determined. Statistical criteria construction based on training samples is closely related to the notion of confidence intervals with main distributed mass of general set [3] and with the notion of confidence set for the vector of selected values . In [4] the structure of confidence intervals for the class of general set with continuous distribution functions is studied.

Consider intervals $\Delta_i = (X_i^{(1)}, X_i^{(m)})$ and $\Delta_i^* = (Y_i^{(1)}, Y_i^{(m)})$ $i = 1, 2, ..., n$, where $X_i^{(j)}$ denotes j th order statistic constructed from the sample $X_i^1, X_i^2, ..., X_i^m$, $i = 1, 2, ..., n$. Parameter (sing) X_i is called informative if corresponding intervals Δ_i and Δ_i^* are noncrossing with probability 1. Let $X_{i_1}, X_{i_2}, ..., X_{i_r}$ be informative parameters. For simplicity, denote $X_{i_k} = \hat{X}_k$, and $\Delta_{i_k} = \hat{\Delta}_k$, $k = 1, 2, ..., r$. Let us define the following random variables:

$$\xi_i = \begin{cases} 1, & \hat{Z}_i \in \hat{\Delta}_i \\ 0, & \hat{Z}_i \notin \hat{\Delta}_i \end{cases} , \quad \eta_i = \begin{cases} 1, & \hat{Z}_i \in \hat{\Delta}_i^* \\ 0, & \hat{Z}_i \notin \hat{\Delta}_i^* \end{cases} \quad i = 1, 2, ..., r; S_r = \sum_{i=1}^r \xi_i, \ T_r = \sum_{i=1}^r \eta_i .$$

It has shown that the small magnitudes of statistic S_r testify against H_0, and we suggest tbe following new consistent test: reject H_0 if $S_r < x_\alpha$. The value x_α is selected according to $P\{S_r < x_\alpha / H_0\} = P\{T_r < x_\alpha / H_1\} = \sum_{k=0}^{x_\alpha} C_r^k \left(\frac{m-1}{m+1}\right)^k \left(\frac{2}{m+1}\right)^{r-k} \leq \alpha$.

References. 1. Bairamov I.G.,Petunin Yu.I.(1991) Statistical criteria based on training mappings. *Cybernetics 27,No.3.*408-413. 2. Bairamov I.G. (1992) Statistical criteria based on training samples in the problem of classification of observations. *Theor. Imovir. ta Mat. Statist. No.46.* 13-17.(in ukrainian).3. Bairamov I.G., Petunin Yu.I. Invariant confidence intervalsthat contains the general part of the distributed mass of general set. *Vichislitelnaya i prikladnaya matematika. Kiev, Vip.65.* 112-117.(in russian).4. Bairamov I.G., Petunin Yu.I.(1990) Structure of invariant confidence intervals containing the main distributed mass. *Theory of probability and its applications. Vol.35. No.1.* 15-26.

SIMULATION OF A PACKED DISTILLATION COLUMN USING ORTHOGONAL COLLOCATION AND FINITE ELEMENT

Y. Cabbar[1], M. Alpbaz[2], H. Hapoğlu[2], S. Karacan[2]
[1]Ministry of Environment, Istanbul Caddesi No:98, 06100 Iskitler, Ankara Turkey, Fax: (90 312) 384 13 61
E-mail: cabbar@eros.science.ankara.edu.tr, [2]Ankara University, Chem. Eng. Dep. 06100 Tandogan, Ankara, Turkey, E-mail:alpbaz@eros.science.ankara.edu.tr

The separation of liquid mixtures into their components is one of the major processes of the chemical, petroleum refining and petrochemical industries and the distillation is the most widely used method of achieving this. Packed columns have found extensive application in liquid -gas contact systems, primarily in absorption and distillation oprations with small cross-section area requirements. In this work, the dynamic properties of a pilot plant packed distillation column were investigated experimentally and theoretically. The time response of output variable, concentration and temperature of top product, under the effect of the step and pulse changes given to the input variables have been observed. The on-line computer connections with thermocouples were utilized to obtain temperature profile through the packed column. The computer connection also was used for PID control of heat exchanger for boiler.

In the first part of the work, discretization in the spatial variable is achieved by orthogonal collocation. This method attempts to minimize the residuals in the differential equations at selected points in the column. The collocation points were chosen as the zeros of orthogonal Jacobi polynomials [1].

In the second part of the work, the solution region is considered as built up of many small, interconnected subregions called finite elements. To find the approximate solution of a nonlinear problems. The finite element method based on Galerkin Criteria [2] and Rayleigh-Ritz method have been tested. However in the finite element method, it may often be possible to improve or refine the approximate solution by spending more computational effort.

In the present work, dynamic and static properties of packed distillation columns were investigated experimentally and theoretically. Comparision of finite element and orthogonal collocation solutions and experimental work show good - agreement given in Figure (1-2).

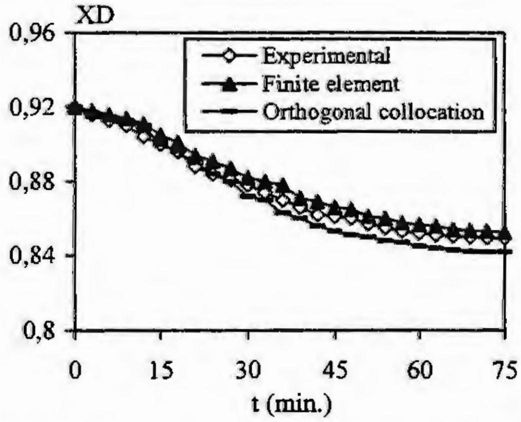


Figure 1 Response of top product mole fraction of methanol step change in Reflux ratio from 3 to 1.

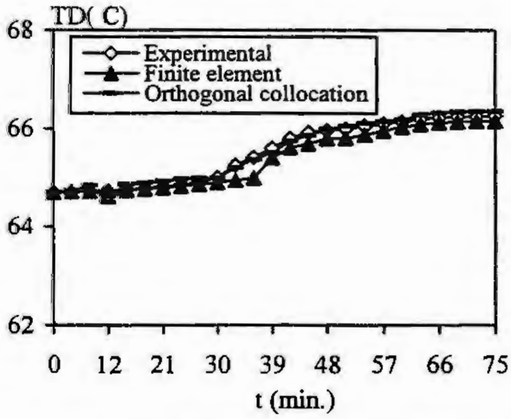


Figure 2 Response of top product temperature step change in reflux ratio from 3 to 1.

Key Words: Packed Distillation Column, Orthogonal Collocation, Finite Element, Dynamic Simulation

References

[1] Srivastava, R. K. and Joseph, B., Simulation of packed bed seperation processes using orthogonal collocation. Copm. And Chem. Eng. **8**, (1984), 1, p. 43-50

[2] Aly, S., Pibouleau, L. and Domenech S., Traitement par une methode d'elemrnts finis de modeled de colonnes de rectification discontinue a garnissage. *Can. J. Chem. Eng.* **65**, (1987), p. 991-1003.

Predictive model of ambulatory systolic blood pressure, using a semi quantitative measure of physical activity.

S. Charbonnier, F. Marques, A. Chéruy
Laboratoire d'Automatique de Grenoble
ENSIEG/INPG BP 46 F-38402 St Martin d'Hères

Ambulatory systolic blood pressure (ASBP) is a time varying variable which may increase significantly, in response to emotional stresses and to physical activity. The diagnosis of hypertension or the test of a patient's response to a drug treatement are based on the analysis of a 24 hours profile of this variable, obtained by a set of measurements realised every 15 minutes. The variability of ASBP makes the physician's analysis more complex ; indeed, an increase in ASBP is not necessarily pathologycal, it may result from a certain level of physical activity. So, a software aid for hypertension diagnosis is under study, based on a predictive model of ASBP variations, using measurements of physical activity. The model, developped with data from healthy persons, will be used to calculate a profile of ASBP in response to a patient's daily activity and the predicted profile will be then compared with the patient's recorded profile. In his hypertension diagnosis, the physician should take into account only the peaks unexplained by the model. However, unmeasured parameters, like emotional stresses, and the disparities of ASBP variations between people makes the modelling task difficult. A preliminary modelling study has been achieved and is presented on this poster.

An experiment has been carried out on 23 healthy young volunteers. The patient's ASBP and heart rate were measured every 15 minutes, during 24 hours, while the patient was living a regular day, and the time of the measurement was recorded. The patient was asked to note down its physical activity at the moment when the measurement was made. In order to have a semi quantitative measure of the activity, he had to choose between 14 levels of activity, scaled from sleeping to running fast. For two patients, the experiment was repeated seven times, during seven non consecutive days. To determine the precision of the ASBP measurement, every measurement was repeated twice on a patient during 12 hours.

At first, an analysis of the data was performed that led to group the 14 levels of activity into 6 levels between which a significant change in ASBP level could be observed. It showed that the level of activity or the heart rate value measured at time t was correlated with the value of ASBP at the same time, but had no influence on the value of ASBP recorded 15 minutes later; the time of the measurement had no influence either on the value of ASBP (i.e. no influence of a circadian circle was observed). A static model was developped, based on analysis of variance, since the activity measurement is only a semiquantitative one. It is of the form: $\Delta\hat{P} = \Delta A_i + \beta_i[\Delta Fc - (\Delta Fc)o_i]$ where $\Delta\hat{P}$ is the variation of ASBP predicted by the model, ΔA_i is the variation of ASBP due to activity i, β_i is the regression coefficient between ASBP and heart rate in the activity i, $(\Delta Fc)_{oi}$ is the mean deviation of heart rate in activity i. The variation of ASBP is predicted in response to the level i of activity and the heart rate deviation, Δfc.

Two kinds of models were developped. An individual one was realised with data from six non consecutive days of experiments recorded on the same patient. It was validated on the seventh day of measurements. A model of the group was achieved with data coming from 22 recordings from 22 different patients and validated on data from the 23rd patient. The model was evaluated on the 23 patients with a 'leaving one out' method, so as not to validate the model on data used for its elaboration.

The individual model predicts 81% of the fluctuations of ASBP on the seventh day for patient A and 66% for patient B (calculated with $R^2 = 100 * [1 - (\Sigma(\Delta P - \Delta\hat{P})^2 / \Sigma\Delta P^2])$. The mean prediction error is about 5%, which is equivalent to the measurement precision. The group model predicts in average 41% of ASBP variations, with extrema from 0% to 72%. The prediction is very poor for four patients (0%), who happened to be patients who had a very quiet day, where only the lowest activities were noted. These patients excepted, the model explains 50% in average of the variations with a minimum of 36% and is especially good at predicting the increases in ASBP, which is the important matter.

This study shows in a quantitative way the influence of physical activity on ASBP variations, and proves the reproducibility of its effect on ASBP. The validation tests show how a model elaborated with data from a group of patients predicts surprisingly well the rises of ASBP due to important physical activities on a new patient. Considering the means used to measure physical activity, the results obtained are very promising in a view of hypertension diagnosis and permits us to conclude that the study should be carry on, with the acquisition of an electronic activity monitor, which would provide, in an automatic way, more reliable and more numerous data on the patient's activity.

DYNAMIC MODEL SIMULATION OF AEROBIC YEAST PRODUCTION IN A BATCH BIOREACTOR

S. ERTUNÇ[1], N. BURSALI[1], Y. CABBAR[2], M. ALPBAZ[1]
[1]Chemical Engineering Department, Ankara University Faculty of Science ,
06100,Tandoğan, Ankara, Turkey
E-mail: Alpbaz@eros.science.ankara.edu.tr
[2]Ministry of Environment, İstanbul Caddesi No:98, 06060 İskitler, Ankara, Turkey

In this study; Mathematical models defining the dynamic properties of *S.cerevisiae* production in a batch bioreactor under aerobic condition were developed for the purpose of the temperature control of the batch bioreactor.

Experimental studies were conducted on a 2L glass bioreactor which was coupled with an on-line computer with cooling jacket. There were an electrical heater, an agitatior, a thermocouple, pH and dissolved oxygen probes in the bioreactor. A thermocouple which measure the bioreactor temperature was connected to the computer with A/D converter. Air was continuously supplied to the bioreactor passing through an air filter and sparger. There was a triac module connected to computer with D/A converter for the purpose of the on-line temperature control.

In the experimental studies, we examine the dynamic properties of this bioprocess. It was observed that the spesific growth rate and the concentration of the microorganism decreased with the increasing temperature in the culture results of the aerobic glucose consumption. Because of the undesired effect of the increasing culture temperature on the microorganism performance it was necessary to perform the temperature control. Therefore modelling of the bioprocess was performed to check experimental results related the both temperature control and dynamic properties of the bioprocess.

The equations illustrating the dynamics of the substrate, microorganism dissolved and gas phase oxygen concentration were established. Also the dynamics of the H^+ ions in the culture was illustrated by developing an extra mass balance equations related to the dynamics of carbondioxide concentrations in gas and liquid phase.

On the other hand mathematical models indicating energy balance for the dynamics of the temperature of the culture in the batch bioreactor and the related equation for the water temperature leaving from the jacket were established. The heat balance between culture medium heated by immersed heater in the bioreactor and coolant jacket was considered as a continuous system.

Besides of them recursive linear model was obtained. The ARMAX model was used for this bioprocess and the parameters of the related model was determined by using Bierman algorithm. The time variation of all the system variables was obtained by using digital computer.

To check the experimental and theoretical results, a step changes were given to the heat input and to the cooling water flow rate and the time variation of all variables were observed .

The theoretical results and experimental data were compered. It was observed that there was very good agreement between them. Hence, it was decided that these related models can be used to design the efficient control system and algorithms for temperature control of the batch bioreactor.

Key Words: Dynamic Model Simulation, Biotechnology, Aerobic Fermentation.

References

1. Cardello, R.J. and San K.Y., The design of controllers for batch bioreactors. Biotechnol. Bioeng., 32 (1988), 519-526.
2. Bierman, G.J. Measurement updating using the U-D factorisation. Automatica, 12 (1976), 375-382.
3. Sweere, A.P.J. and at.all. Modelling the dynamic behaviour of Sacchoromyces cerevisae and its application in control experiments. Appl. Microb. Biotechnol. 28 (1988), 116-127.

MODELLING A COMMUNAL DISPOSAL SYSTEM FOR BIOLOGICAL WASTE

F. Koch[1], **P. Krejsa**[2], **E. Rybin**[3], **M. Schönerklee**
Modellbildung und Simulationen
Bereich Verfahrens- und Umwelttechnik
Österr. Forschungszentrum Seibersdorf Ges.m.b.H.
A-2444 Seibersdorf

F. Breitenecker[4], **M. Holzinger**[5], **M. Lingl**[6],
M. Zimmermann[7]
Dept. of Simulation Techniques
Technical University Vienna
Wiedner Hauptstraße 8-10, A-1040 Wien

Introduction

The system to be modelled consists of several subsystems, such as sorting, composting, drying, and fermentation devices, a sewage, an incinerator, and a device for reducing biological substances with the help of bacteria, which is called "Aerobe Thermophile Stabilization (ATS)". The single devices are examined (where they already exist) or modelled continuously (where they do not) in order to gain data for a discrete simulation of the whole system.

Continuous submodels

Most of the systems already exist, and enough data are available. But two devices (the ATS and a special incinerator working with turbulent processes) need to be newly designed for this disposal system. They are modelled in detail as continuous systems.

The ATS

The ATS is a device consisting of two tanks, where the biological substances of sewage sludge are reduced by bacteria. The sludge is filled into the first tank and left there for one day. Then part of it is moved to the second tank, and the first tank is refilled with new sludge. After the second day, a certain amount of the sludge in the second tank, which is now finished, is taken away, and tank number two is refilled from tank number one, where fresh sludge is added. This batch process is repeated endlessly, thereby always keeping enough bacteria back in order to keep the process running.
Parameters to be considered in this model are:

- concentration of organic substances (dry)
- share of water in the sludge
- concentration of oxygen
- temperature
- concentration of several other non-organic substances

This model is intended to supply data for the discrete model describing how the process works with different input parameters.

The incinerator

Modelling the inner range of the waste burner will be done by discretization of the Navier-Stokes equations by means of finite differences. First, a two-dimensional approach is planned, but the inhomogenity of the problem should require a three-dimensional model.

The next step is to simulate the problem by regarding multiphase flows (e.g. cole, sand, waste, air, ...) and to describe the chemical reactions which cause phase transitions. The model will be identified by a research stove which will be erected by colleagues from the TU-Dresden, Germany. Finally, some changes of geometry of the incinerator are planned in order to optimize the burning process.

The discrete model

The discrete model shall grant a basis for strategic decisions, such as which devices should really be built and at which size. Moreover it shall work as a control tool for the finished system.

To achieve these purposes different methods of optimization will have to be used. As this system is too complex to use analytic methods, it will be necessary to use numeric methods as well as new methods of soft computing, such as fuzzy logic.

[1] koch@.arcs.ac.at, [2] krejsa@.arcs.ac.at, [3] rybin@.arcs.ac.at, [4] Felix.Breitenecker@tuwien.ac.at,
[5] mholz@osiris.tuwien.ac.at, [6] mlingl@osiris.tuwien.ac.at, [7] mzimmer@osiris.tuwien.ac.at

MODELLING OF THE HUMAN ARTERIAL NETWORK FOR AN EXPERT SYSTEM FOR PREOPERATIVE PREDICTIONS

W. Bornatowicz, K. Kaser, J. Krocza, M. Suda
Austrian Research Center Seibersdorf
A-2444 Seibersdorf

C. Almeder, F. Breitenecker, S. Wassertheurer
Technical University, Vienna
Wiedner Hauptstraße 8-10, A-1040 Wien

The aims of this project are the extension and preparation of a mathematical model that describes the relationship of morphology and hydraulics in human arterial networks and the development of an expert system connected with the model [1]. Both parts will be combined in a graphic oriented software package. With this system mean flow velocity, mean flux, flow direction and blood pressure at any point of a vessel network can be calculated. By changing the topology of a network the hydraulic effects of stenoses and bypasses are simulated. The results of these simulations may help physicians in their decisions, if an operation is necessary and which kind of operation has the best chance of success.

In the mathematical model the vessel networks are simplified to hydraulic pipe networks [2]. The input parameters are the length and the diameter of each vessel and the network topology. For easier and faster working, several preconfigured standard networks (e.g.: leg arteries, cerebral arteries, etc.) are included in the package. But the physiology can be quite different, therefore those standard models have to be changed into specific patient models by adapting them on data from ultrasonic Doppler measurements and X-ray pictures. It is planned to automate this process of adjustment using an expert system. When the patient model is configured a physician can simulate different operation methods on the screen (changing the place or dimensions of bypasses, widen closed vessels, etc.). Immediately the effects of these operations relating to the blood flow are shown on the screen.

Some assumptions and simplifications were necessary to get a compact mathematical model that can be handled. During the adaptation of the standard model on the patient data the whole model calculation has to be carried out several times. Therefore these calculations must not be too time-consuming. The fact, that in most cases only few patient data are available for the adaptation, requires a small model, so that its parameters can be identified with the measured data. Nevertheless tests have shown a mean difference of only up to 10% between calculated and measured flow velocity depending on the amount and quality of data available. This inaccuracy is insignificant, because for the decisions on the kind of operation only fundamental statements of the flow velocity, flow direction and blood pressure are necessary, that means that no exact values are needed, but ranges. A more complicated aspect is the adaptation of standard models on the data received from different measurements and diagnoses of the physicians. Much experience and knowledge of the mathematical model are necessary to do this adjustment by hand, and even then it takes some hours. To automate this time-consuming work an expert system will be constructed and implemented between the user interface and the model interface.

In summary, the project should lead to a user-friendly software package for physicians that can be used as an advisor in vessel surgery and maybe as a training tool for medicine students.

References

1. M. Suda, O. J. Eder, B. Kunsch , D. Magometschnigg, H. Magometschnigg, *Preoperative assessment and prediction of postoperative results in artificial arterial network using computer simulation.* Computer Methods and Programs in Biomedicine, 41 (1993), 77-87.
2. R. Epp and A. G. Fowler, *Efficient code for steady-state flows in networks*, J. Hydr. Div., 96 (1970), 43-56

Electro-Hydrodynamics

Abstract

Winston Khan, Ph.D.

University of Puerto Rico, Mayaguez

U.S.A.

The problem contempleted here involves the viscous flow around a fixed sphere, which would occur if the sphere was surrounded by a uniform electric double layer and influenced by a uniform electric field.

The boundary conditions involved in this problem are as follows: $v_r = v_\theta = v_\varphi = 0$ as $r \to \infty$ and $v_r = v_\varphi = 0$, $v_\theta = k\sin\theta$ for $r = a$, where the centre of the sphere is taken as pole of spherical coordinates (r, θ, φ).

The boundary conditions are of an unusual kind in Hydrodynamics, but demanded by the problem contemplated.

We,therefore,seek a solution to the usual hydrodynamic equations of Div $v = 0$ and $\rho\left(\frac{dv}{dt} - F\right) = \nabla P + \eta\nabla^2 v$, inside the double layer with the unusual boundary conditions above.

The author believes that the velocity distribution inside the double-layer could be obtained in closed form as $v_r = f(r)\cos\theta$ and $v_\theta = g(r)\sin\theta$, where both f(r) and g(r) would incorporate the constant k, the applied field and the viscosity of the medium.

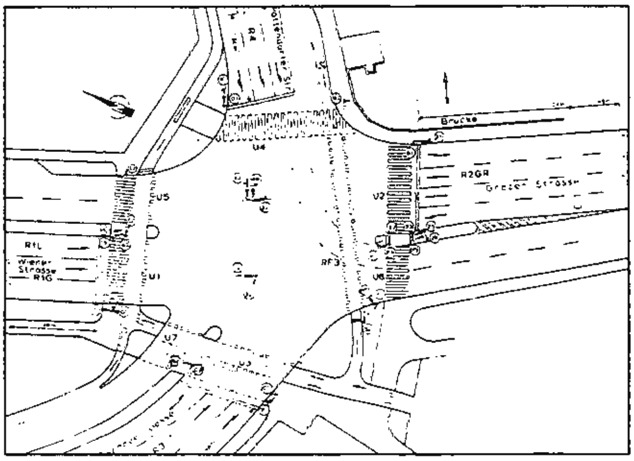
The technique of Henry, D. C. 1931 Proc. Royal Society A. 133 for the complementary problem of Electrophoresis was closely observed.

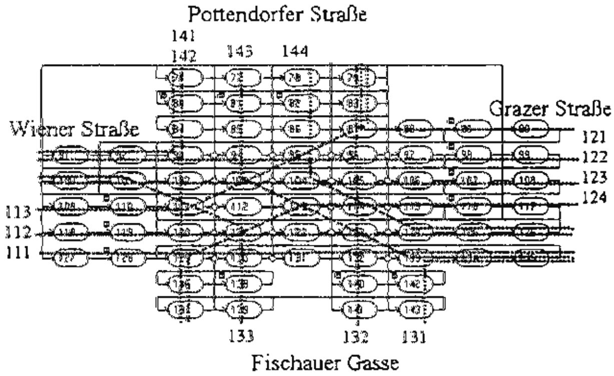
MODELLING OF A TRAFFIC JUNCTION WITH A GENERAL PURPOSE GRAPHICAL DISCRETE SIMULATOR

C. Kiss
Dept. for Simulation Techniques, Technical University of Vienna
Wiedner Hauptstraße 8 - 10, A-1040 Wien

Abstract. The aim of this contribution is to demonstrate a modelling procedure for a traffic system in microscopic view with Micro Saint. The model should show one possible approach to use general purpose discrete simulators for modelling and simulation of traffic systems.

A short overview





The pictures above are showing the real system and a part of the implementation of the model in Micro Saint. The model uses entities to simulate vehicles and „tasks" to realise parts of the street (tasks are the general module in Micro Saint to describe an kind of static objects). Further the logical structure of the traffic light control has to be also modelled by tasks. The structure of the tasks is chosen in a special layout in order to use the model itself as an animation.

The main advantage of Micro Saint is the fact that it is requires very short learning time because of its simple structure. But the problem is that traffic systems are very complex and this complexity. Many phenomena that rise in traffic simulation systems, can not be implemented in Micro Saint without reaching unacceptable runtimes.

A short overview about the risen difficulties will make clear that doing traffic simulation with such a general tool needs a lot of good ideas. But it is also shown that it is not always necessary to buy an expensive specialised traffic simulator. Models of manageable size can often be implemented with less costs and less work in a general purpose simulator, which everyone calls his own, doesn't he.

Further information (if you missed the Poster Session) can be received by Claus Kiss (Email: jeanluc@osiris.tuwien.ac.at, WWW: http://eurosim.tuwien.ac.at/~jeanluc/)

MODELLING INDIVIDUAL PROPERTIES IN MICROSCOPIC TRAFFIC SIMULATION USING GPSS/H®

M. Klug
Department for Simulation Techniques, Technical University of Vienna
Wiedner Hauptstraße 8 - 10, A-1040 Vienna, Austria
mklug@osiris.tuwien.ac.at

For modelling and simulation of road traffic in a more accurate way, it is necessary to implement e.g. individual properties of the "entities" (cars) flowing through the system. A model was built, where the cars may pass a crossing if there is yellow light, they may drive at their own speed in a range between 40 and 60 km/h, they have got individual temporary reaction distance (i.e. they have a distance to the car in front between 0,8 and 1,2 seconds), they use their own space while waiting for green light in a lane at, etc. (all real „daily phenomena). Clearly, each vehicle may have its own length, or belongs to different classes of types (car, truck, bus, etc.)

These individual properties can be seen as attributes to each entity. Using GPSS/H® there is no problem to implement all these individual and partly state-dependent properties by means of general attributes and blocks for changing attributes.

Implementation of individual length

First the type of the car has to be fixed. Depending on the type the length of the car is computed with triangular distributions. The values can be seen in the following table (numbers in cm):

Type of vehicle:	Min.:	Med.:	Max.:
Bike:	150	180	200
Car:	350	420	560
Truck:	600	900	1200
Bus:	800	1000	1200
Truck with Trailer:	1000	1200	1500

Implementation of different velocity and reaction time

The different individual velocities are implemented in the following way: each section of a street (lane) is characterised by its own length, stored as capacity of a storage. Each car entering the lane (storage) goes along with its time depending on its velocity, which is also triangular distributed with parameters 40, 50, 60 km/h, computed by a function. At the end a facility is set, first, to stop the traffic if the lane in front has no space available for another car, and second, to control the stored reaction time. A car remains in this facility for the time triangular distributed with min.: 0,8, med.: 1,0 and max.: 1,2 together with its length divided by its velocity. If the car is able to continue the next car enters this facility and stays there, etc. .

Implementation of passing a yellow traffic light

This very individual behaviour is realised by closing a facility which a car has to pass if it drives into the crossing. Therefore a control entity staying in an endless process is used to control the traffic light at accurate times. If the light starts to show yellow signal, this entity is splitted. The original entity remains controlling the traffic light, while the copied one waits until it closes this mentioned facility. The waiting time is uniformly distributed between 0 and 2 seconds.

Interestingly individual behaviour did not effect the whole system behaviour seriously, but the flow was more efficient (shorter queues). A 24 hours simulation with about 2800 cars in hour took nearly 12 minutes on a 486DX4-133 computer.

SIMULATION AND IDENTIFICATION OF COMPARTMENT MODELS FOR THE DETERMINATION OF PARAMETERS OF RENAL FUNCTION

H. Url and F. Breitenecker
Technical University, Vienna
Wiedner Hauptstraße 8-10, A-1040 Wien

The study presents a mathematical method for the determination of renal function parameters by kinetic experiments. For this purpose suitable markers are applied intravenously. The temporal concentration profiles gained serve as data base for the assessment of characteristic organismic constants such as the clearance or the distribution volumes of the exogenous substances applied. For the estimation of such system parameters dynamic models of the kinetic processes are formulated and adapted to the time courses of marker concentrations in plasma samples [2, 4, 5].

In this so-called 'inverse' problem the system's response to an exogenous perturbation and the theoretical model are given, whereas the system parameters are unknown. Therefore the aim of this study is firstly to explicate a method for the identification of the model parameters on the basis of experimental dynamic data and suitable model equations and secondly, to demonstrate a robust method of estimating the errors of the parameters derived which are a consequence of the 'noise' of the experimental data [3]. In principle such a method of parameter identification consists in an iterative procedure of searching the minimum of an optimization criterion in parameter space. This criterion represents a measure of deviation between the theoretical model prediction and the experimentally observed system response. In the special case studied the two-compartment model is given in the form of a linear system of two simultaneous differential equations. The model solution has been derived in a closed analytical form by solving the eigenvalue problem and by simulating. The parameter optimization has been done by means of the Marquardt-Levenberg algorithm [1, 8].

For the estimation of the errors of the optimally fitted parameters a simulation procedure has been used. Thereby random numbers taken randomly from a Gaussian distribution of mean zero and standard deviation derived from the measure of deviation of the optimal model curve from the experimental data have been superposed on the optimal model fit. 100 such artificial protocols have been subjected to the parameter identification procedure described. The errors of the system parameters have been estimated on the bases of these data samples. As has been verified these error measures are essentially identical to the diagonal elements of the inverse of the so-called information-matrix of Fisher [6, 7].

In summary, the method presented demonstrates firstly the possibility of reducing the original time data to a set of model parameters and secondly the possibility of testing dynamic model hypotheses. The procedure thus allows one to gain by computer-based simulation techniques both qualitative insight and quantitative description of the mechanisms underlying the observed experimental data.

References

1. Buys, J., von Gadow, K., A PASCAL program for fitting nonlinear models on a micro-computer. EDV in Medizin und Biologie, 4 (1987), 105-107.
2. Carson, E. R., Cobelli, C., and Finkelstein, L., The Mathematical Modeling of Metabolic and Endocrine Systems. Model Formulation, Identification and Validation. Wiley, New York, 1982.
3. Carson, E. R., Godfrey, K. R., and Reeve, J., A Review of Modelling and the Role of Dynamic Tracer Studies in Metabolism. In: Quantitative Approaches to Metabolism, (Ed.: Cramp, D. G.) Wiley, Chichester, 1982, 1-72.
4. Estelberger, W., Petek, W., Zitta, S., Mauric, A., Horn, S., Holzer, H., and Pogglitsch H. Determination of glomerular filtration rate by identification of sinistrin kinetics. European Journal of Clinical Chemistry and Clinical Biochemistry, 33 (1995), 200-209
5. Estelberger, W., Zitta, S., Lang, T., Mayer, F., Mauric, A., Horn, S., Holzer, H., Petek, W., and Reibnegger, G., System Identification of the Low-Dose Kinetic of p-Aminopihhuric Acid. European Journal of Clinical Chemistry and Climical Biochemistry, 33 (1995), 847-853.
6. McIntosh, J. E. A., and McIntosh, R. P., Mathematical Modelling and Computers in Endocrinology. Springer, Berlin, 1980.
7. Metzler, C. M., Statistical Properties of Kinetic Parameters. In: Pharmacokinetics During Drug Development: Data Analysis and Evaluation Techniques, (Eds.: Bozler, G. and van Rossum, J. M.) Fischer, Stuttgart, 1982, 138-144.
8. Press, W. H, Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., Numerical Recipes in Pascal. Cambridge University Press, Cambridge, 1989.

WHAT'S WRONG WITH BOND GRAPHS

H. Mann and J. Hozák
Czech Technical University, Computing Centre
Zikova 4, CZ-166 35 Praha 6, Czech Republic

Surprisingly, the sect believing that bond graphs are the best – or even the only possible approach to physical-level modelling of multidisciplinary systems seems to still survive. Tbis contribution is to demonstrate the Achilles tendon of bond graphs and the advantages of multipole models over them.

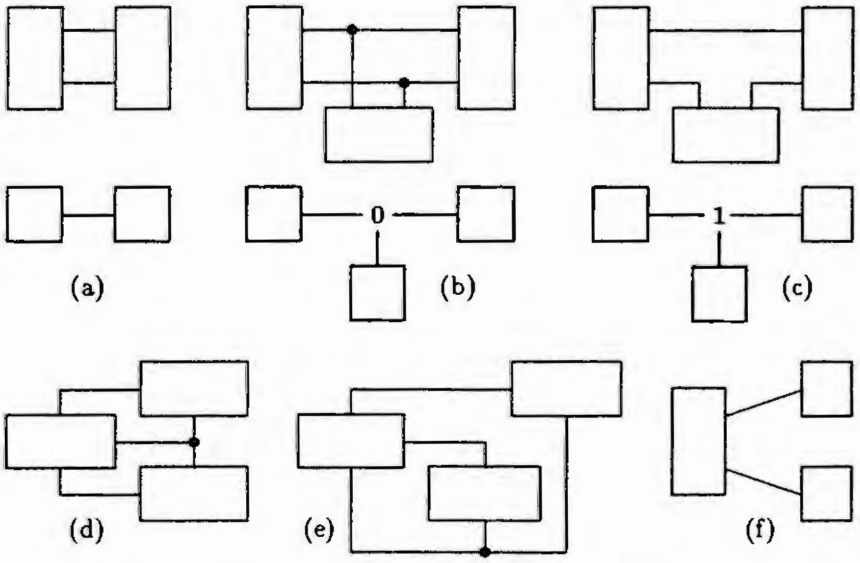


Fig. *a* shows two interconnected two-ports in the form of two-poles at the top, the same interconnection is represented by the bond-graph at the bottom. In the former case, each port is shown as a couple of poles, whereas in the case of the bond graph, each port is associated with a half-bond indicating a flow of energy. In most systems, however, besides the energy flows between system components, we are interested also in the pairs of complementary *across-* and *through-variables*[1] the product of which determines each of the energy flows. Unlike the two-pole ports, the bond-graph ports do not indicate the way in which across variables are associated with component models. It is this oversimplification of port representation that makes the bond graph approach so cumbersome. To distinguish between parallel and series interconnection of three one-ports shown in Figs. *b* and *c* respectively, the zero and unit junctions had to be introduced. Despite of this clumsy precaution the uncertainty about the variable identification is still there, however, especially in the case of multiports. Fig. *f* demonstrates this by a bond graph corresponding both to Fig. *d* and *e* despite the fact that their structure is quite different.

As it is very difficult to find any good simulation software based on bond graphs, they are converted usually into *block diagrams*. To simulate a physical model using a block-diagram oriented program, the model in the multipole form can be converted into a block diagram very easily using the *salesman* (or shortest-path) *algorithm*.

Fortunately, there are modelling and simulation methods as well as software tools which can be applied directly to physical models in the form of *multipole diagrams* in a straightforward manner. Unlike block diagrams, the multipole diagrams can be set-up from symbols representing the individual system components in a kit-like way based on mere inspection of real systems without any (bond) graph construction or equation derivation. The topological structure of multipole diagrams and that of the real systems which they represent is identical.

Examples of multipole-based modelling and simulation in different energy domains and engineering disciplines, besides the MATHMOD Symposium, can be also seen at

`http://icosym.cvut.cz/dyn/`

[1] In the multipole notation, *across-variables* are identical with the bond-diagram *efforts*, and *through-variables* correspond to the bond-graph *flows* except the mechanical energy domain where it is vice-versa. Whereas the across- and through-variables can be differentiated unambiguously by tbe way in which they are measured, the bond-graph notation stems from the historical view on causes of dynamic changes and their effects in different energy domains.

MATHEMATICAL MODELLING VIA INTERNET

H. Mann and L. Waldmann
Czech Technical University, Computing Centre
Zikova 4, CZ-166 35 Praha 6, Czech Republic

A set of projects focused on mathematical modelling and simulation, called *TeleSimulation*, is developed in the CTU Computing Centre with the following objectives in mind:

- to give to regular, distant or disabled students as well as to practicing engineers an opportunity to learn and to experience an efficient simulation methodology for multidisciplinary systems
- to develop a cost-effective access to a versatile simulation toolset in a way suited well especially to small- and medium-size industrial enterprises
- to support cooperation of remote teams in common research or engineering design projects
- to allow for a telelearning engineering course on simulation as a part of an 'in workplace', continuous, or life-long education programme

There are several components of the project, all of them accessible via Internet (the superscript numbers indicate the grant supporting the component development):

- an access to a powerful simulation engine[1,2]
- a library of models[1] of interdisciplinary system components
- a collection of simulation examples[1] of such systems
- a multimedial textbook[3] for systematic modelling of of engineering systems
- a conference system[1] for discussing simulation problems between users mutually as well as between users and a consultant or a course instructor

To provide the simulation engine, the package DYNAST^{TM} was implemented on the CTU supercomputer IMB PS2, and an access to it has been developed

- by e-mail[1] with a pseudographical form of output in the form of a printplot via the address:

  `dyn@sp05.civ.cvut.cz`
- from a WWW homepage[2] with a true graphical output via the URL:

  `http://icosym.cvut.cz/dyn/`
- by a user's environment[1] downloaded from the server – it allows for very well supported input data preparation and output data evaluation

The systems under investigation can be characterized

- by a set of nonlinear implicit-form algebro-differential equations in a textual form
- by a functional model in the form of a block diagram with any 'algebraic loops'
- by a physical model in the form of a multipole diagram representing the real system structure

or by a combination of these three approaches utilizing a library of models of typical system components.

This research has been conducted at the CTU Computing Centre within the following projects:

[1] *"Supercomputer and multimedial support to interdisciplinary design", grant TEN34CZ ID96032 by MŠMT (CR Ministry of Education, Youth and Sports)*

[2] *"Simulation of dynamic systems based on WWW" supported by the grant 230/96 by FRVŠ (CR Foundation for the Development of Universities)*

[3] *"Multimedial presentation of a textbook on engineering design using Internet", grant PREZENTACE PR96 393 by MŠMT*

CHANCES AND OPPORTUNITIES OF NUMERICAL SIMULATION IN REFRIGERATION

J. Philipp and M. Kauschke
Institute of Energymachinery and Machine Laboratory
Technical University Dresden
Mommsenstr. 13, D - 01062 Dresden

As the energy consumption of household and industrial refrigerators becomes an more and more important issue in current and future designs, tools such as numerical simulation to forecast the system behaviour are needed in order to find new approaches to control the system or be able to forecast the energy consumption under different environment and load conditions. We want to show the use of simulation on the optimisation of two refrigeration systems. In those distributed thermodynamic systems we have to include non-linear algebraic equations for the refrigerant properties. These properties may be the enthalpy, entropy, pressure, density etc.. Also the usage of different equations for the heat- and mass transfer coefficients should be possible. The behaviour of the systems itself is modelled by PDE which we solve by the method of lines. We want to explain how the programs work and will discuss the results.

The refrigeration capacity of household refrigerators is adopted to the requirement by on - off switching of the compressor. The system reaches very seldom a staedy state: most of the time the pressures, temperatures and mass flow rate are in transient. In order to forecast energy consumption one needs to take into acount those transient responses.

For industrial refrigerators working with Helium as refrigerant at a temperature of about 4 K the time to cool down the refrigerator from ambient temperature to the operating temperature is important. To minimize cooldown times (and save on energy consumption) one needs to investigate at the dynamics of the system.

References

/1/ A Dynamic Model for Helium Core Heat Exchangers; W.E.Schiesser, H.J. Shih, D.G. Hartzog, A.C.Hindemarsh, Supercolider 2, Edited by M. McAshan, Plenum Press, New York, 1990

/2/ Numerische Simulation von Haushaltkühl- und -gefriergeräten; J. Philipp, W.E. Kraus, H. Quack; Paper presented at the Deutsche Kälte- und Klimatagung 1996 in Leipzig / Germany

/3/ Safe and Efficient Operation of Multistage Compressor Systems; M. Kauschke, C. Haberstroh, H. Quack, Preceeding sof the 16th International Cryogenic Engineering Conference, Kitakyushu, Japan, 1996

TWO-WAY COMPOSITION FOR LOSSLESS LAYERED MEDIA SYSTEMS

Edward Szaraniec
Cracow University of Technology
ul. Warszawska 24, 31-155 Kraków, Poland
edszar @usk.pk.edu.pl

In a wide range of fields, transmission and/or scattering data in the lossless layered media systems have in the bacground a one-dimensional (1-D) profile of reflection coefficients (Schur parameters). Reflection coefficients are one of the most attractive kinds of parameters in parametrizing material profile in transmission and scattering problems. Most frequently, the reflection coefficients are represented by real numbers but in some applications (polarizable media [2], filters and transmission lines built-up from complex impedances [1]) they are complex-valued.

It is assumed to be known that the function f featuring energy propagation in 1-D medium belongs to C-, or S-class of analytic functions (positive-real, or bounded on the unit circle functions). The moments of f, collected in complex-frequency plane, serve as the data. The medium is discretized and scaled in such a way to deal, in the z-transform domain, with function f in z^2. The reflection coefficients and, consequently, the moments of function f are considered to be complex-valued.

The data are ordered on a line on complex frequency plane. In frequent cases this is a radial trajectory (relative to the unit circle). For example, earth resistivity sounding involves a segment of real axis, seismic sounding involves a segment of imaginary axis, and magnetotelluric sounding involves radially going straight line inclined under angle equal $(3/4)\pi$ radians [3].

Our point is that the data can be composed/decomposed involving some kernel profile of reflection coefficients which has a specific two-fold appearence: 1) monotone structure (all the reflection coefficients of the same sign), and 2) anticorrelated structure (reflection coefficients alternating in signs). This is depending on a trajectory of the arguments of kernel data on the complex frequency plane.

Definitely

$$f(z^2, A) = f(z, M) \odot f(z, O) = f(z, M) \odot f(-z, M)$$

where

f - function class C, or S,
A - given arbitrary structure,
M - kernel (monotone) structure developed from A,
O - alternating structure corresponding to M, $f(z,O) = f(-z,M)$,
z - variable in z-transform domain,
$\odot$ - composition symbol (summation for class C, multiplication for class S).

A parallel is drawn between composition of data and that of *structure under experiment*. The theorem involves composition of a monotone kernel structure with itself, but for different arguments. The arguments z, and -z refer to different (symmetric) trajectories (translated one with respect to another in complex frequency plane).

This recently published theory [4,5] has been found in terms of real-valued reflection coefficients. Now an extension is made towards inclusion of complex-valued reflection coefficients. To this end the main theorems and proofs are reformulated and done over again. Resulting theory is complete and general, composition/ decomposition of geophysical sounding data is naturally included.

Most advantageous application of such a model seems to be in data inversion, monitoring the changes of a structure during an experiment, and solving the structure sampling problem.

1. Bultheel, A., Laurent Series and their Padé Approximations, Birkhäuser, Basel-Boston, 1987.
2. Chew, W.Ch., Waves and Fields in Inhomogeneous Media, Van Nostrand Reinhold, New York, 1990.
3. Szaraniec, E., Fundamental functions for horizontally stratified earth, Geophysical Prospecting, 24 (1976), 528-548.
4. Szaraniec, E., A stable component inverse from insufficient data towards an eventual model. In: Inverse modelling in exploration geophysics, (Eds.: Vogel, A, et al.) Vieweg, Braunschweig/Wiesbaden,1989, 48-62.
5. Szaraniec, E., In-depth parameter for inversion in terms of the type of stratification. In: Geophysical data inversion and applications, (Eds.: Vogel, A., et al.) Vieweg, Braunschweig/Wiesbaden, 1990, 471-482.

ONE MODEL OF A HUMAN-MACHINE PROBLEM SOLVING SYSTEM

V.A.Vishnykov, O.V.German

Higher College of Communication, Staroborisovsky tract 8/2,
220114, Minsk, Belarus, tel.+372 17 2638552, fax +372 14 2641068,
email: vish@micro.rei.minsk.by

Abstract. The approach to the problem solving is considered. The structure of solution problem process is shown. The task concept form representation and it structure is declared and the architecture of the solving system is worked out. The strategies using wear methods in direction their utilising are proposed. The advantages of the human-machine problem solving system (HMPSS) in conclusion is given.

Practical realisation of the human-machine problem-solving systems (HMPSS) has some difficulties emanating from (i)the insufficiency of the pure logic methods and theorem proving techniques, and (ii) impossibility of formal interpretation and constructing conjecture making machine.

The solution process in the HMPSS with new paradigm is organised accordingly to the following scheme: (1) Specifying the problem; 2) making the semantic problem specification; 3) if there is an algorithm A with the specification Sa ineffaceable with the problem P specification Sp or Sp can be reduced to Sa by means of the equivalent transformations the A produces solution to P. (4) if there is no any suitable algorithm A then the human-solver controls solution process in an interactive mode using problem manipulation language.

Nondeterministic actions connected to a solution process are performed by the human-solver; meanwhile the computer performs the following operations: (1) testing model consistency; (2)variable and term substitution;(3) equivalent formal transformations; (4) partial and mixed computations; (5) concept unification; (6) generating and modifying problem objects; (7) definition of the required domains in the problem structure; (8) organising on intelligent helper; (9) producing consequences from hypothesis, etc.

The system architecture includes the following components: 1. Solution-Tree making subsystem; 2. Node-context developing subsystem; 3. Running subsystem 4. File and data base interface 5. Dynamic Library Linking .

Weak methods constitute an essential part of the HMPSS-architecture. By a weak method one understands a solution method which does not warrant obtaining an exact (correct) solution. The main problem of weak method application is divided into two subproblems. (A) Creating a number of strategies making weak methods strong. (B) Providing facilities to specify and integrate weak methods in solution procedure. We regard only subproblem (A). There are two basic strategies utilising weak methods: the strategy of iterations with cut's and branches-and-bounds schemata. It is commonly recognised that the latter one produces more efficient computational schemes than the former one. However, the strategy of iterations enables one to apply probabilistic methods for solving rather complicated combinatorial problems (see, for example, /7,8/). In this case, a solution is not warranted to be an optimal one but it is optimal in a probabilistic sense. The strategies utilising wear methods appears to be rather efficient for the large-size problems as well. The other idea is in combining branches-and-bounds and iteration schemata. An initial part of a solution process is realised on the basis of branches-and-bounds strategy and the rest of the solution process is accomplished as iteration procedure with cuttings.

In conclusion we give a list of the most essential advantages of the considered paradigm. These are: integrating man capabilities in solution process such as, intuition, abilities to learning and interpretation of errors, etc., preserving human interest in the solving activity, since human-solver remains the central factor in a "human-computer-problem" fried; putting wear methods to the forefront of the theoretical paradigm; providing new approaches to programming languages oriented to the processes in the HMPSS; considering different kinds of human-solvers and the corresponding system features suiting each type of the users; organising training courses in problem solving on the basis of weak methods and heuristic reasoning; developing semantic representation of the problem and manipulating it accordingly to the purposes of the solution procedure.

SIMULATION OF TEMPERATURE FIELDS BROUGHT ABOUT BY GRINDING HARDLY WORKED MATERIALS

S. F. Lushpenko
Institute for Problems for Machinery
Pjzharskogo, 2/10, 310046, Kharkov, Ukrainė

Abstract. In the paper the technique of simulation of the thermal process within the workpiece under grinding are given. At that, two- and three-dimensional formulations are used, temperature dependencies of the thermal properties and dependence of boundary actions on time and space coordinates are taken into account. The specific character of the simulated processes forces to pay special attention to information support of the simulation, fist of all, for appropriate modelling of boundary conditions.

Introduction

Effectiveness of grinding hard and superhard materials in the process of making tools of them as well as quality of these tools depend to a large extent on temperature levels characterising this process [4]. Temperature measurement at inner points of the workpiece under grinding or close by the cutting zone entails great difficulties. The most effective way of obtaining the information in this case is simulation of the temperature field of the workpiece with preliminary experimental determination of the boundary conditions of heat exchange.

As a rule, the field of temperatures in the workpiece under grinding is an object of simulation alone, for this temperatures and their gradients specify presence or absence of the causes for undesirable thermal structural modifications of the workpiece material as well as deformations and crack development and work as sufficient indications of efficiency of the whole grinding process. The surroundings and adjoining objects with inclusion of the grinding wheel are taken into account only by means of their thermal effects on the temperature field in the workpiece under grinding. To put it in another way, if we do not solve the problem in conjugate formulation and restrict ourselves with simulation of the process taking place in the workpiece then the actions of surrounding details find their reflection in boundary conditions. At this the simulation procedure becomes essentially simpler. Nevertheless, correct setting of the boundary conditions (which may have complicated form because of specific features of grinding processes) makes such simulation rather a difficult problem. In most cases such a problem can not be solved successfully with analytical methods of the mathematical physics, therefore the use of numerical simulation methods (see [3] for example) is one of the main features of investigation of thermal phenomena of grinding. Obtaining the main body of input data for simulation (first of all, parameters of the boundary conditions) from thermo-physical experiment is the second peculiarity. This causes the main points of the structure of the identification of the temperature field of a grinding workpiece.

Index of authors

Authors of papers:

Akesson K., 223
Albrecht-Früh U., 407
Alessandrini G., 1015
Antonelli G., 667
Armada M., 649
Arva P., 907
Atanasijevic-Kunc M., 63
Atherton D. P., 353, 787
Aude P., 97
Avula X. J. R., 883
Babarowski J., 993
Babary J. P., 877
Babuska R., 51
Bairamov I. G., 1105
Bajd T., 637
Banaszak Z., 135
Belic A., 1027
Benko M., 835
Bernzen W., 265
Bertrand J. M., 123
Beteau J. F., 853
Beuck G., 729
Bingshu W., 365
Bogaerts Ph., 57, 913
Bogatyriev M. U., 583
Bonsignore A., 655
Borodich F. M., 1089
Bourjij A., 253
Boutayeb M., 253, 307
Boz A. F., 787
Breit R., 599
Breitenecker F., 87, 229, 805, 1053
Brenner E., 347
Bretas N. G., 553
Brexel D., 605
Brown M., 775
Bungartz H., 827
Buzasy Gy., 835
Byrski W., 1071
Cabassud M., 895
Caccavale F., 667
Carpanzano E., 1039
Casamatta G., 895
Castillo J., 57, 913
Celle-Couenne F., 901
Chapuis J., 377
Charbonnier S., 1107
Chaume F., 853
Chiacchio P., 667
Claes J. E., 919
Claeys F., 859
Coen F., 859
Constales D., 799
Costa Alberto L. F., 553
Cruickshank C., 737
Cuno B., 793
Darouach M., 253, 307, 1077
Datsko B., 931
Dauphin-Tanguy G., 123, 129
del Re L., 383
Demchuk A., 931
Depecker P., 97
Derco J., 871
Derighetti M., 395
Dolanc G., 815
Dupre L. R., 541
Elmqvist H., 705
Enderlein V., 617
Engelhardt Ch., 827
Engell S., 17, 203
Estelberger W., 925
Fanni A., 217
Farkas A., 835
Fedorova A. N., 1083
Fedunov B. E., 1005
Ferkolj S., 577
Fernandes J., 509
Fernandez G., 649
Ferretti G., 655, 685
Fette M., 561, 569
Fetz Th., 81
Fiser R., 577
Flath M., 605
Flaus J. -M., 211
Fleischer G., 1009
Fotsu Ngwompo R., 117
Frank P. M., 307
Frei C. W., 395
Gafiychuk V., 931, 949
Gal I., 45
Gao B. F., 977
Gao H., 977
Gausemeier J., 605
Genrich H. J., 149
Gerharz I., 831
Gerstlauer A., 341
Gibon-Fargeot A. M., 901
Gilles E. D., 341
Girelli R., 335, 685, 1039
Giua A., 159, 217
Gontar V., 1063, 1095
Gonzalez de Santos P., 649
Gotlih K., 693
Göttsche Th. H., 751
Graber A., 491
Grabnar I., 1027

Gray G. J., 301, 1033
Grieco J. C., 649
Günther M., 535
Gutenbaum J., 993
Gutman M., 1063
Güttinger H., 389
Gwinner J., 809
Hahn H., 271, 281, 291
Hahn M., 617
Hammouri H., 901
Hangos K. M., 45
Hanisch H. -M., 165
Hanus R., 913
Haoxun C., 183
Haring G., 525
Hartmann S., 265
Hasegawa K., 135
Hecker F., 271
Hecker O., 643
Heidrich M., 611
Heijnen J. J., 865
Heiner M., 171
Heiss M., 757
Hellekalek P., 1
Hellinga C., 865
Hisseine D., 569
Höflinger W., 425
Hofmann B., 1009
Holodnov V., 103
Holzinger M., 805, 1109
Honekamp U., 617
Hromyak Yo., 931
Hunt K. J., 751
Hutter M., 235
Hvala N., 69
Ilchuk G., 931
Inkielman M., 993
Jahn T., 259
Janhsen A., 473, 485
Javor A., 835
Jimenez A., 631
Johansen T. A., 751
Jörgl H. P., 723
Julien S., 877
Juricic D., 45, 815, 1021
Kallmeyer F., 605
Kamel A., 129
Kamnik R., 637
Karba R., 63, 643, 1027
Karbuz S., 961
Kavsek-Biasizzo K., 643
Keller J. Y., 1077
Kiss C., 229
Klug M., 229
Köhne M., 743
Kopp R., 407
Kops B., 859
Korb R., 723
Kovacs A., 871
Kowalewski St., 203
Krämling A., 517
Kraus F., 383, 401
Krone M., 485
Kropp M., 763
Kubik P., 1071
Kugi A., 111
Kuznetsov L., 319
Kwak J. H., 359
Lang T., 925
Lautenbach K., 141
Le Lann M. V., 895
Lebret G., 313
Lee K. T., 359
Lee W. H., 359
Leimbach K. -D., 291
Lemmer K., 473
Lessard P., 877
Li Y., 301
Lingl M., 87
Lohmann B., 449
Lopez B., 981
Lubashevsky I., 949
Lückel J., 589
Lüder A., 165
Lunze J., 33
Lüthi J., 525
Lutter P., 937
Lutz B., 723
Lux Th., 847
Maffezzoni C., 335
Magnani G., 655
Marques F., 1107
Marschall G., 241
Marti K., 679
Martins de Carvalho J. L., 955
Matia F., 631
Matko D., 247, 637, 643
Mayer F., 925
Mechmeche C., 1077
Meißner F., 497
Melkebeek J. A. A., 541
Merz L., 729
Meyer D., 419
Michl T., 599
Mikhailov S. A., 781
Miksic M., 605
Milanic S., 63, 69, 643
Milek J., 389
Mircescu A., 1057
Mittwollen N., 599
Moeini A., 353
Moreira A. P. G. M., 955
Mrhar A., 1027
Müller K., 479
Müller P. C., 699, 781
Murray-Smith D. J., 301, 1033

Music G., 247
Naka T., 943
Naumov A. A., 323
Ndjab C., 371
Neck R., 961
Nemeth S., 907
Neumann M., 291
Neves J., 509
Nikishin V. B., 443
Nowakowski S., 253
Oberguggenberger M., 75
Obermüller J., 617
Orlowski C., 841
Otter M., 705
Pacheco J. M., 981
Pan D. H., 977
Panov G., 103
Park C. J., 359
Paul A., 611
Pavlovic V., 461
Pavlysh V., 931
Peters D. A., 455
Petras D., 517
Pfeiffer F., 661
Piepenbrink A., 281
Pijpers M. J. H., 189
Plank J., 1053
Plotnikov A. S., 443
Plotnikov P. K., 443
Prasad J. V. R., 455, 1065
Preisig H. A., 189
Primozic S., 1027
Prochnow D., 827
Prokopiou P., 623
Ptok B., 473
Rabenstein R., 1047
Rafaralahy H., 1077
Raisch J., 195
Rakar A., 815
Rausch M., 177
Reibnegger G., 925
Reinke K., 611
Remmele P. R., 763
Rempe M., 717
Riege B., 265
Rivera C. J., 1065
Rocco P., 673
Roos T., 763
Rosset E., 1015
Roussel J. M., 371
Rükgauer A., 329
Rusaouen G., 97
Rusinov L., 103
Rybin E., 1109
Sachs R., 377
Scavarda S., 117
Schäfer R. -P., 847
Schaufelberger W., 401
Scheidl R., 111
Schempp W., 437
Schiehlen W., 329
Schlacher K., 111
Schlothane M., 711
Schmid E., 235
Schmidt M., 847
Schneider A., 413
Schnieder E., 473, 479, 1057
Schnittger S., 503
Schönberger D., 743
Schuhen M., 743
Schultheiss J., 383
Sel D., 69
Setnes M., 51
Shansab S., 871
Sharman K. C., 301
Si J., 91
Sisterhenn R., 561
Skiba Yu. N., 431
Sosnin P. I., 999
Sridhar M., 883
Smik J., 661
Starkermann R., 985
Staudinger S., 971
Stöcklmayer Ch., 425
Stoschitzky K., 925
Strmcnik S., 69
Stursberg O., 203
Sueur C., 123
Sugisawa M., 135
Sydow A., 831, 847
Szucs G., 835
Theobald S., 793
Thieme J., 165
Thomasset D., 117
Tittus M., 223
Tolle H., 769
Tränkle F., 341
Tubielewicz A., 841
Tzafestas C. S., 623
Tzafestas S. G., 623
Ullrich T., 769, 775
Unger S., 831
Unger St., 821
Urbaniak A., 889
Van Impe J. F., 919
Van Keer R., 541, 799
van Loosdrecht M. C. M., 865
Vangheluwe H., 859
Vanrolleghem P., 859
Vansteenkiste G. C., 859

Authors of posters:

Almeder C., 1110
Alpbaz M., 1106, 1108
Bairamov I. G., 1105
Breitenecker F, 1109, 1110
Bursali N., 1108
Cabbar Y., 1106, 1108
Charbonnier S., 1107
Cheruy A., 1107
Ertunc S., 1108
German O. V., 1119
Hapoglu H., 1106
Hozak J., 1115
Karacan S., 1106
Kaser K., 1110
Kauschke M., 1117
Khan W., 1111
Kiss C, 1112
Klug M, 1113
Koch F., 1109
Krejsa P., 1109
Krocza J., 1110
Lingl M, 1109
Mann H., 1115, 1116
Marques F., 1107
Philipp J., 1117
Rybin E., 1109
Suda M., 1110
Szaraniec E., 1118

2nd MATHMOD
VIENNA
2nd MATHMOD
VIENNA
2nd MATHMOD
VIENNA

2nd MATHMOD VIENNA

PROCEEDINGS

late contribution volume

IMACS Symposium on
MATHEMATICAL MODELLING
February 5-7, 1997
Technical University Vienna, Austria

I. Troch
F. Breitenecker
editors

ARGESIM Report No. 11

ISBN 3-901608-11-7

2nd MATHMOD
VIENNA

2nd MATHMOD
VIENNA

2nd MATHMOD
VIENNA

2nd MATHMOD VIENNA

PROCEEDINGS

late contribution volume

IMACS Symposium on
MATHEMATICAL MODELLING
February 5-7, 1997
Technical University Vienna, Austria

I. Troch
F. Breitenecker
editors

ARGESIM Report No. 11

ISBN 3-901608-11-7

ISBN 3-901608-11-7

ARGESIM Report No. 11

ARGE Simulation News (ARGESIM)
c/o Technical University Vienna
Wiedner Hauptstraße 8-10
A-1040 Vienna, Austria
Tel: +43-1-58801 5386, 5374, 5484
Fax: +43-1-5874211
Email: argesim@argesim.tuwien.ac.at
WWW: <URL:http://argesim.tuwien.ac.at>

Printed by *CA Druckerei*, Vienna, Austria

Preface

The possibility to solve a certain problem and the quality of a solution of a certain task depend essentially on appropriate modelling of the question and of all available information. In some cases, the system under investigation and its behaviour are understood rather well. In such cases an appropriate model will assist in finding a good solution of the problem to be solved. In other situations such a model is primarily intended to help for a better understanding of what is going on in the system. Examples for the first case are many types of design problems being encounterd in typical engineering sytems, such as controller design, design of a production line etc. whereas the request for an improved understanding is often found in connections with non-engineering systems such as biological or medical systems, economic or environmental systems and their control etc.

There is a rather wide consensus that mathematical modelling i. e. abstraction and formalization, is of intrinsic importance. Moreover, most engineers and scientists know quite well that appropriate modelling is far from being easy and that the quality of a design depends strongly on the quality of the model. One of the most important challenges connected with proper modelling is the request to model indeed the given task i. e. all relevant information, restrictions, demands etc. In control engineering not only a model of the plant and constraints on relevant physical variables must be put in a mathematical form but also other requests such as that the resulting mathematical control law must allow for implementation with a certain type of equipment etc.

By now, considerations such as these are accepted by practically all people involved in solving problems by using mathematical methods, no matter whether they work at a scientific institution or in an industrial environment.

However, the area of application determines to a certain extent the knowledge of basic modelling principles, preferences of modelling approaches, of methods for model simplification or for parameter estimation etc. Moreover, many things are discovered repeatedly. Therefore, a conference having mathematical modelling as its center will allow for a fruitful and stimulating exchange of ideas. Consequently, the second IMACS symposium on Mathematical Modelling (2nd MATHMOD) is devoted to the mathematical (or formal) modelling of all type of systems no matter whether the system is

* dynamic or static
* lumped parameter or distributed parameter
* deterministic or stochastic
* linear or nonlinear
* continuous or discrete
* or of any other nature.

Consequently, a wide variety of formal models is to be discussed and the term "mathematical model" includes classical models such as differential or difference equations, Markov processes, ARMA models as well as more recent approaches such as bond graphs or Petri nets.

The written versions of the contributions to 2nd MATHMOD Vienna are collected in these Proceedings starting with the manuscripts of the invited lectures. The first survey to be presented deals with recent algorithms for the generation of good random numbers needed for certain modelling approaches. Such numbers will be of interest especially when prediction of a systems behaviour – usually called simulation – is the main goal and when the practitioner has to decide which random number generator will suit his needs best. A survey follows on hybrid systems such as arise in technical systems and where continuous and discrete event dynamics interact. The third invited lecture is concerned with qualitative modelling and surveys the principle lines of current research and explains the main ideas of an automata-theoretic approach being successfully used in supervisory control.

Then follow contributed papers – which were selected for presentation by a reviewing process based on submitted extended abstracts in which the members of the IPC took active part together with groups of papers contributed upon invitation of a session organizer. The volume concludes with the abstracts of posters being on display during the conference. All these contributions were colleted and arranged in sessions according to their main thematic point:

* Fuzzy and Qualitative Modelling
* Automation of Modelling and Bond Graphs
* Petri Nets and Discrete Event Modelling
* Identification
* Software Tools
* Modelling in Practice
* General Engineering Applications
* Traffic Modelling
* Electrical Systems
* Mechanics and Mechatronics, incl. Robotics
* Automatic Control
* Physical Applications
* Environmental Systems
* Biology and Biotechnical Engineering
* Economic and Social Systems
* Theoretic Aspects

Such a grouping is by no means easy because many contributions address several different aspects in a balanced manner. Therefore, the arrangement chosen for this volume follows rather closely the one of the conference where also time limitations had to be observed.

The editors wish to express their sincere thanks to all who have assisted them by making the idea of this symposium known within the scientific community by acting as sponsor or cosponsor, who have assisted them in the reviewing process and have done a good job by putting together special sessions devoted to one main theme. Last but not least the editors thank Creditanstalt-Bankverein for their generous support for the production of these Proceedings.

Vienna, January 1997 I.Troch, F. Breitenecker

Contents

Contributed Papers

Evaluation of DECT-systems by simulations 1
M. Scheibenbogen (Aachen, D)
Some non-linearities in bioeconomic modelling of biological resources 7
K. S. Chaudhuri (Calcutta, India)
The mathematical model of long river and neural network in estimation process 13
T. Kwater, Z. Kedzior, R. Pekala (Rzeszow, PL)

Posters

Modeling aspects for parallelizing continuous simulation tasks 21
R. Schachinger (Wien, A)
The Flexible-Production-Analyzer (FPA) 22
N. Kraus, R. Schachinger (Wien, A)
Modeling of systems with events by means of the model interconnection concept of the simulation system mosis 23
G. Schuster, J. Plank (Wien, A)
Simulation of stress and strain distribution in a snow cover by FEM (finite element method) 24
S. Wieshofer, O. Kautzky, K. Kleemayer (Wien, A)
Optimal management of railway under special consideration of technology and personell management 25
P. Kralicek (Wien, A)
Parallel simulation programs optimum partitioning 26
A. Uljanov (Wr. Neustadt, A)

Index of authors

27

Evaluation of DECT-systems by simulations

Markus Scheibenbogen
Kommunikationsnetze. Rhein. Westfälische Technische Hochschule Aachen
Kopernikusstr.16. 52054 Aachen
EMail: msc@comnets.rwth-aachen.de
WWW: `http://www.comnets.rwth-aachen.de/~msc`

Abstract — **This paper deals with the evaluation of DECT-systems. It shows the variety of the different aspects of evaluation, which can be dealt with simulations. In this paper the evaluation of capacity for outdoor DECT-systems under different propagation conditions and the performance of the ISDN-64kbit/s bearer service is shown. These two examples shows on the one side a full-system evaluation and on the other a single equipment evaluation.**

1 Introduction

In the discussions about the liberalisation of communication markets in Europe. DECT was identified as one of the most promising systems for outdoor applications like the Wireless Local Loop (WLL) and Personal Communication Systems (PCS). From an operators point of view there are still some open questions like the available capacity, and the behaviour of the system in different environments. These are aspects, which adress the whole system (e.g many base- and mobilestations). On the other side operators would like to know the quality (e.g resulting bit error rates for special services) of the services under given conditions. This paper focuses on both topics:

- capacity of one operator in dependency of different environmental situations (using different propagation models)
- ability of DECT to carry ISDN-traffic (especially for the 64 kbit/s ISDN-bearer service).

Further simulation results concerning the suitability of DECT in outdoor environments can be found in [6]. In this paper also the competition between differnet operators is analysed.

Limitations to the carrying of traffic

In a system which is based on radio-waves you have to distinguish between two different situations limiting the capacity.

- trunk limitation
- interference limitation

In the following these two types of limitation shall be explained: Trunk limitation means that the physical channels of a transceiver are missing, but that there are existing free time-/frequency slots in the allocated bandwidth. The blocking probability of such a situation can be examined with the Erlang-formula. Contrary to this is the interference limitation. This bound of capacity limits the available channels due to the limited bandwidth. Therefore in this case there are existing free channels at the transceiver but there is no free time-/frequency slot in the spectrum. The aim of this investigation was to find these physical limits for the DECT-system.

2 Simulation model for full system evaluation

The simulations were done with a simulator, which was developed in the last years at the chair of communication networks [4]. One important part of a simulator for mobile radio systems is the propagation model. For the simulations an ETSI-propagation model for RLL systems from [1] was used. This model

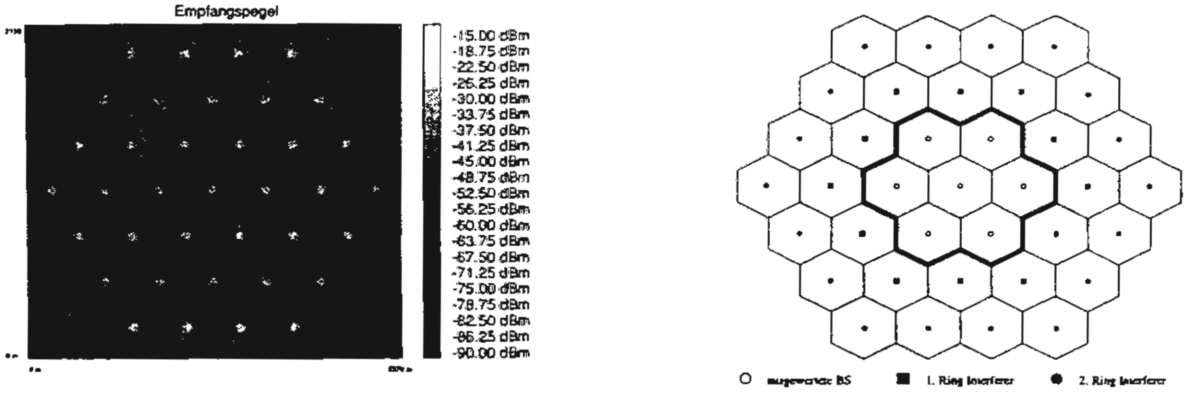


Figure 1: Scenario used for simulations

does not take into account the topology and the morpho-structure of the environment. For the different morpho-structures different models should be used. For cluttered areas formula 1 is valid.

$$L_{path} = 58 + 10\gamma \cdot (\log_{10}(d) - 1) \qquad \gamma = 3.5 \tag{1}$$

Equation 2 should be used for areas like a suburban (semi cluttered) one

$$L_{path} = 58 + 10\gamma \cdot (\log_{10}(d) - 1) \qquad \gamma = 3.0 \tag{2}$$

and for open areas equation 3 is valid. This propagation model is also used for RLL-systems above the roof-tops.

$$L_{path} = 53 + 10\gamma \cdot \log_{10}(d) \qquad \gamma = 2.0 \tag{3}$$

In the simulations only equations 3 and 1 were used. The propagation condition in our scenario is depicted in figure 1. This paper doesn't declare that a pathloss model is correct or should be used. The intention is to show the influence of different propagation environments and the impact on the capacity.

Parameter of the evaluation

The following parameters were evaluated for the DECT-system: Number of blocked setup trails respectively number of interrupted handovers. With these two parameters the *Grade of Service, (GOS)* of the system can be determined, which is an important parameter of the quality of a radio network.

$$GOS = \frac{blocked\,SetupTrials + 10 \cdot dropped\,Calls}{Number\ of\ Calls} \tag{4}$$

A small value of the Grade of Service means a good Grade of Service for the system, because nearly no failed setups occur and no calls will be dropped.

Number of transceiver	Traffic/RFP for Free-Space	Traffic/RFP for ETSI	theor. carryable Traffic
1	4.9	7.56	5.8
2	4.8	13.0	15.3

Table 1: Carryable traffic with different propagation models for GoS 1%

Results of the Simulation of one single large DECT-system

The following investigations of analysing the capacity and the Grade of service of DECT-systems were done in a hexagonal scenario of 37 RFPs. The scenario and its illumination are depicted in figure 1.

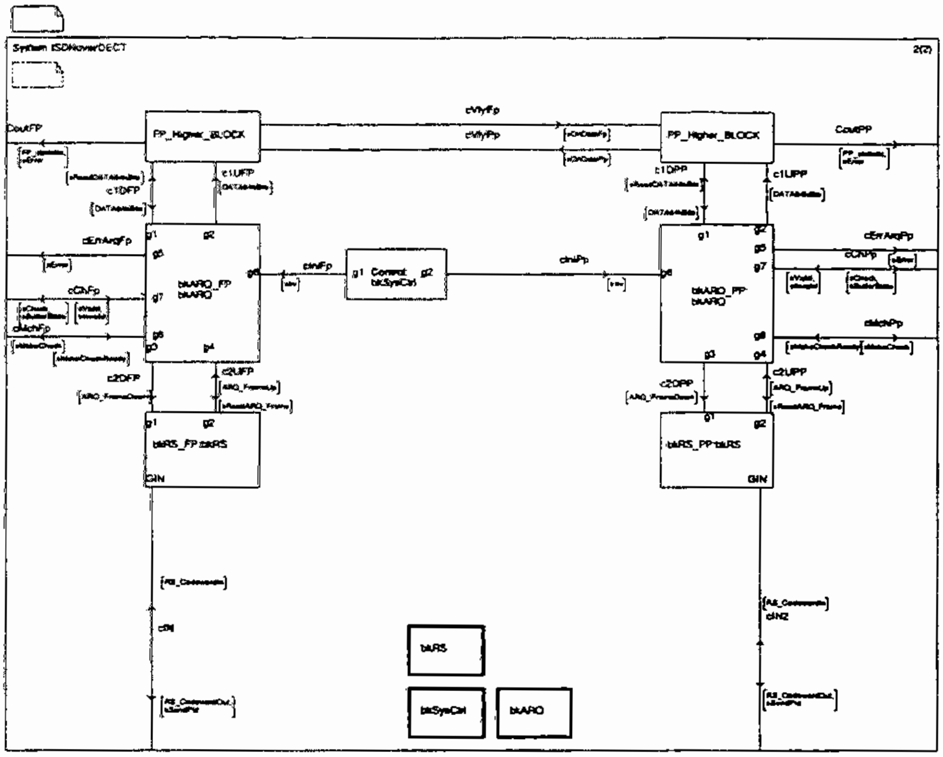


Figure 2: SDL-system

Only the innermost seven basestations were evaluated in order to ensure that effects of the border are negligible. Parameter of the investigations was, beside the offered traffic, the number of transceivers installed at every Radio Fixed Part. It was also investigated the influence of sectorisation on the capacity of DECT-systems, and different antenna-types [6]. Results with the use of relays can be found in [5].

In table 1 it is shown that with one transceiver (ETSI-propagation) the system is trunk limited, because with the use of two transceiver you can increase the carryable traffic per Radio Fixed Part. The traffic can not be increased with the use of three transceivers. Therefore the system is then not trunk limited but interference limited. This is a physical limit. You can not enlarge the traffic, not even by using additional transceivers. The limit of the carryable traffic is at 13 Erlang/RFP. On the other side the physical limit (interference limit) is reached with free-space propagation with only one transceiver.

3 DECT and ISDN (single equipment evaluation)

In this section the standards of ISDN-services over a DECT-airinterface will be described very shortly and the 64kbit/s ISDN-service over the DECT Air-interface will be evaluated. Contrary to the full system simulation in section 2 here only one RFP and one Portable Part is taken into account. Therefore the simulation model is quite different and regarding the protocols much more specific and detailed. In figure 2 the system specified in SDL is shown. The physical channel was modelled as a Bernoulli-channel or as a Gilbert-channel.

End System und Intermediate System

The ETSI has standardised two different standards for DECT and its interworking with ISDN. The two solutions are the so called *End System* ([3]) and the *Intermediate System* ([2]). The difference could shortly be described as follows. In the End System the Radio Fixed Part and the Portable Part are forming an ISDN-endterminal. Thus it is mainly for the end-user. It transports the different ISDN-services, but on the side of the Portable Part no tranparent ISDN is available. The Intermediate System forms more or less the function of a real transport network for ISDN. Thus on the side of the Portable Part transparent ISDN is available. This system is neccessary for an WLL-operator, who wants to provide ISDN.

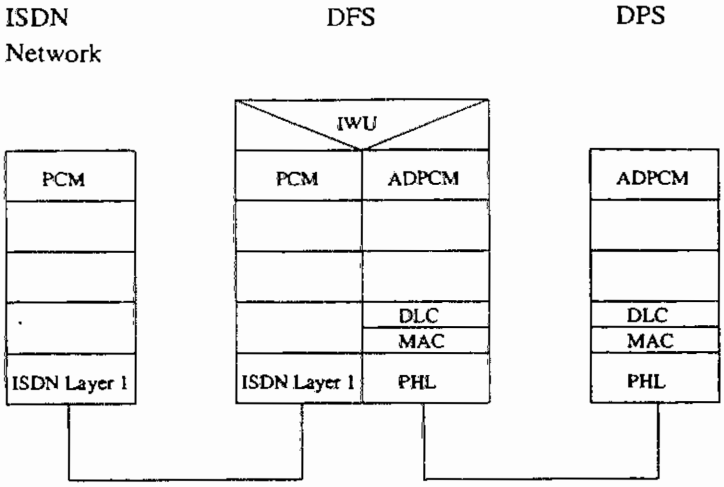


Figure 3: Speech service

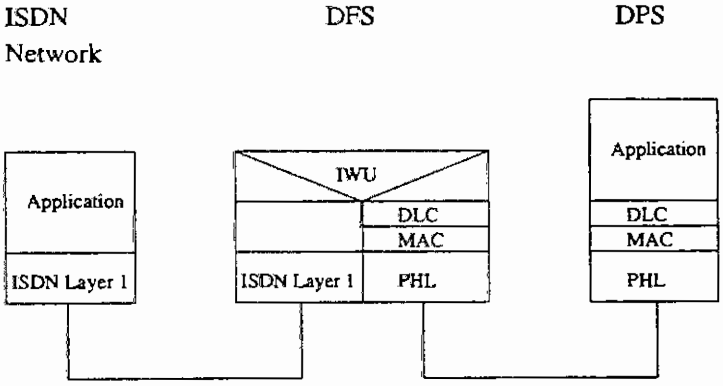


Figure 4: 64 kbit/s-service

ISDN-services (Speech and Data)

In figure 3 the protocolstack of the U-plane for the ISDN-speech service can be seen. The speech will be coded from PCM to ADPCM and thus the data rate can be reduced from 64kbit/s to 32 kbit/s. In figure 4 tbe 64kbit/s Data-service is depicted. The amount of data can't be reduced, therefore no form of recoding is used. Due to the reason that a mobile radio channel has a higher bit error rate the brutto data rate between the Radio Fixed Part and the Portable Part has to be higher than 64kbit/s to add Forward Error Correction (FEC) and to facilitate Automatic Repeat Request (ARQ). For the data-service a double slot is used. The physical packet (P80) provides an 80kbit/s unprotected service. A buffer on sending and receiving side was introduced to facilitate an ARQ-procedure. This buffer will result in a fixed delay of 80ms, because it is able to buffer 640 bytes, which is equvialent to 8 double slots of data. In a normal situation (the buffer is filled) 72kbit/s of the 80kbit/s is used. The unused 8kbit/s are filled up with zeros. The difference between 64kbit/s net data rate and the 72kbit/s used transmission rate resulted in the additional FEC.

In figure 5 one result of the simulations can be seen. The resulting BER on the 64kbit/s data stream is shown versus the BER on the radio channel. To ensure a ISDN-service the BER has to be about 10^{-6} or better. The normal DECT speech service is a able to work down to radio channel quality of about 10^{-3}. Higher bit error rates will normaly result in a bandover (intercell or intracell). Therefore it seems to make sense, that also the 64kbit/s service should work in environments down to a BER of 10^{-3}. In the simulations this could be shown.

4 Summary

It was shown, that simulations are helpful and necessary to evaluate the behaviour of complex radio-systems or the complex combination of FEC and ARQ. Especially radio system always influence itself

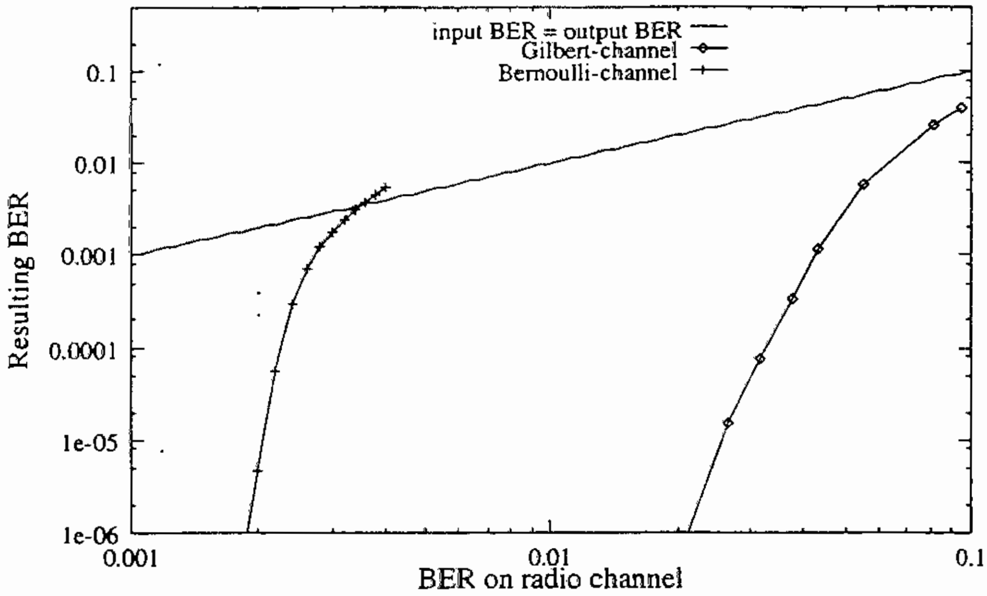


Figure 5: Bit error rates

very much, so that simulations are essential to get a feeling for the system. The simulations were done with simulators, which were developped at the chair for communication networks. The results are twofold: On the one side the capacity of the DECT-system could be evaluated and is about 13Erlang/RFP with omnidirectional antennas. On the other side it could be shown that the recently standardised DECT-ISDN 64kbit/s-service is able to work in an environment with a BER of 10^{-3} on the radio channel and still provide ISDN-quality.

5 References

[1] ETSI Technical Committee Radio Equipment, Systems TC-RES. *ETR139, Radio in the Local Loop (RLL).* 06921 Sophia Antipolis Cedex - France, November 1994.

[2] ETSI. *Radio Equipment and Systems (RES); Digital Enhanced Cordless Telecommunictions (DECT); DECT/ISDN Interworking for Intermediate System Configuration; Interworking and Profile Specification.* proposed Standard ETS 300 XXX, European Telecommunications Standards Institute, July 1996.

[3] ETSI. *Radio Equipment and Systems (RES); Digital European Cordless Telecommunictions (DECT) and Integrated Services Digital Network (ISDN) Interworking for end system configuration Part 1: Interworking specification.* Standard ETS 300 434-1, European Telecommunications Standards Institute, April 1996.

[4] Holger Hußmann. *Performance Evaluation of the DECT radio resource management.* In *Aachener Kolloquium Signaltheorie*, March 1994.

[5] I Lenzen, C. Plenge. *Die Entwicklung und Bewertung eines Relais-Konzeptes als Erweiterung des DECT-Standards.* In *2. ITG-Fachtagung Mobile Kommunikation '95*, pp. 379–387, Neu Ulm, D, September 1995.

[6] M. Scheibenbogen. *Suitability of DECT for outdoor applications.* In *Proceedings 7th Nordic Seminar on Digital Mobile Radio Communications DMR VII*, Kopenhagen, Danmark, October 1996.

SOME NON-LINEARITIES IN BIOECONOMIC MODELLING OF BIOLOGICAL RESOURCES

K. S. Chaudhuri
Department of Mathematics, Jadavpur University,
Calcutta-700 032, India.

ABSTRACT

This article focuses attention on the various types of nonlinearities that may occur in bioeconomic modelling of exploitation of biological resources. Nonlinearities generally creep into the model through (1) growth rate functions, (2) intraspecific competition within a species, (3) interspecific competition amongst different species, (4) the production function of a resource stock, (5) revenue function, (6) cost function, (7) demand function and (8) age structure. These nonlinearities lead to nonlinear dynamical systems and nonlinear control problems for harvesting. Implications of these various nonlinearities are discussed.

1.INTRODUCTION

The dynamics of an exploited biological resources (e.g. fish, deer, bird, tree, etc.) can be mathematically described by a simple looking differential equation of the form [1] :

$$\frac{dx}{dt} = F(x) - h(t) \tag{1}$$

where $x = x(t)$ denotes the size (or biomass) of the resource population at any time t,
$F(x)$ = the natural growth rate of the population,
$h(t)$ = the rate at which the population is harvested at any time t.

We find that the population increases, decreases or maintains a constant level according as $F(x) >, <$ or $= h(t)$. The case $F(x) = h(t)$ implies that the population can be maintained or sustained at a fixed level even after harvesting it at a rate equal to the natural growth rate. This is why we call the natural growth to be the SUSTAINABLE YIELD (SY).

The harvest (or catch) rate $h(t)$ is usually taken in the form $h(t) = qE(t)x(t)$ on the basis of the *catch-per-unit-effort*(CPUE) hypothesis [1] where E denotes the effort and q is a constant known as *catchability coefficient*. In a more general form $h = Q(E, x)$ where the function Q is the *production function* for the given resource industry. In order to have a linear optimization problem, we often take $h = EG(x)$ where $G(x)$ is a non-decreasing function of x.

The net economic revenue R for an input of effort in time Δt is given by $R\Delta t = [ph - cE]\Delta t = [p - c(x)]h\Delta t$
where $c(x) = \frac{c}{G(x)}$ is the unit harvesting cost when the stock level is x,
p = constant price per unit biomass of the harvested resource and
c = constant cost per unit of harvesting effort.

Taking $\delta(> 0)$ to be the uniform rate of discount, the objective of the management of the given resource industry is to maximize the *present value of all future revenues* obtainable from the resources, given by the present value integral

$$PV = \int_0^\infty e^{-\delta t} R(x, E)dt = \int_0^\infty e^{-\delta t}(p - c[x(t)])h(t)dt \tag{2}$$

subject to the *state equation*(1) and the constraints $x(t) \geq 0$ and $0 \leq h(t) \leq h_{max}$ where $h_{max} = h_{max}(x, t)$ is a nonnegative function representing the maximum harvest capacity. We may solve this maximization problem either by using the techniques of *calculus of variation* or by using *Pontryagin's Maximum Principle*[2] with $x(t)$ as the *state variable* and $h(t)$ or $E(t)$ as the *control variable*. We then obtain the *optimal equilibrium population level* $x = x^*(t)$ as the unique solution, if it exists, of the equation

$$F'(x) - \frac{c'(x)F(x)}{p - c(x)} = \delta \tag{3}$$

The optimal path is the *most-rapid (or bang-bang)* approach policy given by $h^*(t) = h_{max}, F(x^*)$ or 0 according as $h >, = or < x^*$.

2. TYPES OF NONLINEARITIES

We now discuss some possible types of nonlinearities in single species models.

(A) NONLINEARITIES IN THE GROWTH FUNCTION

The growth function $F(x)$ becomes nonlinear in x when we take *intraspecific competition* into account, some examples being the *Logistic Growth Law* $F(x) = rx(1 - x/k)$, *modified Logistic Growth Law* $F(x) = rx^\alpha(1 - x/k), \alpha > 0$ and the *Gompertz Growth Law* $F(x) = rx\ln(x/k)$. If we write $F(x) = xf(x)$ where $f(x)$ is a single-valued function of x, there are two types of growth. In the first type in which $f(x)$ is a monotonically decreasing function of x, the growth process is of the nature of pure *compensation* or *feed back* as in the *Logistic Growth*. The second type in which $f(x)$ is non-monotonic, is encountered in populations (e.g. schools of fish, colonial birds, etc.) having a team behaviour and group defence against predators' attack. As a result of this group defence, the growth rate increases with x which is known as *depensation*. When the population level becomes considerably high, the general lack of resources causes the growth rate to decline. This type of growth pattern is described by the so called *Alle Curve* [3] having several steady states.

(B) NONLINEARITIES IN THE CATCH-RATE FUNCTION

The catch-rate function $h(t) = qE(t)x(t)$ based on the CPUE hypothesis has several unrealistic features:

(i) random search for fish, (ii) equal likelyhood of being captured for every fish, (iii) unbounded linear increase of h with E for a fixed x and (iv) unbounded linear increase of h with x for a fixed E. These severe restrictions are largely removed [4] in the functional form $h(t) = qE(t)x(t)/(aE(t) + bx(t))$ where a, b are positive constants. Here $h \to (q/a)x$ as $E \to \infty$ for a fixed value of x and $h \to (q/b)E$ as $x \to \infty$ for a fixed value of E. Thus h exhibits saturation effects with respect to both the effort level and the stock abundance. The parameter a is propotional to the ratio of the stock level to the catch-rate at higher levels of effort. The parameter b is propotional to the ratio of the effort level to the catch rate at higher stock levels.

The Cobb-Douglas form of the production function of a biological resource is
$h = Q(E, x) = Ax^{\alpha}E^{\beta}$ where $A > 0, 0 < \alpha < 1, 0 < \beta < 1$.

Marginal productivity of the effort E is
$MP_E = \frac{\partial Q}{\partial E} = \frac{\beta}{E}Q > 0$; however, $\frac{\partial}{\partial E}(MP_E) = \frac{\beta(\beta-1)Q}{E^2} < 0$ when $0 < \beta < 1$. Thus the productivity of the resource industry increases with respect to E; but the rate of increase of productivity decreases with E. This behaviour is quite logical in the exploitation of a biological resource. On the otherhand, marginal productivity of x is $MP_x = \frac{\partial Q}{\partial x} = \frac{\alpha Q}{x} > 0$ and $\frac{\partial}{\partial x}(MP_x) = \frac{\alpha(\alpha-1)Q}{x^2} < 0$ if $0 < \alpha < 1$. For a given E, the productivity of the resource industry increases with the increase of the stock level. This rate of increase of productivity should go on increasing if the stock level also goes on increasing. Thus $0 < \alpha < 1$ is inappropriate here and the Cobb-Douglas production function does not fully fit in biological resources. The appropriate form is, therefore, $h = Ax^{\alpha}E^{\beta}$ with $A > 0, \alpha > 1$ and $0 < \beta < 1$. This leads to a nonlinear control problem for determining the optimal harvest policy.

(C) NONLINEARITY IN THE REVENUE FUNCTION :

Let the revenue be a nonlinear function $R(h)$ with $R(h)$ a smooth, convex, non-negative function of $h \geq 0$. For example, if the price p of fish depends on the rate of output h, we then have $R(h) = hp(h)$. If we neglect costs of harvesting, our problem is to maximize $J(h) = \int_0^\infty e^{-\delta t}R(h)dt$ subject to (1).

The Hamiltonian is
$H = e^{-\delta t}R(h) + \lambda(t)(F(x) - h)$. If the control constraints are not binding, we have $\frac{\partial H}{\partial h} = 0 \to \lambda = e^{-\delta t}R'(h)$. Using this result in the adjoint equation $\frac{d\lambda}{dt} = -\frac{\partial H}{\partial x}$, we

have $\frac{dh}{dt} = \frac{R'(h)}{R''(h)}[\delta - F'(x)]$ which alongwith (1) are necessary conditions for the optimal control $h(t)$ and the corresponding response $x(t)$. Assuming $R'(h) > 0$ and $F''(x) < 0$, these two equations give a unique equilibrium point (x^*, h^*) that is determined by $F'(x^*) = \delta$ and $h^* = F(x^*)$. The x-isocline is the curve $h = F(x)$ whereas the h-isocline is the vertical line $x = x^*$ where $F'(x^*) = \delta$. The intersection (x^*, h^*) of these two isoclines is the optimal equilibrium solution and it is a saddle point [1]. It can be easily ascertained through some analysis of the trajectories that the optimal approach to equilibrium is no longer a bang-bang approach, as it was in the linear harvesting problem. For the case of an infinite time horizon, the optimal harvest rate $h(t)$ is obtained by following the seperatrix that begins at $x = x_0$ and by approaching the equlibrium $x = x^*$ asymptotically. In the linear model, the revenue was assumed to be directly proportional to the harvest and the optimal policy consisted of adjusting $x(t)$ as rapidly as possible to the optimal level $x^*(t)$. The questions of a fall in the price due to a high rate of harvesting and a hike in the price due to low harvests were not considered. The asymptotic approach in the nonlinear model here reflects a more gradual approach engendered by market reactions to the harvest rate.

(D) NONLINEARITY IN THE COST FUNCTION

Let the cost of harvesting be given by a function of the form $C(x, h) = \psi(x)\phi(h)$ where $\phi(h) > 0, \phi'(h) > 0$ and $\phi''(h) > 0$. This implies that the marginal cost is an increasing function of the harvest rate h. The objective function for socially optimal management is

$J(h) = \int_0^\infty e^{-\delta t}[U(h) - C(x, h)]dt$ where $U(h)$ is the *total social utility of consumption.* It can be easily ascertained that any optimal equilibrium solution (x^*, h^*) must satisfy the relation

$$F'(x^*) - \frac{\psi'(x)\phi'(h^*)F(x^*)}{U'(h^*) - \psi(x^*)\phi'(h^*)} = \delta \text{ where } h^* = F(x^*).$$

The corresponding *discounted supply curve* is obtained by writing $p = U'(h^*)$ and solving this equatlon for $p : p = \psi(x^*)\phi(h^*) - \frac{\psi'(x^*)\phi'(h^*)F(x^*)}{\delta - F'(x^*)}$.

Depending upon the nature of this discounted supply curve, the following results are obtained [1] :

(i) The harvest rate is *choked off* at a point h'which is not large enough to cause biological overexploitation.

(ii) Once overexploitation occurs due to any reasons whatsoever, the resource may be trapped near a stable equilibrium h'' which is very small and $h'' < h'$.

(iii) Bionomic instability may arise from nonlinear cost factors as well as nonlinear demand factors. When both nonlinearities are present, instability is even more likely to occur.

(E) NONLINEARITIES IN THE INTERSPECIFIC RELATIONSHIPS

Exploitation of multispecies biological resources, especially marine fisheries, is a complicated job because a fishing vessel, no matter how modern it is, can not harvest exclusively only a target species of fish. Incidental catches may lead to severe depletion of nontarget species because of complexity in the marine ecosystem ; a single species may be predatory on several species while serving as prey for other species. Two species may even be simultaneously predator and prey of each other at different life stages, an example being *Atlantic cod (Gadua morhua)* which feeds on *Capelin (Mallotus villosus)* which, in turn, feeds on cod eggs. Many marine fish species are also self predatory. Taking the simplest case of a two-species fishery, we may take the growth equation as

$\frac{dx}{dt} = F(x,y) - h_1(t)$ and $\frac{dy}{dt} = G(x,y) - h_2(t)$ where $F(x,y) = f(x)+yp(x)$, $G(x,y) = g(y) + xq(y)$. Here f(x), g(y) are growth rates of two species and p(x), q(y) are their functional response.

For linear functional responses, $F(x,y) = f(x)+\alpha xy$ and $G(x,y) = g(y)+\beta x$ y. The case $\alpha < 0, \beta < 0$ gives the model of *interspecific competition.* The case $\alpha < 0, \beta > 0$ gives rise to a *prey-predator* model. The third case $\alpha > 0, \beta > 0$ stands for mutualism or symbiosysis.

One of the important aspects of study of such a system is to examine its *global stability.* When $F(x,y)$ and $G(x,y)$ are nonlinear functions of x and y, we may prove global stability of a unique positive equilibrium by several methods. One such method is to construct a *Lyapunov function* and establish the global stability by *LaSalle's invariance principle* [5]. The second method is to employ *Dulac criterion* to eliminate the existence of periodic orbits and prove the global stability by *Poincare-Bendizson theorem* ([6], [7]). The third one is the geometric *method of comparison*([8],[9]) which compares the trajectories of the system with that of an auxiliary system obtained by mirror reflection. The fourth method ([7] [8], [10]) is the *method of limit cycle stability analysis.* The idea of this method is to prove the nonexistence of periodic solutions by contradiction.

Another important factor that contributes to nonlinearity of a system is the consideration of age structure in the population.

REFERENCES

1. Clark, C.W., Mathematical Bioeconomics : The Optimal Management of Renewable Resources. John Wiley and Sons, New York, 1990.

2. Pontryagin, L.S., Boltyanskii, V.S., Gamkrelidze, R.V. and Mishchenko, E.F., The mathematical Theory of Optimal Processes. Wiley-Interscience, New York, 1962.

3. Svirezhev, Y.M. and Logofet, D.O., Stability of Biological Communities, Mir Publishers, Moscow, 1983.

4. Ganguly, S. and Chaudhuri, K.S., Regulation of a single species fishery by taxation. Ecological Modelling, 82(1995), 51-60.

5. Hale, J.K., Ordinary Differential Equations. Wiley-Interscience, New York, 1969.

6. Hsu, S.B., Hubbell, S. P. and Waltman, P., Competing predators. SIAM J. Appl. Math., 35(1978), 525-617.

7. Kuang, Y., Global stability of Gauss-type predator-prey systems. J. Math. Biology, 28 (1990), 463-474.

8. Cheng, K.S., Hsu, S.B. and Lin, S.S., Some results on global stability of a predator-prey system. J. Math. Biology, 12(1981), 115-126.

9. Liou, L.P. and Cheng, K.S., Global stability of a predator-prey system. J. Math., Biology, 26(1988), 26-71.

10.Butler, G.J., Hsu, S.B., Waltman, P., Coexistence of competing predators in a chemostat. J. Math. Biology, 17(1983), 133-151.

Tadeusz Kwater, Zdzisław Kędzior, Robert Pękala
Rzeszów Pedagogical University, 16 Rejtana str.,
Rzeszów 35-059, Poland

THE MATHEMATICAL MODEL OF LONG RIVER AND NEURAL NETWORK IN ESTIMATION PROCESS

Keywords: *mathematical model of biochemical river pollution, estimation process, Kalman's filter, artificial neural network.*

Mathematical model of the polluted river

Sources of clean water for home and industrial using, which are mostly rivers, still decreases. More sewage discharge into river is a reason of poorer quality of water. In our paper we presented model of river biochemical pollution. Water quality is measured basically by two concentration indicators, dissolved oxygen (DO) and biochemical oxygen demand (BOD). The model also allows to take into consideration another indexes of pollution, i.e.: concentration of chlorides, sulphuric and other. This makes it universal. We considered pollution in which main role takes dissolved oxygen indispensable in oxidation reaction. In these models oxygen balance is applied. The Streeter-Phelps equations describing self cleaning process are used in these models. In basic form the Streeter-Phelps equations give speed of decomposition organic substances and change of level of dissolved oxygen in water. For constant liquid volume these equations are:

$$\begin{aligned} &\frac{d}{dt}x_1 = -k_1 x_1 \\ &\frac{d}{dt}x_2 = -k_2 x_1 + k_3\left(x_{2N} - x_2\right) + a \end{aligned} \qquad (1)$$

where: $x_1[mg/l]$ - biochemical oxygen demand (BOD), $x_2[mg/l]$ - demanded oxygen (DO), $k_1[1/day]$, $k_2[mg/l]$ - coefficients, rate of reaction between BOD and DO, $k_3[1/day]$ natural aeration rate coefficient, $x_{2N}[mg/l]$ - state of saturation DO, $a[mg/l*day]$ - intensity of increasing oxygen from photo-synthesis process or decreasing oxygen drawing by bottoms deposit.

The coefficients depend mostly on temperature, and their values are from 0.1 to 0.4 for k_1, k_2 and from 0.2 to 1 for k_3. Similar dependence show x_{2N} and oscillate from 14.6 to 9.2 in temperature 0^{o} C and 20^{o} C. Some times the coefficient k_2 is taken instead of k_1. Moreover, to ensure better interpretation the state of water pollution the deficiency of dissolved oxygen instead dissolved oxygen is used. This is the difference between actual DO and state of its saturation. It is worth noting that such approach concern closed reservoirs. In order to use such model river can be treated as a cascade of reservoirs with continuos mixing (cascade of chemical reactors). This way of river presentation follows by many simplifications, i.e.: change of BOD and DO is independent on coordinate of river length. Moreover, homogeneity of parameters is taken and it gives model with ordinary differential equations. Such approach is considerate with assumption, that these segments are very short. It gives a lot of segments for long river. From other side, if we assume longer segments than we omit time of flow.

These restrictions can be omitted by using mathematical model of river described by partial differential equations. The long river can be divided on segments for take into consideration of its specific i.e.: sudden change of flow, and change of pollution's. It is shown on Fig.1. Assuming the dividing of river on the sections presuming that ideal mixing occurs at the boundaries of these sections.

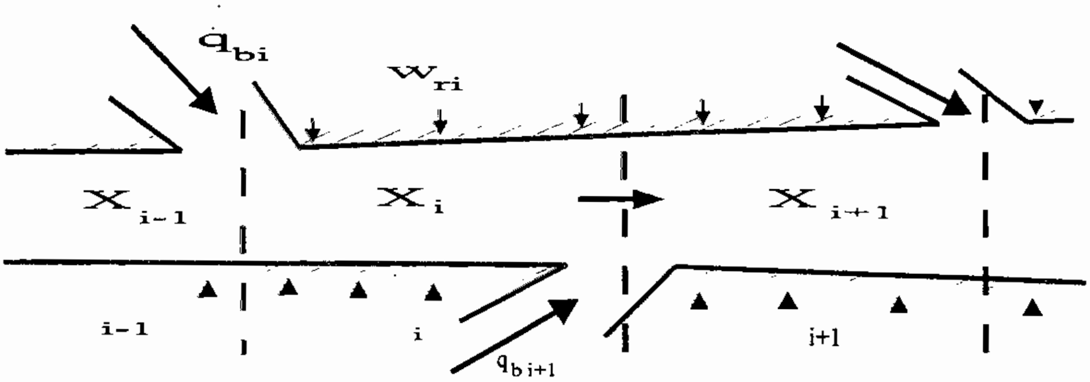


Figure 1. River sections defined by tributary.

If we consider mass balance on section boundaries it provides the boundary conditions for model with distributed parameter system and connects sections in cascade. On the Fig.1. can easy see the larger tributaries and small sewage inflows along the river. Due to previous asumption implicit dependence on temperature all the parameters have been treated as function of length of river and time. It was assumed either, that all coefficients are sufficiently smooth. Equations for one section of model polluted river with BOD and DO are given:

$$\frac{\partial}{\partial t}X_i(z,t)+V_i(z)\frac{\partial}{\partial t}X_i(z,t)=A_i(z)X_i(z,t)+W_{ri}(z,t) \qquad (2)$$

with conditions:

- *boundary condition*: $X_i(0,t)=M_iX_{i-1}(1,t)+W_{bi}(t)$
- *initial condition*: $X_i(z,0)=X_{i0}(z), \quad i=1...N$

Vector of state consist of BOD and DO indicators, i.e.:

$$X_i(z,t)=col\left[x_{1i}(z,t),x_{2i}(z,t)\right], \qquad z\in[0,1], \qquad M_1=0$$

Matrix of state is given as follows:

$$A_i(z)=\begin{bmatrix}-k_{1i} & 0\\ -k_{2i} & -k_{3i}\end{bmatrix}.$$

Elements of vectors W_r and W_b represent disturbances with distributed and boundary character.

According to above assumptions the equation of model river polluted are non-linear. By linearization them around of steady state we get model as a large scale distributed parameter system with boundary coupling between subsystems. Such circumvention more precisely describes biochemical processes in the river but not takes into consideration the diffusion process. It is worth noting that velocity matrix has the same diagonals. It yields to ordinary equation along characteristic line that are Streeter-Phelps equation. Such approach provides the model of polluted river in the separate part of free flow of water. Since the state vector can be consider as DO, BOD around free swim boat. This assumption was taken into our investigations of estimation process. For the estimation process, ordinary differential equations along characteristic line of our model and measurement are necessary. It occurs that BOD measurement requires laboratory service and its time consuming. Denote this measurement include the time delay and are useless in consideration BOD and DO around free swim boat. Just the opposite the DO measurement is simple to realise and is immediate. Consequently we resign from the measurement of BOD indicator. The equation of measurements is as follows:

$$Y(t_k) = CX(t_k) + V(t_k), \quad \text{where} \quad C=col\,[0,1]\,, \quad V(t_k)\text{ -measurements noise} \tag{3}$$

Idea of swim boat does not require dividing river on sections. So, in equation (3) we omit numbering sections of the river. Taking into consideration interpretation of freely swim boat imagine that boat meets measurements of state of water only in marked out points. It means, that we get discrete values in spite of their continuos realisations. For continuos equations (1), which present BOD and DO along characteristics we used discrete values of measurements and we applied Kalman's filter.

Estimation process.

In the estimation process for continuous objects with discrete measurements we can mark out two phases i.e.: filtration and prediction. Filtration is process of estimate generation in moment t_k on the basis of measurements in this moment and previous. In prediction we want to get estimates for the future to next measurement point. Filter with discrete measurements is described by equations:

- **filtration:**

$$\begin{aligned}
&\hat{X}(t_k|t_k) = \hat{X}(t_k|t_{k-1}) + K_F(t_k)\left[Y(t_k) - C\hat{X}(t_k|t_{k-1})\right] \\
&\hat{X}(t_k|t_{k-1}) = \overline{X}_0 \\
&P(t_k|t_k) = P(t_k|t_{k-1}) - K_F(t_k)CP(t_k|t_{k-1}) \\
&P(t_k|t_{k-1}) = P_0 \\
&K_F(t_k) = P(t_k|t_{k-1})C^T\left[CP(t_k|t_{k-1})C^T + V_p(t_k)\right]^{-1}
\end{aligned} \tag{4}$$

where: $\hat{X}(t_k|t_k)$ - estimate in moment t_k, obtained on the basis of measurements $Y(t_0)...Y(t_k)$, $P(t_k|t_k)$ - covariance of estimation error, $K_F(t_k)$ - amplification coefficient,

- **prediction**

$$\begin{aligned}
&\frac{\partial}{\partial t}\hat{X}(t|t_k) = A(t_k)\hat{X}(t|t_k), \qquad \hat{X}(t|t_k) \\
&\frac{\partial}{\partial t}P(t|t_k) = P(t|t_k)A(t)^T + A(t_k)P(t|t_k) + W_r, \qquad P(t|t_k)
\end{aligned} \tag{5}$$

where: $\hat{X}(t|t_k)$, $P(t|t_k)$ - means adequately estimate and covariance of estimation error predicted for $t>t_k$, W_r - covariance of disturbances W_r.

Equations (4) and (5) were used in estimation process. Measurements were obtained by simulation mathematical model of polluted river (1). Runge-Kutta-Fehlberg's of 4-th order fomula was used to solve ordinary differential equations . Figure 5 shows obtained results of the estimation and prediction process. It easy to see that estimates follow on simulated states. Characteristic feature of received estimates are non-continuity in measurement moments. It is clear, that the better quality of estimation process can be obtained by increase measurement frequency (it means more measurement devices along river).

Neural networks method.

Kalman's filter generates estimates that are strongly oscillate around state of model while the level of noise increases. It seems, this disadvantage may be omit by using artificial neural networks for monitoring polluted river. It is important because level of disturbances is difficult to determine. The estimation system of state of river quality with neural network is shown on Fig.2. Neural network with loop-feedback was used.

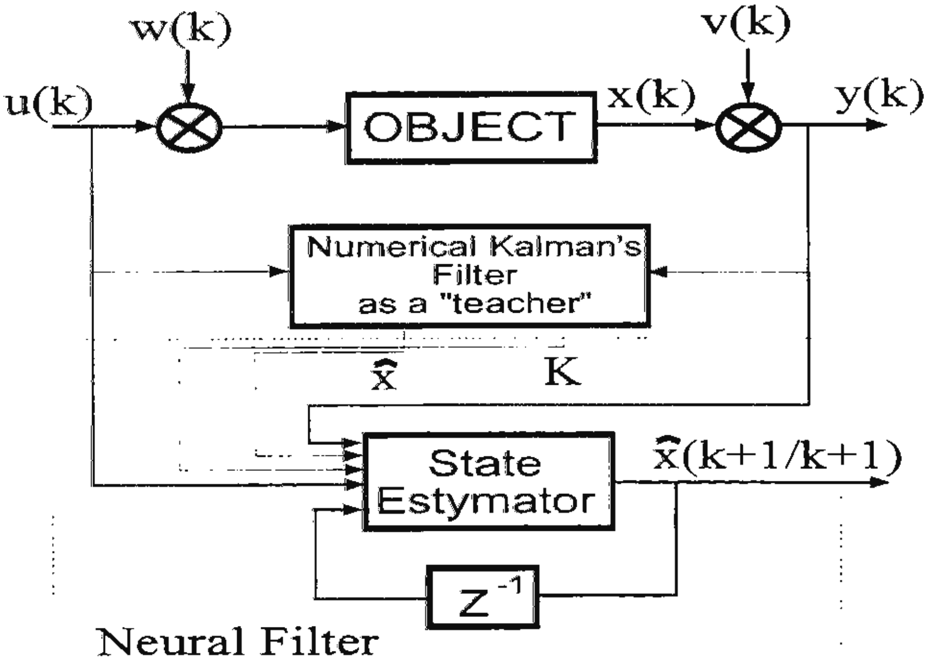


Figure 2. Scheme of estimation system.

In the feedback loop ideal estimate delay of one step is given. Measurements DO from mathematical model were given to the neural network learning process. The initial conditions of estimate state and amplification coefficient from numerical Kalman's filter were given. It is worth nothing, that neural network with loop-feedback can to determine variables of state which are not obtained from measurement. In our case we get BOD estimate. In investigations we applied neural network with two layers neurones and receptors layer. We present only one scheme of neural network with 25 hidden neurones for estimation process of polluted river.

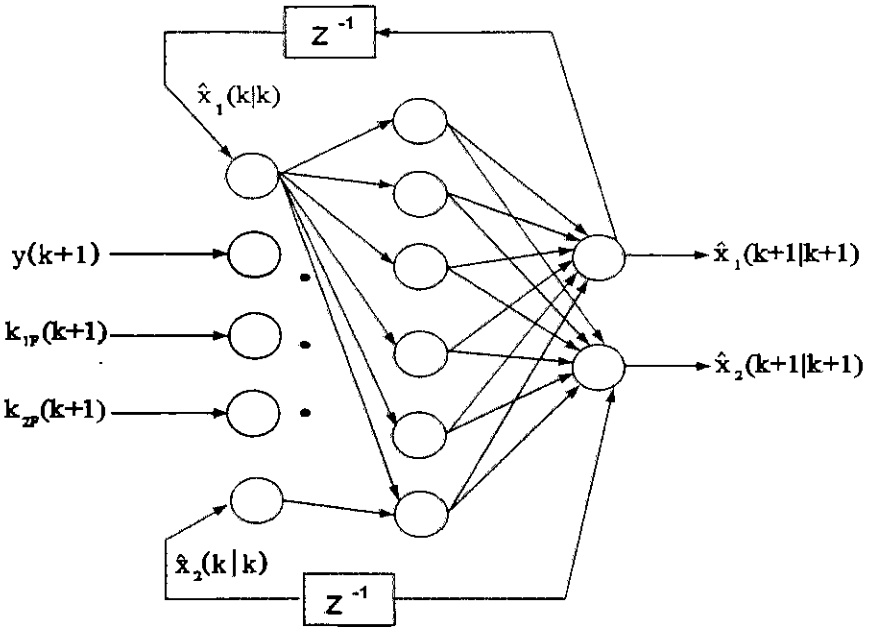


Figure 3. Architecture of neural network used in estimation system.

Experiments.

Efficiency of neural network learning process is given on Fig. 4 On the vertical axes indicator of network efficiency called Sum Squared Error (SSE) is given. It depends on number of epochs (number of pattern presented to the network).

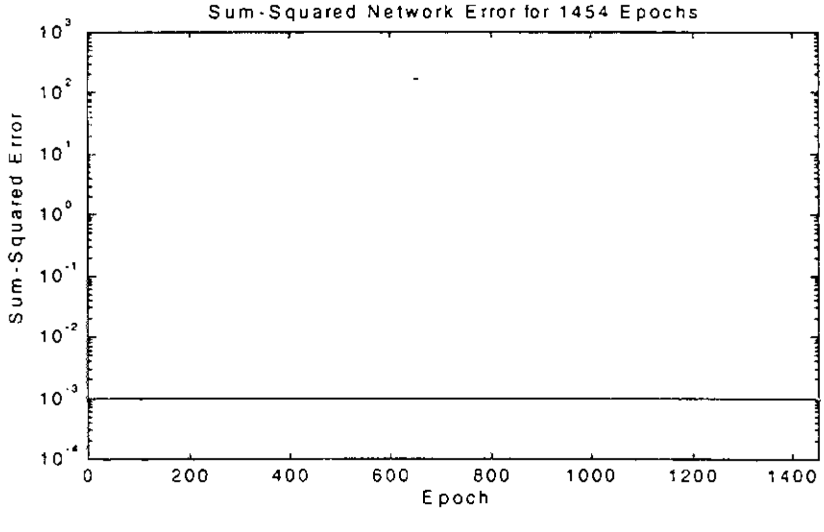


Figure 4. The network efficiency in the learning process.

As easy to see SSE quickly comes down and next stabilise on low level of value.

Figure 5 presents comparison neural network estimates with estimates obtained from numerical Kalman's filter.

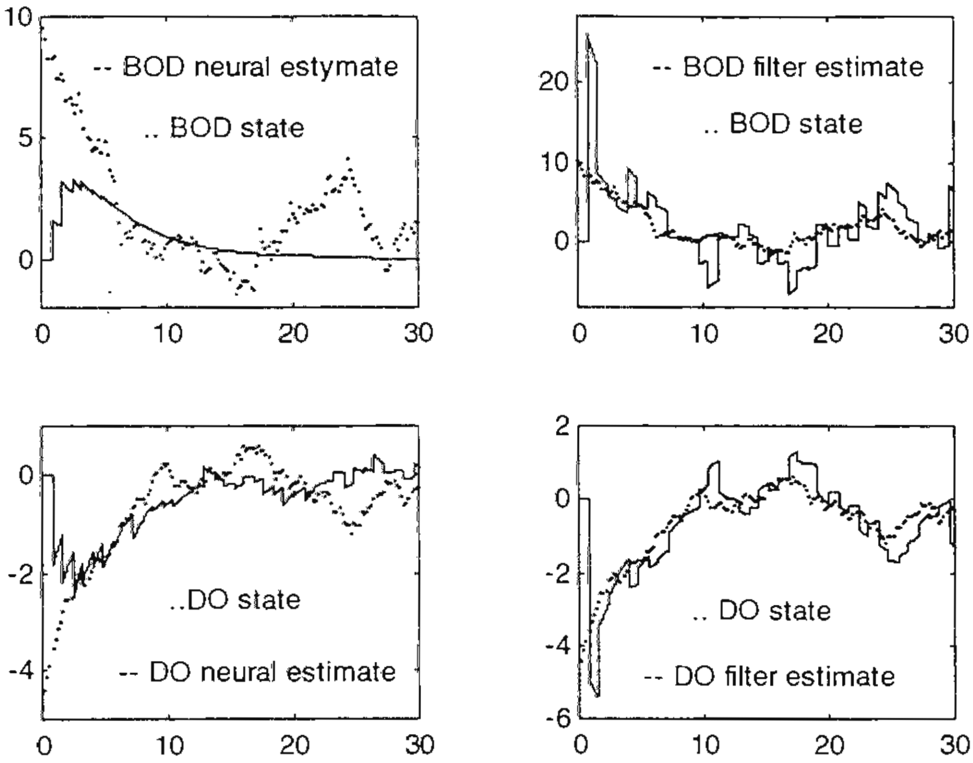


Figure 5. The results of simulations of state estimation.

Neural network generates more gently estimates. Investigations shows, that strong disturbances do not have significant influence on fluctuation of estimates around the states. This feature is reach out in the case of numerical Kalman's filter.

References.

[1] Anderson B, Moore B.: *Optimal Filtering* WNT: Warsaw,1984 .

[2] Demuth H, Beale M.: *Neural Networks Toolbox User's Guide* The Math Works Inc.

[3] Hertz J, Krogh A, Palme G.R.: *Introduction to the theory of neural computation,* Massachusetts, 1991.

[4] Kwater T., Pękala R., Twaróg B.: *Monitoring of river quality by neural network,* 1-st Modeling International Modeling School Krym 1996.

[5] Rinaldi B., Soncini-Sessa R., Stehfest H., Tamura H.: *Modeling and control of river quality* Mc Graw-Hill Inter Book Company 1979.

[6] Tadeusiewicz R. : *Neural Networks* PWN: Warsaw,1993.

Poster Abstracts

MODELING ASPECTS FOR PARALLELIZING CONTINUOUS SIMULATION TASKS

Rupert Schachinger
Hofzeile 21/1/12, A-1190 Vienna

Abstract

The past few years have seen a revolution in the way parallel computing is practiced. Computer systems with up to 128 parallel CPU's have been developed and hardware specialists work on developing systems with more than 1000 processors respectively large networks of single processor machines which can perform the same tasks.

The great advantage of parallel computers (one can process several independent tasks on several processors) has led to new software development, parallel languages. FORTRAN and C have (partially) reached a "parallel level".

One of the main problems in continuous simulation is to solve an initial value problem (**IVP**) for ordinary differential equations (**ODE**). To perform this task either for high order equations or equivalentely systems of first order equations, one has to implement algorithms for the numerical solution of the IVP (of course for both, "ordinary" IVP's and stiff ones). In this work, the following four integration algorithms have been examined:

- Euler's Method
- Runge–Kutta–Fehlberg method of order 4
- Adams–Bashforth–Moulton 4 step Predictor–Corrector method
- Rosenbrock's implicit Runge–Kutta method for stiff systems,

all of the above with adaptive stepsize control.

To parallelize those integration algorithms, we split the whole model in smaller model parts, so called *submodels*. Each submodel is evaluated by one processor. The more submodels you have, the more processors you can use. One has to define intervals for synchronizing Results in order to handle the feedbacks. Those synchronization intervalls have to be predefined, usually as synchronization after each integration step. To handle integration algorithms of order greater 1 and to handle adaptive step size control, it is necessary to perform interpolation (or extrapolation) for several values of the independent variable *t*. One of the goals of this work was to examine dependencies of the result quality versus setting appropriate synchronization intervals and setting the right interpolation method. The following two interpolation methods have been used:

- Polynomial Interpolation
- Rational Bulirsh-Stoer Interpolation

This work is dealing with the numerical aspects of parallel integration algrithms. The given results were generated by special Simulation software "PARINT", which is able to solve given IVP's by "simulating" the behaviour of a multiprocessor machine on a PC or a Workstation. A "real" implementation of parallel integration algorithms can be found in "MOSIS", a very powerful simulation tool, which was developed by G.Schuster at the Technical University of Vienna.

Aspects of great interest are the behaviour of the system by using the same integration algorithm for each submodel with and without adaptive stepsize control, the use of different interpolation algorithms, especially using different order of polynomial interpolation and using linear interpolation, the use of different integration algorithms for each submodel and the behaviour of stiff systems under parallel integration.

The Flexible-Production-Analyzer (FPA)

For analyzing rule based production systems

N.Kraus, R.Schachinger

In the FPA, a modern integrated simulation tool for analyzing rule based production systems has been created. With the help of automatic model generation and integration into the client server environment of a database, high power flexibility cán be realized. In the FPA the main simulator is encapsulated. Due to this, the FPA can be used as a tool for production control by staff without simulation or programming knowledge. The FPA is used with the help of a data front end and a data back end on client PCs; the tool can be tailored to the user's needs by adapting the clients for front and back end.

1. The Analysis Tool FPA

With the FPA complete production processes can be modeled. A model consists of several production modules, which represent different production technologies. Each of these production modules has several workcenters, which consist of machines performing similar process steps. The FPA requires input data, detailing resources, operations and orders. The data can be imported from a PPS-system and be modified with a data converter, in order to obtain data suitable for the simulation. These are:

- Resource data: Data of production units, workcenters and machines; data detailing failures, interruptions, etc.
- Operational Data: Process plans, process definitions, product information, work calendars etc.
- Order Data: Order lists, lot attributes etc.

2. The Simulation Analysis Module

For the investigation of simulation runs, some standard statistics, such as queue length in the buffer, number of products or machine utilization, are gathered and documented in tables. Furthermore, with the FPA it is possible to collect user defined statistics from any resource or process in the model. Standard statistics are:

- Machine statistics Machine or work center related data, number of lots produced, interrelationships between processing, setup, interruptions, pauses etc.
- Buffer statistics Queuing data, queue length for the most utilized buffers
- Order statistics Global order data for the whole production or local data for particular work centers
- Product statistics Data of processed products (globally or work center related)

The statistics can be set active or inactive globally or locally for particular work centers, as per request of the job to be investigated. Furthermore, it is possible to gather statistics during a specified period and to save the data in files.

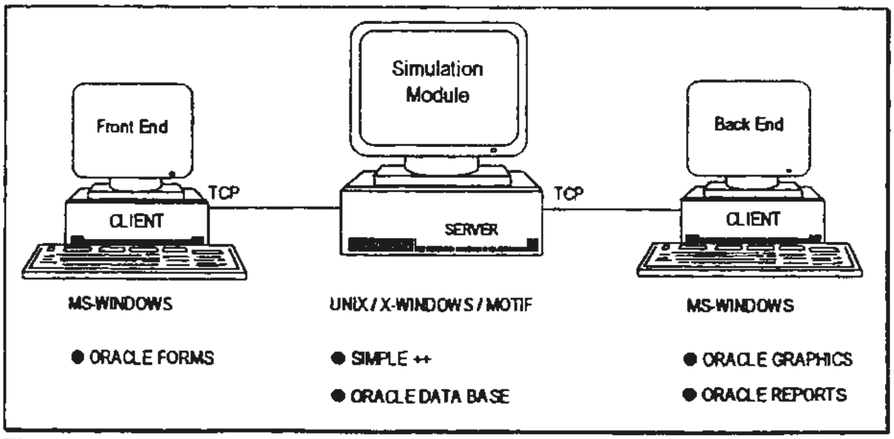


Fig. 1: Integration of the FPA in a client server environment.

MODELING OF SYSTEMS WITH EVENTS BY MEANS OF THE MODEL INTERCONNECTION CONCEPT OF THE SIMULATION SYSTEM mosis

[1] G. Schuster and [2] J.Plank
[1] Advanced Technical Software, ARGESIM
Flurschützstraße 16/10, A-1120 Wien
eMail: guenter@osiris.tuwien.ac.at
[2] Technical University Vienna, ARGESIM
Wiedner Hauptstraße 8-10, A-1040 Wien

1. General Description of the Model Interconnection Concept

The *Model Interconnection Concept (MIC)* was introduced with the simulation system mosis which has been a simulation systems project at the Technical University of Vienna since 1992 and is a general method to describe systems to be simulated as a connection of several independent models that communicate with each other. The advantages of this concept are modular development of simulation models (as every part can be implemented independently from all others), easy expandability for extensions like hardware-in-the-loop or connections to other simulation systems and parallelization of models, which was the main reason for the introduction of this concept: When a model is described as a connection of several smaller models, those parts can be simulated on different processors which is a very simple, but efficient method of simulation systems parallelization which gives the user full control about the parallelization strategy (in contrary to other strategies that do anything automatically and give the user absolutely no control).

The concept also comprises an object oriented point of view, as the model itself is only a description of the real system but does not contain any real data. For actual simulation, first an *instance* of a model has to be created which contains all data, but also the information about connections to other instances, as the instance connections are not done in the model definition, but during the experiment phase in the run time environment. Using this feature, several instances of the same model can be calculated simultaneously using different parameter settings or can even be used to create different model connections to create different comprehensive models.

2. The MIC for describing events

As stated above, the MIC was originally only designed to describe concurrent model connections. But in real world simulation tasks there is often the situation to describe discontinuities in dynamic systems. Those can be either *time events* (triggered at a certain value of the independent variable, which is usually the time) or *state events* (that are triggered when the system enters a certain state, e.g. when a the value of a state variable crosses the „zero"-line). When such an event is triggered, an appropriate action has to be taken which can be either the change of a system parameter (e.g. the reversion of the velocity vector at the model of a bouncing ball), but can also be the change of the model description itself (e.g. when a non-elastic ball hits the surface and breaks, or rolls down a ramp). In most simulation systems (including mosis), such situations have been described by using an „IF"-structure (or similar) in the model description. Even if this works, this is not a very elegant way which can also lead to calculation problems depending on the simulation system used (by integrating over this discontinuity).

For this application, the MIC has been extended to enable also serial connections of instances. When an event is triggered, the currently running model is replaced by another one, maintaining the same instance. To enable this change a *transition function* has to be defined which transfers the necessary variables defined in the previous model to the variables used by the following one.

When the two models were defined as *„model firstm()"* and *„model secondm()"* (input and output signal definitions within the parentheses), the connection function is defined *as „transition(firstm, secondm){ }"* where the braces enclose the variable transfer statements between the variables of *firstm* to those of *secondm*. The model sequence can be determined either statically (by writing *„transition (secondm);"* into the event handler code) or dynamically (only *„transition;"*) which makes it necessary to describe the sequence at run time level by using the *„sequence"*-command (*„sequence(firstm,secondm)"*) where even more than two models can be connected to form a sequence consisting of several steps.

Care should be taken that the number and meaning of input and output signals of both models are the same. The number is checked by the „mosis-to-C" translator (it allows only transition functions of models with the same signal count), but the meaning cannot be checked automatically.

[1] Schuster, G.: Definition and Implementation of a Model Interconnection Concept in Continuous Simulation, Dissertation, TU Vienna, 1994

SIMULATION OF STRESS AND STRAIN DISTRIBUTION IN A SNOW COVER BY FEM (FINITE ELEMENT METHOD)

S. Wieshofer[1], O. Kautzky[2], Dr. K Kleemayer[3]
[1]Ausstellungsstr. 51/13, A-1020 Vienna
[2]Oskar Jaschag. 96, A-1130 Vienna
[3]University of Agricultural Sciences Vienna
Peter Jordan Straße 82, A-1190 Vienna

In co-operation with the University of Agricultural Sciences Vienna a snow model was developed to calculate the forces and displacements in snow to obtain information on damages snow causes to the woods. The analysis is carried out using a finite element program called ABAQUS.

Two components of motion can be observed in a settled snowpack: creep (internal deformation) and glide (slip of the entire snowpack over the ground) which are influenced by different mechanism. On the one hand creeping depends on internal body weight (material properties), on the other hand gliding is created by external factors e.g. free water in the boundary layer (between snow and ground), lack of macroscopic roughness (smooth grass surface).

The model was applied with following assumptions:

- slope with a uniform inclination
- snow is isotropic and homogeneous
- top of the snow cover is stress free

For the application the snow slope is divided into quadrangular finite elements to which material behaviour is assigned to and the ground is defined as *RIGID SURFACE.

To explain the creeping procedure snow is described by material constitutive equations. There are two different possibilities (actually there are three, but only two of them are relevant in this case) to implement material properties into an ABAQUS model. Either we refer to already completed material subroutines - in this case *VISCOELASTIC based on the hereditary integral:

$$\sigma(t) = \int_0^t 2G(t-t')\dot{\gamma}dt' + I\int_0^t K(t-t')\dot{\varepsilon}dt'$$

whereby *G* is the shear modulus and *K* is the bulk modulus, which are functions of the reduced time *t* - or it is possible to add own constitutive equations into user subroutines - *CREEP, LAW=USER. Thereby a linear viscous equation is applied:

$$\Delta\varepsilon = \frac{2}{3\eta}\sigma\Delta t - \frac{3}{\eta}\left(\frac{m+1}{m-2}\right)p\Delta t$$

Gliding is described by verifying the parameter of friction and by geometric means of description whereby the ground roughness is assumed to be modelled by a simple sine wave.

This model can be used to calculate the stresses of the snowpack on obstacles e.g. avalanche barriers, trees as well as the maximal stresses in the snowpack caused by these obstacles.
By varying the different parameters of the model such as slope inclination, snow thickness, density the significance of the single parameters can be recognized and the impacts on the motion.

This is only a very basic and simplistic model which can be expanded by further research e.g. by including temperature.

OPTIMAL MANAGEMENT OF RAILWAY UNDER SPECIAL CONSIDERATION OF TECHNOLOGY AND PERSONELL MANAGEMENT

Dipl.-Ing. P. Kralicek
Unternehmensberatung Kralicek
Mantlergasse 17/1, A-1130 Wien
email: e8926312@fbma.tuwien.ac.at

For this type of problem a special computer simulation programme was developed, which helped to examine the impacts on the use of the railway section by a step by step upgrading of the railway section with new technology. This computer simulation programme serves as a planning and decision making tool for railway securing engineers and for representatives of firms providing the latest technology.

After checking the necessary requirements SIMPLE++ was chosen as the most suitable language for this computer simulation. SIMPLE++ is an object orientated language where the model is been put together from self made user parts. Therefore the model for this particular railway section can easily be adapted to many other railway sections. In addition SIMPLE++ is equipped with an extraordinary computer animation which proved to be important for a convincing presentation of the simulation results.

The examinations were implemented by using a single tracked local line in the Czech Republic, on which there are four stations.

The external installation consists of several different elements. Each of these elements can be found in many designs with various technologies:

- **Switch:** manual - by cable - electric - electric-hydraulic
- **Derailing point:** manual - by cable - electric - electric-hydraulic
- **Signal:** mechanic - by light

The elements of the external installation can be driven by the station's control point in three different ways:

- mechanically
- by relay
- electronically

For the different types of technology of the external installation and the control points SIMPLE++ user parts have been developed to support the creation of the model.

A very important part of the computer simulation model is the reproduction of the activities of the area managers. The work of the area managers includes the implementation of track-actions. Four different types of track-actions can be distinguished: thoroughfare, arrival, departure and shunting.

Each of these track-actions runs through five to six phases. These phases can vary in length of time depending on the technology of the external installation and the control points used.

announcement → control → control point →
administration → signal operation → welcoming the train

In addition to the technology of the external installation and the control points the computer simulation programme can change the train schedule as well as the track-layout.

The computer animation is able to show the track-layout in a very detailed way. During the computer simulation each switch position, each signal position and each position of the trains can be shown.

The results of the computer simulation are as follows:

- table of track capacity used
- activities of the area managers
- delays and detours

Alltogether three different computer simulation models were developed. One for the actual situation and two different variations (variant 1, variant 2), where the railway section was step by step graded up with new technology.

By using the new technology the whole running of the railway of this special examined section can be operated by one (variant 2) respectively two (variant 1) area managers. 75 per cent (variant 1) respectively 87,5 per cent (variant 2) of the number of area managers can be saved. The use of capacity of the defile track segments is declining by one (variant 1) respectively two (variant 2) per cent.

Uljanov Alexander

Bahngasse, A-2700 Wr. Neustadt

Optimum partitioning of data is very important for speed-up in SPMD (single program multiple data) parallel simulation. Formally the problem consists of following: there is a graph $G=(X,U)$ with functions of weights of edges: $\rho_u : U \rightarrow R^{+}$ and functions of weights of vertexes $\rho_x : X \rightarrow R^{+}$. We have to calculate a partitioning X' of set X with limitation $\sum_{x_i \in X^{*}, X^{*} \in X'} \rho_x(x_i) \leq V^{*}$ and a criterion $J=\sum_{U_r} \rho_u(x_i,x_j)$ must be minimum, where

$$U_r = \{(x_i,x_j)/(x_i,x_j) \in U \wedge (\exists X^{*} \in X')[(x_i \in X^{*} \wedge x_j \notin X^{*}) \vee (x_j \in X^{*} \wedge x_i \notin X^{*})]\}.$$

To solv this problem the Floyd-matrix of the shortest chains $R' = \|r'_{ij}\|$ is used, where $R' = \|r'_{ij}\|, i,j=1,2,\ldots,n, r'_{ij} = \rho_u(x_i,x_j)$ when $(x_i,x_j) \in U$ and $r'_{ij} = \infty$, when $(x_i,x_j) \notin U$.

Let us accept, then vertex subsets $x^{(1)}_{l_{n-1}}$ for the vertex x'. $d(x', x^{(1)}_{l_1}) \leq V^{*}$ and a shortest chain $C_{l_1} = x', x^{(1)}_{l_1}, x^{(1)}_{l_2}, \ldots, x^{(1)}_{l_{n}}$ exist. We have a net with the source $x^{(1)}_{lk_1-1} \in X^{(1)}_1$ and flow-off $x^{(1)}_{lk_1}$ (or in the opposition direction). The maximal flow in this net $U^{(1)}_1 = \min_{\sigma^1 \subset U} \sum_{(x_i,x_j) \in \sigma^1} \rho_u(x_i,x_j)$ can be calculated with the Ford-Fulkerson algorithm.

The cut σ^1 determines the partitioning of set X in two sets $X^{(1)}_1$ and $\overline{X}^{(1)}_1$, where $x^{(1)}_{lk_1} \in \overline{X}^{(1)}_1, X^{(1)}_1 \wedge \overline{X}^{(1)}_1 = \{\emptyset\}, X^{(1)}_1 \cup \overline{X}^{(1)}_1 = X$.

We will get a value $V^{(1)}_1 = \sum_{x_i \in X^{(1)}_1} \rho_x(x_i)$, then let us accept, the vertex $x^{(1)}_{l_{n-1}}$ is a source and the vertex $x^{(1)}_{lk_1-1}$ is a flow-off. Now the maximal flow is to get according to formula:

$$U^{(1)}_2 = \min_{\sigma^2 \subset U} \sum_{(x_i,x_j) \in \sigma^2} \rho_u(x_i,x_j), V^{(1)}_2 = \sum_{x_i \in X^{(1)}_2} \rho_x(x_i), x^{(1)}_{lk_1-2} \in \overline{X}^{(1)}_2, X^{(1)}_2 \wedge \overline{X}^{(1)}_2 = \{\emptyset\}, X^{(1)}_2 \wedge \overline{X}^{(1)}_2 = X, \ldots$$

and so on, until we get the maximal flow between x' and $x_{l_1}^{(1)}$. The k_1 minimal partitionings and k_1 variants, $X^{(1)}_1, X^{(1)}_2, \ldots, X^{(1)}_{k_1}, X^{(1)}_i \subset X, i=1,2,\ldots,k_1$ will be calculated using the chain C_{l_1}.

Further the following three components will be got for all the chains:

$$V = \{V^{(1)}_1, V^{(1)}_2, \ldots, V^{(1)}_{k_1}, V^{(2)}_1, \ldots, V^{(2)}_{k_2}, \ldots, V^{(n)}_1, \ldots, V^{(n)}_{k_n}\}$$

$$X' = \{X^{(1)}_1, X^{(1)}_2, \ldots, X^{(1)}_{k_1}, X^{(2)}_1, \ldots, X^{(2)}_{k_2}, \ldots, X^{(n)}_1, \ldots, X^{(n)}_{k_n}\} \quad \text{where}$$

$$U' = \{U^{(1)}_1, U^{(1)}_2, \ldots, U^{(1)}_{k_1}, U^{(2)}_1, \ldots, U^{(2)}_{k_2}, \ldots, U^{(n)}_1, \ldots, U^{(n)}_{k_n}\}.$$

$$|V| = |X'| = |U'| = \sum_{l=1}^{n} k_l$$

$$(\exists i,j,m,l)[V^{(l)}_{ij} = V^{(l)}_{lm} \mathrm{I} X^{(l)}_{ij} = X^{(l)}_{lm} \mathrm{I} U^{(l)}_{ij} = U^{(l)}_{lm}]$$

$$U^{(l)}_{ij} \geq \rho_u, X^{(l)}_{ij} \subset X.$$

Then the optimal partitioning $X^{(q)}_r \in X'$ is to calculate with function

$$J = \min_{i=1,n} \min_{j=1,k_i} \{l_1 * [V^{*} - \sum_{x_i \in X^{(i)}_j} \rho_x(x_i)] + l_2 * \frac{\sum_{(x_r,x_m) \in U^{(i)}_j} \rho_u(x_r,x_m) - P_u}{\sum_{\substack{(x_r,x_m) \in U \\ x_r,x_m \in X^{(i)}_j}} \rho_u(x_r,x_m) - P_u}\}.$$

where minimum is to calculate for all i,j. $C^{(i)}_j = \sum_{x_i \in X^{(i)}_j} \rho_x(x_i) \leq V^{*}$

The alternation of the values $l_1 l_2$ gives the variants with the minimum of parts (machines, blocks) or minimum value of the cut (interfaces). This method is effective when the value of $|V| = |X'| = |U'| = \sum_{l=1}^{n} k_l$ is enough great.

Index of Authors

Authors of Papers

Chaudhuri K. S., 7
Kedzior Z., 13
Kwater T., 13
Pekala R., 13
Scheibenbogen M., 1

Authors of Posters

Kautzky O., 24
Kleemayer K., 24
Kralicek P., 25
Kraus N., 22
Plank J., 23
Schachinger R., 21, 22
Schuster G., 23
Uljanov A., 26
Wieshofer S., 24