Proceedings

Late Paper Volume

Session 'Software Tools and Products

Poster Book

F. Breitenecker
I. Husinsky
editors

ARGESIM Report No. 1 / No. 2 / No. 3

ISBN ebook 978-3-901608-70-4

DOI 10.11128/arep.01-02-03

ISBN 3-901608-01-X

ISBN ebook 978-3-901608-70-4

DOI 10.11128/arep.01-02-03

ARGESIM Report No. 1 / **No. 2 / No. 3**

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

Printed by *CA Druckerei*, Vienna, Austria

Foreword

Developments over the last years show that beside the classical tools theory and experiment simulation becomes more and more the third major tool for problem solving in application and research. Nowadays simulation is found in nearly every application area, research activities result in new methodologies and tools for simulation, and more and more simulation software, simulators, and simulation systems are offered on the market.
The *EUROSIM Congress*, the European Simulation Congress, an international event normally held every three years, aims to be a common forum for presenting European and international recent results and applications in simulation, and to stimulate the exchange of ideas and experiences among scientists and engineers active in simulation.
EUROSIM is the Federation of the European Simulation Societies, acting as a European forum for Simulation Societies and promoting the advancement of system simulation in industry, research, and education.

All these intentions are reflected in the 5th European Simulation Congress *EUROSIM 95*, the 2nd Congress after the formal foundation of EUROSIM. The scientific programme consists of invited and contributed papers to regular sessions and to "Special Interest Sessions", of contributions to the session "Software Tools and Products", and of posters.

The invited and contributed papers to regular sessions and to "Special Interest Sessions" are published in the Congress Proceedings printed by Elsevier Science B.V. and in a Late Paper Volume (ARGESIM Report no. 1, ISBN 3-901608-01-X). The Proceedings contain eight invited papers and 212 contributed papers, the Late Paper Volume contains 20 contributed papers. The papers were selected by the International Programme Committee from 459 abstracts received.

The session "Software Tools and Products" presents papers dealing with State-of-the-Art and new features of simulation languages, simulators, and simulation environments. These contributions also passed the review process and are published as ARGESIM Report no. 2 (Proceedings EUROSIM'95 - Session "Software Tools and Products", ISBN 3-901608-01-X).

The reviewed Poster Session completes the scientific programme. The abstracts of the 111 posters are published as ARGESIM Report no. 3 (EUROSIM'95 - Poster Book, ISBN 3-901608-01-X).

It is interesting to compare the titles of papers and posters presented at previous European Simulation Congresses with those at the present congress. Even a brief glance through the four volumes of Proceedings and Late Paper Volumes shows that in this twelve year period considerable, remarkable, and sometimes astonishing advances have been made in a number of different areas. For example, developments in parallelism and distributed processing are now not only being seen in simulation applications but are also frequently used.
Object-oriented methods are being implemented now, and artificial intelligence and knowledge-based tools appear to be an established part of system modelling and simulation methodology. The availability of improved graphic algorithms and tools is also leading to some very interesting and innovative research and application in terms of man-machine interface and of animation and visualisation, both for discrete-event and continuous-system simulation.

New developments in terms of mathematical modelling and simulation techniques as well as in terms of general methodology are of little significance unless they are stimulated by the requirements of the real world in terms of industry, business, agriculture and the sciences. We are very pleased, therefore, that application papers are so well represented. This also applies to papers on parallel and distributed simulation, where beside graphics the fastest development can be observed.

We are also pleased that the idea of "Special Interest Sessions" could be realized. These sessions deal with recent developments in areas where methodology and application are considered together. The results of the closing discussion at the end of these sessions are summarized in manuscripts which will be edited and published in abbreviated form in *EUROSIM - Simulation News Europe* (SNE), the newsletter of the EUROSIM member societies. Some of these papers will be prepared for publication in EUROSIM's scientific journal *SIMULATION PRACTICE AND THEORY*. A separate role is played by the Industry Session on "Model Exchange and Software Independent Modeling" where people mainly from industry report on this topic without necessarily having to publish a paper in the Proceedings. Furthermore, we are pleased, that the contributions to the session "Software Tools and Products" show a very broad spectrum of simulation software, and that the Poster Session presents new ideas under development.

The European Simulation Congress *EUROSIM 95*, held in Vienna (Austria) at the Technical University of Vienna from September 11 through September 15, 1995, is organized by *ASIM* (Arbeitsgemeinschaft Simulation), the German speaking Simulation Society, in co-operation with the other member societies of EUROSIM: *AES* (Asociación Española de Simulación), *CSSS* (Czech & Slovak Simulation Society), *DBSS* (Dutch Benelux Simulation Society), *FRANCOSIM* (Société Francophone de Simulation), *HSTAG* (Hungarian Simulation Tools and Application Group), *ISCS* (Italian Society for Computer Simulation), *SIMS* (Simulation Society of Scandinavia), *SLOSIM* (Slovenian Society for Simulation and Modelling), *UKSS* (United Kingdom Simulation Society).

The moral co-sponsorship of *CASS* (Chinese Association for System Simulation), CROSSIM (Croatian Society for Simulation Modelling), *IFAC* Advisory Board Austria, *IMACS* (International Association for Mathematics and Computers in Simulation), *JSST* (Japanese Society for Simulation Technology), *LSS* (Latvian Simulation Society), *OCG* (Austrian Computer Society), *PSCS* (Polish Society for Computer Simulation), *ROMSIM* (Romanian Society for Modelling and Simulation), *SCS* (Society for Computer Simulation), *SiE* Esprit Working Group "Simulation in Europe" supports this congress.

A successful conference is always due to the efforts of the many people involved. To this purpose, particular acknowledgement goes to the members of the Scientific Committee for their contributions in the paper selection process, to the members of the Local Organizing Committee, and more especially to the head of this committee, to Mr. Manfred Salzmann. We would like to thank *Unseld + Partner* and *CA (Creditanstalt)* for sponsoring the printing of this report.

Felix Breitenecker Irmgard Husinsky

Technical University of Vienna

Proceedings

Late Paper Volume

F. Breitenecker
I. Husinsky
editors

ARGESIM Report No. 1

ISBN 3-901608-01-X

ISBN ebook 978-3-901608-01-8

DOI 10.11128/arep.1-2-3.ar1

ISBN 3-901608-01-X

ISBN ebook 978-3-901608-01-8

DOI 10.11128/arep.1-2-3.ar1

ARGESIM Report No. 1

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

Printed by *CA Druckerei*, Vienna, Austria

Foreword

Developments over the last years show that beside the classical tools theory and experiment simulation becomes more and more the third major tool for problem solving in application and research. Nowadays simulation is found in nearly every application area, research activities result in new methodologies and tools for simulation, and more and more simulation software, simulators, and simulation systems are offered on the market.
The *EUROSIM Congress*, the European Simulation Congress, an international event normally held every three years, aims to be a common forum for presenting European and international recent results and applications in simulation, and to stimulate the exchange of ideas and experiences among scientists and engineers active in simulation.
EUROSIM is the Federation of the European Simulation Societies, acting as a European forum for Simulation Societies and promoting the advancement of system simulation in industry, research, and education.

All these intentions are reflected in the 5th European Simulation Congress *EUROSIM 95*, the 2nd Congress after the formal foundation of EUROSIM. The scientific programme consists of invited and contributed papers to regular sessions and to "Special Interest Sessions", of contributions to the session "Software Tools and Products", and of posters.

The invited and contributed papers to regular sessions and to "Special Interest Sessions" are published in the Congress Proceedings printed by Elsevier Science B.V. and in a Late Paper Volume (ARGESIM Report no. 1, ISBN 3-901608-01-X). The Proceedings contain eight invited papers and 212 contributed papers, the Late Paper Volume contains 20 contributed papers. The papers were selected by the International Programme Committee from 459 abstracts received.

The session "Software Tools and Products" presents papers dealing with State-of-the-Art and new features of simulation languages, simulators, and simulation environments. These contributions also passed the review process and are published as ARGESIM Report no. 2 (Proceedings EUROSIM'95 - Session "Software Tools and Products", ISBN 3-901608-01-X).

The reviewed Poster Session completes the scientific programme. The abstracts of the 111 posters are published as ARGESIM Report no. 3 (EUROSIM'95 - Poster Book, ISBN 3-901608-01-X).

It is interesting to compare the titles of papers and posters presented at previous European Simulation Congresses with those at the present congress. Even a brief glance through the four volumes of Proceedings and Late Paper Volumes shows that in this twelve year period considerable, remarkable, and sometimes astonishing advances have been made in a number of different areas. For example, developments in parallelism and distributed processing are now not only being seen in simulation applications but are also frequently used.
Object-oriented methods are being implemented now, and artificial intelligence and knowledge-based tools appear to be an established part of system modelling and simulation methodology. The availability of improved graphic algorithms and tools is also leading to some very interesting and innovative research and application in terms of man-machine interface and of animation and visualisation, both for discrete-event and continuous-system simulation.

New developments in terms of mathematical modelling and simulation techniques as well as in terms of general methodology are of little significance unless they are stimulated by the requirements of the real world in terms of industry, business, agriculture and the sciences. We are very pleased, therefore, that application papers are so well represented. This also applies to papers on parallel and distributed simulation, where beside graphics the fastest development can be observed.

We are also pleased that the idea of "Special Interest Sessions" could be realized. These sessions deal with recent developments in areas where methodology and application are considered together. The results of the closing discussion at the end of these sessions are summarized in manuscripts which will be edited and published in abbreviated form in *EUROSIM - Simulation News Europe* (SNE), the newsletter of the EUROSIM member societies. Some of these papers will be prepared for publication in EUROSIM's scientific journal *SIMULATION PRACTICE AND THEORY*. A separate role is played by the Industry Session on "Model Exchange and Software Independent Modeling" where people mainly from industry report on this topic without necessarily having to publish a paper in the Proceedings. Furthermore, we are pleased, that the contributions to the session "Software Tools and Products" show a very broad spectrum of simulation software, and that the Poster Session presents new ideas under development.

The European Simulation Congress *EUROSIM 95*, held in Vienna (Austria) at the Technical University of Vienna from September 11 through September 15, 1995, is organized by *ASIM* (Arbeitsgemeinschaft Simulation), the German speaking Simulation Society, in co-operation with the other member societies of EUROSIM: *AES* (Asociación Española de Simulación), *CSSS* (Czech & Slovak Simulation Society), *DBSS* (Dutch Benelux Simulation Society), *FRANCOSIM* (Société Francophone de Simulation), *HSTAG* (Hungarian Simulation Tools and Application Group), *ISCS* (Italian Society for Computer Simulation), *SIMS* (Simulation Society of Scandinavia), *SLOSIM* (Slovenian Society for Simulation and Modelling), *UKSS* (United Kingdom Simulation Society).

The moral co-sponsorship of *CASS* (Chinese Association for System Simulation), CROSSIM (Croatian Society for Simulation Modelling), *IFAC* Advisory Board Austria, *IMACS* (International Association for Mathematics and Computers in Simulation), *JSST* (Japanese Society for Simulation Technology), *LSS* (Latvian Simulation Society), *OCG* (Austrian Computer Society), *PSCS* (Polish Society for Computer Simulation), *ROMSIM* (Romanian Society for Modelling and Simulation), *SCS* (Society for Computer Simulation), *SiE* Esprit Working Group "Simulation in Europe" supports this congress.

A successful conference is always due to the efforts of the many people involved. To this purpose, particular acknowledgement goes to the members of the Scientific Committee for their contributions in the paper selection process, to the members of the Local Organizing Committee, and more especially to the head of this committee, to Mr. Manfred Salzmann. We would like to thank *Unseld + Partner* and *CA (Creditanstalt)* for sponsoring the printing of this report.

Felix Breitenecker Irmgard Husinsky

Technical University of Vienna

Scientific Committee

F. Breitenecker (Austria), Chairman

H. Adelsberger (D)
M. Alexik (SQ)
W. Ameling (D)
S. Balsamo (I)
I. Bausch-Gall (D)
S.W. Brok (NL)
F.E. Cellier (USA)
V. Ceric (HR)
L. Dekker (NL)
J. Dongarra (USA)
V. De Nitto (I)
H. Ecker (A)
G. Feichtinger (A)
P. Fishwick (USA)
J.M. Giron-Sierra (E)
H.J. Halin (CH)
N. Houbak (DK)
R. Huntsinger (USA)
I. Husinsky (A)
T. Iversen (N)
A. Jávor (H)
K. Juslin (FIN)
G. Kampe (D)
E. Kerckhoffs (NL)
W. Kleinert (A)
P. Kopacek (A)
W. Kreutzer (NZ)
M. Lebrun (F)
F. Lorenz (B)
F. Maceri (I)
D. Maclay (GB)
H. Mang (A)
Y. Merkuryev (LV)
Z. Minglian (VRC)
I. Molnar (H)
D.P.F. Möller (D)
D. Murray-Smith (GB)
F.J. Pasveer (NL)
R. Pooley (U.K.)
W. Purgathofer (A)
P. Schäfer (D)
T. Schriber (USA)
A. Seila (USA)
W. Smit (NL)
F. Stanciulescu (RO)
A. Sydow (D)
H. Szczerbicka (D)
S. Takaba (J)
I. Troch (A)
G.C. Vansteenkiste (B)
W. Weisz (A)
J. Wilkinson (GB)
R. Zobel (GB)
B. Zupancic (SLO)

Organization Committee

Felix Breitenecker, Dept. Simulation Techniques,
Institute for Technical Mathematics, Technical University of Vienna

Irmgard Husinsky, Dept. for High Performance Computing, Computing Services,
Technical University of Vienna

Local Organizers are the Dept. Simulation Techniques of the Technical University of Vienna and the Dept. for High Performance Computing of the Computing Services of the Technical University of Vienna, and the "ARGE Simulation News".

Local Organizing Committee

M. Salzmann;

K. Breitenecker, B. Gabler, M. Holzinger, C. Kiss, M. Klug, N. Kraus, M. Lingl,
I. Mannsberger, J. Schuch, G. Schuster, H. Strauß, S. Wassertheurer, W. Zeller

About ARGESIM

ARGE Simulation News (ARGESIM) is a non-profit working group providing the infra structure for the administration of **EUROSIM** activities and other activities in the area of modelling and simulation.

ARGESIM organizes and provides the infra structure for

- the production of the journal EUROSIM Simulation News Europe
- the comparison of simulation software (EUROSIM Comparisons)
- the organisation of seminars and courses on modelling and simulation
- COMETT Courses on Simulation
- "Seminare über Modellbildung und Simulation"
- development of simulation software, for instance: mosis - continuous parallel simulation, D_SIM - discrete simulation with Petri Nets, GOMA - optimization in ACSL
- running a WWW - server on EUROSIM activities and on activities of member societies of EUROSIM
- running a FTP-Server with software demos, for instance
  - * demos of continuous simulation software
  - * demos of discrete simulation software
  - * demos of engineering software tools
  - * full versions of tools developed within ARGESIM

At present ARGESIM consists mainly of staff members of the Dept. Simulation Technique and of the Computing Services of the Technical University Vienna.

In 1995 ARGESIM became also a publisher and started the series **ARGESIM Reports**. These reports will publish short monographs on new developments in modelling and simulation, course material for COMETT courses and other simulation courses, Proceedings for simulation conferences, summaries of the EUROSIM comparisons, etc.

Up to now the following reports have been published:

No.	Title	Authors / Editors	ISBN
# 1	Congress EUROSIM'95 - Late Paper Volume	F. Breitenecker, I. Husinsky	3-901608-01-X
# 2	Congress EUROSIM'95 - Session Software Products and Tools	F. Breitenecker, I. Husinsky	3-901608-01-X
# 3	EUROSIM'95 - Poster Book	F. Breitenecker, I. Husinsky	3-901608-01-X
# 4	Seminar Modellbildung und Simulation - Simulation in der Didaktik	F. Breitenecker, I. Husinsky, M. Salzmann	3-901608-04-4
# 5	Seminar Modellbildung und Simulation - COMETT - Course "Fuzzy Logic"	D. Murray-Smith, D.P.F. Möller, F. Breitenecker	3-901608-04-4
# 6	Seminar Modellbildung und Simulation -COMETT - Course "Object-Oriented Discrete Simulation"	N. Kraus, F. Breitenecker	3-901608-04-4
# 7	EUROSIM Comparison 1 - Solutions and Results	F. Breitenecker, I. Husinsky	3-901608-07-9
# 8	EUROSIM Comparison 2 - Solutions and Results	F. Breitenecker, I. Husinsky	3-901608-07-9

For information contact: ARGESIM, c/o Dept. Simulation Techniques,
attn. F. Breitenecker, Technical University Vienna
Wiedner Hauptstraße 8-10, A - 1040 Vienna
Tel. +43-1-58801-5374, -5386, -5484, Fax: +43-1-5874211
Email: argesim@simserv.tuwien.ac.at

Table of Contents

Neural Nets in Modelling and Simulation

Simulation and Artificial Intelligence

Simulation for the Future: Progress of the ESPRIT Basic Research Working Group 8467 (acronym: SiE-WG)

Eugene J.H. Kerckhoffs
Delft University of Technology
Faculty of Technical Mathematics & Informatics
Julianalaan 132
2628 BL Delft, the Netherlands

The ESPRIT Basic Research Working Group 8467 on "Simulation for the Future: New Concepts, Tools and Applications", with the acronym SiE-WG (Simulation in Europe - Working Group), started its work officially on December 1, 1993; the duration of the working period is three years. It was an initiative of SiE-SIG (Simulation in Europe - Special Interest Group), established in January 1992, to submit a proposal to the European Commission for financial support of such a simulation-oriented Basic Research Working Group. The list of participants of SiE-WG consists of the contractor (the University of Ghent, Belgium) and 23 associated (academic and industrial partners) from all over Europe. The Special Interest Group SiE- SIG has over 120 academic and industrial contributors and interested persons Europe-wide. The Working Group SiE-WG works in close collaboration with the Special Interest Group SiE-SIG. In fact, SiE-SIG is a platform and forum for SiE-WG, where activities of SiE-WG are initiated and results and follow-ups are discussed; SiE-SIG is the ruling body with respect to the eventual industrial applicability. Chairmen of SiE-WG are prof. G.C. Vansteenkiste (University of Ghent) and prof. E.J.H. Kerckhoffs (Delft University of Technology, the Netherlands).

The SiE-WG's programme of work in the three-years period is subdivided in two phases of 1.5 year each. Phase 1 is a transition phase and a phase of meta-research, which should result in a detailed specification of phase 2, during which SiE-WG shall work as a normal Working Grouop as meant in the Esprit Basic Research Workprogramme. The Working Group has organized in its first 1.5 year period two major workshops (Brussels, June 16-17, 1994 and Brussels, June 29 - July 1, 1995). In the first Workshop, the focus was on "Improvement of the Modelling and Simulation Process" (with the tracks "Generic, Multiparadigm Modelling", Multilanguage Modelling and Simulation", and "Modelling and Simulation Life Cycle"); the second Workshop was organized around the themes "Adaptive Interfaces", "Distributed Interactive Simulation (DIS) and Synthetic Environments" and "High-Performance Simulation".

In the lecture, SiE-WG's working procedure and the results of the afore-mentioned workshops will be discussed.

Meanwhile, the Working Group decided to focus in the second 1.5 year period on the following (sub)topics and actions:

1. Modelling:
   -- Neutral model definition: formats & standards
   -- Multiparadigm modelling
2. Simulation (model implementation and experimentation):
   -- Neutral separator (between model definition and execution): formats & standards
   -- New languages: aspects, standards
   -- Communication between distributed models; DIS
3. Human-simulator Interfaces:
   -- Adaptive interfaces (with AI, Hypermedia, etc.).

For each of the above topics 1-3 the following actions are foreseen:

a. Classify state-of-the-art
b. Study selected current applications; formulate generic aspects
c. Develop methodologies, methods, techniques
d. Select (industrial) benchmark applications
e. Build demonstrators.

For all of these the following should (implicitly) be taken into account:

-- M&S life cycle aspects
-- impact of possible implementation on parallel and distributed hardware (high-performance simulation)
-- orientation to M&S for the future, especially in the development of "Synthetic Environments" (SEs) and "Virtual Design and Manufacturing Environments" (VDMEs).

The above actions and topics determine a 5 * 3 (action - topic) matrix (the actions a - e forming the rows, and the topics 1 - 3 the columns of the matrix). Activities falling in one or more "elements" of this matrix can be proposed to SiE-WG for possible sponsoring. Only communication overhead costs (such as expenses for exchange of scientific people between research institutions, meetings, etc.) can be covered.

CHOOSING SIMULATION SOFTWARE FOR FLEXIBLE JOB SHOP ENVIRONMENT

Janusz Niwinski
Danube Hydro Austria
Parkring 12, A-1010 Vienna, Austria

ABSTRACT

The realization of a JIT in flexible manufacturing is usually connected with the ability to effectively translate the workings of the decision making processes into the job shop. The success in carrying out such tasks depends in many cases on availability of proper (job shop and user conform) tools. Properly employed simulation via its forecasting abilities can serve as a base for developing such cooperation. In this paper some of the features of the simulation systems are seen through the scope of „on-line" job shop experiments.

INTRODUCTION

A simulation system should not simply be investigated as a concoction of various routines and functions. It must be also scrutinized and seen as an existing developer's philosophy, which has an important impact on its usability and future extendibility. Such information could be found in the analysis of software development and ripening process, the vendor attitude toward the users, universities and operating cases. First, after gaining a given portion of background knowledge about the overall system environment, a case or application related examination should be conducted.

CONSTRUCTION OF THE SIMULATION SYSTEMS (SYNTAX)

The first stage of system scrutiny ought to be an examination of its construction and syntax. Already this stage can decide if the further checks of the system are necessary and can further save a lot of work done for detailed investigating of non-applicable systems. Within these analyses at least the following elements have to be scanned in detail:

- the number and the type of offered functions,
- conditions of the treatment of the discreet and continuous systems,
- number and names of the system variables,
- linking of external functions,
- form and functions of the debugger,

The simulation system should support simulation of all the activities and situations that arise in the manufacturing environment. The multitude of aspects under which flexible manufacturing causes, together with the decision making processes, the number of the involved cases to be

analyzed as the subsystems rapidly grow. This causes requesting of availability of case dependent functions, which can not be included in standard software. Offering such special functions by the vendor would be connected with additional costs of the base software and endanger the system readability, this would often result in the canceling of the venture. To avoid this, a simulation system should support definition of user specific functions.

Although the logic of the material flow in the manufacturing environment can be usually described by discrete elements only, it could be advantageous to employ elements of continuous systems. The availability of this option is especially important it the simulation model should be used for automatic monitoring of the facility performance, where continuous processes can be used as triggers for predefined activities.

In many simulation systems variables are referred to through numbers, not names. It is in many cases advantageous for the software vendor, because he/she can use it for direct variable referencing, but it is very difficult to handle in model building and application. This feature is often accompanied by the non-extendibility of the variable sets. Both restrictions lead to the necessity of employing various tricks and external defined structures, this can cause additional errors and growing modeling costs. To avoid this, a good simulation software should use mnemonic variable names and allow user to define his own variable set and give them his own names.

One of the more important features of a good simulation system is its openness. It allows the user to write and employ in his simulation model problem specific functions. These can be coded in one of the standard programming languages (C, FORTRAN, ...) and integrated with the system. User defined routines should be callable not only before and after but also during simulation experiments. They should be able to be triggered by both external and internal events and states and treat the external situations as model internal ones. It would not only allow the real model-user interaction, but also support the integration of the model with the real elements of examined facility.

In a flexible job shop changes in the model can occur very often. Usually there is no time for long tests of implemented modifications. In such an environment the debugger can not be a kind of external module requiring a lot of additional procedures and preparations. It must be an on-line element of the simulation system (developed model) allowing its activation any time, without lowering the overall experiment performance.

INPUT

In flexible manufacturing simulation models should be used direct in the office of the job shop supervisor. It means that it will be used by a person which not necessarily is a computer professional or 'freak'. The quality of the implemented input procedures will decide if the tool will be accepted and if its potential can be utilized. The completely different from the standard application of the simulation user profile causes that the input functions have to be examined very scrupulously.

In addition to the „hard-coded" parameter values, there must be a possibility for interactive changes of the model and experiment parameters before the simulation run. It has to be

accomplished without any changes in the model (experiment) source code, which would required additional compiling and linking. The initialization of the experiment parameter must be possible using one of the following methods:

- interactive input of the model parameter,
- initializing of the experiment with the parameter file,
- acquiring data from the external systems.

These options are not as obvious as they should be. In many older but still widely sold and used simulation systems they are still not fully implemented. Some of them support just the second option, which was enough in the experimentation of facility planning.

Simulation parameters are seldom available in the form of direct numerical values and have to be computed using various mathematical functions. The searched simulation system has to include typical distribution functions and the appropriate solutions for random number generation ability being available (callable) before and during experiments. These functions should support both the manual (user) as well as the automatic (model or external system) processes of input generation.

To support the on-line controlling and monitoring of the flexible job, the employed simulation system must offer an on-line interface with the external system. It should guarantee, that the external events (coming from user, MPR, data collection systems, data based, etc.) can be received and processed during experiments in „real time". This feature must embrace not only signals for start and stop of the experiment, but also for the input and change of model internal parameters.

OUTPUT

The usefulness of the simulation model is based upon the quality of the decisions and prediction about simulated processes. This causes that the quantity and quality of the analysis together with evaluation and presentation features decide about the employment of the simulation tool.

A selected simulation system should support the computation of standard statistics and present them jointly with the system and model parameter in form of diagrams, plots, histograms, etc. Computed (collected) data should be stored in user defined structures (files, databases). It must be possible to store the graphical depictures of the analyses and rework them using other software tools. In addition to the standard output forms, user must be able to define his own case dependent output forms and formats.

The quick development taking place in the software area brings about a wide range of various tools for data handling, analysis and presentation. This availability, together with the expansion in network technology allows for the employment of distributed specialized work places. In order to utilize such a system the simulation model must support on-line and off-line data transfer with such external data handling tools. This demand becomes an additional high priority when the simulation system has to cooperate with the MRP/MRP II and data collection system, conducting real time analyses of manufacturing processes.

ANIMATION

In the past, animation was mostly seen just as a nice but not very useful feature of the simulation packages. This meaning led to certain underestimation of the value of this tool and a low profile of offered solutions. In standard applications of facility planning without time critical processes, simulation studies could afford long and detailed studies based only on numerical data. In flexible JIT manufacturing decisions have to be made very quickly. There is seldom time for long extensive analysis of numerical results. In such cases animation becomes an almost ideal tool for fast estimation of the expected results. In the following some important elements of animation are shortly discussed.

The quality of the animation screens should reflect state of the art of computer graphic. The quality of the animation shows the value this tool has for the software developers. It lets us predict the developments and status it will have in the future extensions and updates of the system. Depending on the used hardware, the following characteristics of the animation should be investigated:

- art of employed graphic (raster, vector),
- resolution of the picture,
- number of the colors,
- employing transparent and solid objects,
- multi frame picture preparation,
- offering multi layer animation,
- offering multi window animation.

Because the animation picture includes a set of elements that are not normally used in a standard graphic application (dynamic diagrams, AGV-systems, ...) the searched simulation package should include an integrated graphic editor. This editor must offer a possibility of importing (exporting) animation elements from other highly specialized external graphical packages. It allows for the use of already existing graphical elements (CAD), reduces the training costs and should cause, on the vendor side, the reduction of development costs and the price of the simulation system.

The complexity of analyzed processes of flexible manufacturing makes, in many cases, the animation picture too small for observing all of the selected elements. This problem can be solved by employing zoom or/and multi window animation. Both of the solutions have their advantages and disadvantages and it is a heavy task to weight them properly. Zoom-option based mainly on vector graphics is much faster but has to cope with problems of filled areas and requests positioning of all the elements on one picture. Multi-window options can use simpler (raster) graphics allowing elegant presentation of the modeled elements in freely defined multiple windows

In flexible manufacturing dynamic interaction with the model is one of the most important elements. It allows for a proper control of the experiments (changing speed of the animation, changing scale of the picture, switching between various windows of the data presentation, stopping and initiating experiments, etc.). In addition to monitoring functions it must support input (change) of selected model parameters. For this purpose a simulation system should support employing switches

in the animation picture which will trigger, during experiments, user defined functions and processes.

A proper analysis of the system can seldom be done based just on summary statistics collected during simulation experiments. In flexible manufacturing, there is hardly enough time for such tasks. Many decisions can be met already during simulation experiment as a result of the understanding of changes of selected system parameters. For this reason a searched simulation system should support presentation of not just temporary status of chosen elements, but also their time dependent diagrams (Gantt diagrams, plots, etc.).

PERFORMANCE

The best analyses of a dynamic manufacturing system are worthless, if they are too late. One of the important characteristics of the simulation tool is a good performance. The performance features are very difficult to judge for their case dependence. The only way is it to let the vendor solve a typical problem of the future employment and measure the counts. In general, independent from the hard- and software the simulation system is running on, one of the main factors of the performance scores is the language and art that the simulation system is implemented. The first simulation languages have been implemented in FORTRAN and many of them use till now FORTRAN libraries, FORTRAN compatible structures and FORTRAN methodology. Unfortunately, the performance of such systems represents not always the latest state of computer science. For this reason, a searched system should be C-based, the performance should go together with the openness and extendibility of the simulation models

INTERFACES

The „must feature" of the simulation in the flexible job shop is the communication with the external systems like MRP/MRP II and data collection systems. It should also be able to exchange data with other software products like spreadsheets, data analysis and presentation tools, word processing, data bases, etc. This communication has to be realized not only off-line before and after simulation experiment, but also on-line during simulation runs. The hardware and the software platform of the simulation system must support such interfaces and the system has to make use of them.

SOFTWARE AND HARDWARE PLATFORM

The availability of the system (number and type of the hardware and software system where it is already ported) can be a one of the meaningful characteristic of the product; Its dissemination, portability and the customer acceptance. It allows models developed on one hardware (software) platform to be used on an other one, and for new developments and updates expected in the future. The portability of the models reduces the initial investments allowing for testing models on relatively simple and inexpensive machines, transferring later already examined solutions on the real

work stations. The implementing of the software on the GUI (Graphical User Interface) is at the time the must condition for searched simulation system.

PRICE AND SERVICE

Prices for simulation packages lie between $500 US and $ 150 000 US. The price alone says nothing about the usability of the product. Price should be examined with the other elements of the simulation system and its environment, which can cause future time and money expenditures:

- ease of use and ease of learning,
- quality and language of the documentation,
- on-line help and tutorials,
- cooperation with the universities,
- customer support,
  - training ,
  - site of technical support (hot-line),
  - updates and enhancements,
  - user groups and newsletters,

Properly weighting of these elements can allow estimating purchase price and future costs of the software. An important element are the user groups. Any leaflet, a promise or prospect can have the value of the direct contact with other users.

CLOSING REMARKS

Selecting a proper simulation software is a very complex and heavy task. The named above characteristics comprise just a few general elements important in on-line simulation on the flexible job shop floor. The conditions named in this paper should show the direction where the examination should be concentrated and can be identified with the words: - '**User-friendly**', '**Fast**' and '**Open**'. If after tests (not demos and presentation) an appropriate ('**UFO**') system can be traced, it should be taken and used extensively. The advantages of appropriate simulation methods in the supporting of decision making in the flexible job shop will soon pay back the investment and secure the competitiveness of the company.

Results of the EUROSIM Comparison "Lithium Cluster Dynamics" - Trends in Continuous Simulation Software

F. Breitenecker[a] and I. Husinsky[b]

[a] Dept. Simulation Techniques, fbreiten@email:tuwien.ac.at
[b] Computing Services, husinsky@edvz.tuwien.ac.at
Technical University Vienna, Wiedner Hauptstraße 8-10, A - 1040 Vienna

This contribution summarizes the solutions of the EUROSIM Comparison on Simulation Software "Lithium Cluster Dynamics". The EUROSIM Software Comparisons (up to now eight) and the solutions sent in are published in the journal **EUROSIM Simulation News Europe (SNE)**. Based on the results some developments and trends in continuous simulation software and related problems are briefly sketched.

1. THE EUROSIM SOFTWARE COMPARISONS

EUROSIM, the Federation of European Simulation Societies, started in 1990 the publication of the journal **EUROSIM Simulation News Europe (SNE)**, a newsletter distributed to all members of the European simulation societies under **EUROSIM's** umbrella and to people and institutions interested in simulation. **SNE** is also part of **Simulation Practice and Theory (SIMPRA)**, the scientific journal of **EUROSIM**.
The idea of the journal **SNE** (circulation 2500; edited by F. Breitenecker and I. Husinsky, **ARGE Simulation News (ARGESIM)** , Technical University of Vienna, Austria; three issues per year) is to disseminate information related to all aspects of modeling and simulation. The contents of **SNE** are news in simulation, simulation society information, industry news, calendar of events, essays on new developments, conference announcements, simulation in the European Community, introduction of simulation centers and the comparison of simulation software, hardware, and simulation tools.
The series on comparisons of simulation software has been very successful. Based on simple, easily comprehensible models special features of modeling and experimentation within simulation languages are being compared (in case of continuous simulation):

- modeling technique
- event handling
- submodel features
- numerical integration
- steady-state calculation
- frequency domain
- plot features
- parameter sweep
- optimization

Eight comparisons (4 continuous, 3 discrete, and one special comparison for parallel simulation software and hardware) have been set up. The continuous comparisons are: "Lithium-Cluster Dynamics" (Comparison 1, 11/1990) - stiff systems, "Generalized Class-E Amplifier" (Comparison 3, 7/1991) - electronic circuits and eigenvalue analysis; "Two State Model" (Comparison 5, 3/1992) - high accuracy computation; and "Constrained Pendulum" (Comparison 7, 3/1993) - concentrating on state events (for more details and a summarizing table see the paper on the parallel comparison in these proceedings). The series will be continued.

2. THE EUROSIM COMPARISON C1 "LITHIUM CLUSTER DYNAMICS"

EUROSIM comparison 1 (Lithium-Cluster Dynamics under Electron Bombardment) has been performed by 26 simulation languages or simulators. This comparison is based on a stiff third order system of ODE's describing the concentrations $f(t)$, $m(t)$, and $r(t)$ of molecule agglomerates (F- , M - and R- centers) of alkali halides under electron bombardment:

$$dr/dt = -d_r r + k_r mf$$
$$dm/dt = d_r r - d_m m + k_f f^2 - k_r mf$$
$$df/dt = d_r r + 2\,d_m m - k_r mf - 2\,k_f f^2 - l_f f + p$$
$$k_r = d_m = 1,\ k_f = d_r = 0.1,\ l_f = 1000$$
$$r(0) = 9.975,\ m(0) = 1.674,\ r(0) = 84.99$$

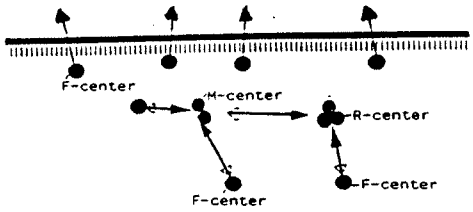


Fig. 1: Comparison1, physical background

Three tasks or had to be performed: **i)** test and comparison of integration algorithms ($t \in [0, 100]$), **ii)** parameter sweep of l_f (100, ... 10000) with log plots, and **iii)** steady state calculation for $p = 0$, $p = 10000$.

LANGUAGE	MODEL DESCRIPTION	REMARKS
ACSL	equations (ODEs)	CSSL-language with rich structure; 2 solutions
DESIRE	equations (ODEs)	combination with neural network simulation; interfaces to C and Turbo Pascal
DYNAST	equations (DAEs) (*) graphical blocks (sub models) port diagrams (graphical)	semi-symbolic analysis for linear systems; documentation environment based on AutoCAD or TeX for PC version
ESACAP	equations (DAEs) (*), nodes / branches, arbitrary expressions	"European Space Agency Circuit Analysis Program"; based on numerical algorithm for circuit analysis
ESL	equations (ODEs) (*) graphical blocks (sub models)	interpretative and compile mode; graphic postprocessor
EXTEND	graphical blocks	continuous and next event modeling; mainly Macintosh.
FSIMUL	graphical blocks (sub models)	"Control Engineering" - optimisation features
HYBSYS	blocks (elementary) (*) equations	"Hybrid Simulation System" (1980 TU - Wien) interpretative simulator; direct data base compilation;
IDAS / SIMPLORER	graphical(ORCAD, ...) equations (Description Language) by dialog (Windows) (*)	specialized for electronic circuits and control problems; based on Windows
I Think	graphical blocks	modeling based on system dynamics; no slot to other modeling or programming languages
MATLAB	equations (MATLAB functions)	tool for mathematical and engineering calculations
MATRIXx	graphical blocks (*) matrix manipulation	interactive matrix-manipulation; using LINPACK and EISPACK
mosis	equations	"modular simulation system"; CSSL-type on C basis; features for parallelization on MIMD-systems;
NAP2	blocks (electronic circuits)	specialized for circuit simulation
POWERSIM	graphical blocks description	based on System Dynamics formulation
PROSIGN	equations (ODEs) graphical blocks (sub models) application-oriented components	"Process Design"; combination of modeling techniques; interfaces to C, Fortran, Modula2; variable number of input and output parameters
SABER	equations (ODEs)	specialized for analogue circuit simulation
SIL	equations (ODEs, DAEs)	simulation of discrete and continuous systems; free format
SIMNON	equations (ODEs) (*) macro function, sub models	simulation of discrete and continuous systems; real-time features; connecting systems; direct data base compilation
SIMULINK	graphical blocks (sub models) (*) equations (MATLAB functions)	based on MATLAB; special integration-algorithm Linsim; no limits for number of states and variables; 2 solutions
SIMUL_R	equations (ODEs) (*) bond graphs (graphical preprocessor) blocks (graphical preprocessor)	simulation of discrete and continuous systems; open system, based on C; runtime interpreter; combined simulation
STEM	equations (ODEs)	"Sim. Tool for Easy Modeling"; basis on Turbo Pascal
TUTSIM	graphical blocks, bond graphs equations (ODEs) (*)	"Twente University of Technology" (NL); simulation of discrete and continuous systems
XANALOG	graphical blocks (sub models)	sophisticated linearization, real-time features

Table 1: General features of simulation languages

First it has to be noted that all simulation languages fulfilled the tasks with sufficient accuracy. Table 1 gives an overview about simulation languages and simulators, where solutions were sent in (column 1). The simulators can be divided roughly into three groups: equation oriented languages, (graphical) block-oriented languages, application-oriented languages. The table indicates these different modeling techniques (column 2). As some languages offer different modeling approaches, the one used in the solution sent in is marked with an asterisk. Special features and essential properties are remarked in column 3.

3. RESULTS AND EVALUATION OF THE COMPARISON

Eigenvalue analysis of the linearized model results in three eigenvalues being negative real numbers. At t = 0 the eigenvalues are -0.00898, -11.06, -1005.66, at t = 10 the values are -0.0978, -1.018, -1003.4 . Dividing the absolute value of the biggest eigenvalue by the absolute value of the smallest eigenvalue results in a stiffness factor. At t = 0 this factor is approximately 120000, *at* t = 10 the factor is about 10000.
Figure 2 shows this stiffness changing over the time (logarithmic scales): fast transients happen at the very beginning of the simulation, afterwards the system is relatively smooth.

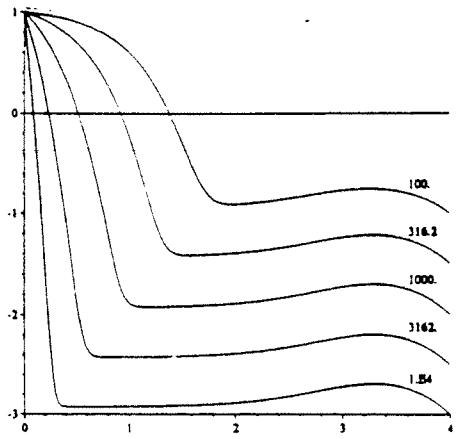


Fig.2: Results f(t), variation of l_f

3.1 Task i): Test and comparison of integration algorithms

It is relatively difficult to compare the results of this task. Although most languages offer exact CPU-times for the different algorithms, these results suffer from side effects like I/O-time, straight-forward or tricky modeling, well tuned algorithm parameters (model-dependent!) or standard values, etc. Therefore, for the comparison of the algorithms the relation between the different algorithms is more significant than absolute CPU-times (normalized to Euler algorithm).
Table 2, summarizing these results, is mostly restricted to three algorithms: Gear stiff algorithm (variable stepsize, variable order), Euler algorithm (fixed stepsize) and Runge- Kutta algorithm (RK4, mainly fixed stepsize), because these algorithms all work "well" (in case one or more of these algorithms are missing, preferably results of Runge-Kutta-Fehlberg and Adams-Moulton algorithm are given).
Table 2 generally shows that the Gear algorithm is the best one for this model because of the stiffness of the system. Unfortunately some reports do not indicate which order the Gear algorithm had to choose in order to fulfill the constraints on the relative or absolute errors, resp.
Insight into these questions offers for instance ESACAP, which compares different BDF-algorithms (Backward Differential Formulas, the predecessors of the Gear algorithms) on the basis of number of steps, function evaluations, calculations of the Jacobian matrix, etc. Furthermore, the most efficient Gear algorithms or BDFs are offered by languages (DYNAST, ESACAP, SIL) using model description on basis of DAEs (Differential Algebraic Equations) - by reformulating the model in implicit form.
The classical RK4 algorithm works well, if an appropriate stepsize and an appropriate relative error is chosen, being approximately 10 times slower than the Gear algorithm. RKF algorithms (Runge-Kutta-Fehlberg) speed up the integration time using stepsize control.
It is known from theory that the Adams-Moulton and /or Adams-Bashforth-algorithms are not suitable for this kind of systems; but it is astonishing that they are really very slow.
Another astonishing phenomenon is the result of the Linsim algorithm of SIMULINK, which is twice faster than the classical Gear algorithm. This algorithm extracts the linear parts of the models and calculates the linear dynamics via power series, the nonlinear parts are integrated in the usual manner.
Three solutions sent in showed that it is worth thinking over a model before simulating it. The authors made use of the fact that fast transients happen only at the very beginning. Consequently, the second ACSL solution choose exponentially spread sampling points, resulting also in related stepsize (also better

suited for log plots). The DESIRE solution and the first SIMNON solution performed this exponential time shift directly in the model equations (logarithmic time transformation). As a consequence, the integration algorithms became (much) faster, the system became nearly non-stiff.

3.1 Task ii): Parameter sweep and log plots

The second task should test whether a simulation language offers features for parameter sweeps. Table 3 summarises the results in column 2, where it is tried to distinguish between parameter loops in the model description and at run-time level. In case of graphical model description model frame and experimental frame are mixed, so that this distinction becomes difficult.
Furthermore, it turned out that the additional requirement of a logarithmic parameter sweep and logarithmic plot was no further challenge: if parameter loops are available, different increments can be used; if the parameter sweep has to be formulated in a "manual" way, the logarithmic sweep is also simple. The third column in table 3 therefore indicates only, whether logarithmic representations are supported directly ("standard") or not ("manual" transformation).

LANGUAGE	SNE-NR C1-NR	COMPUTER	ALGORITHM	STEPSIZE ACCURACY	COMPUTATION TIME OTHERS
ACSL	SNE-1 C1-3	PC 80287/12	Adams-Moulton	5.E-3 iss	1 (155.055 sec)
			Gear	5.E-3 iss	0.022
			RKF 4/5, vs	5.E-3 iss	0.355
ACSL	SNE-5 C1-17	Micro VAX/ Sun4	Euler	1.E-5 / 2.E-1 ss	1 (8.43 sec) / 0.056
			RK 4	1.E-5 / 2.E-1 ss	1.981 / 0.101
			Gear	1.E-8 ae, 1.E-5 iss	0.236 / 0.018
DESIRE	SNE-4 C1-	PC 80387/16	Gear	1.E-5 ae, 1.E-6 logiss	10 sec
		Sun 4c	Gear	1.E-5 ae, 1.E-6 logiss	1.7 sec
DYNAST	SNE-3 C1-12	PC 80387	Gear-Newton-Raphson	1.E-3 re, 1.E-5 iss	0.506
				1.E-6 ae, 1.E-5 iss	1 (4.45 sec)
ESACAP	SNE-1 C1-1	PC 80387	BDF 1o, vs	1.E-3 re/ 1.E-7 re	118ns,237f/10271ns,20547f
			BDF 2o, vs	1.E-3 re/ 1.E-7 re	53 ns,105f/ 316 ns, 632f
			BDF 3o, vs	1.E-3 re/ 1.E-7 re	51 ns,102f/185 ns,370 f
ESL	SNE-2 C1-8	PC 80387 SX/16	RK 4	1.E-3 ss	0.571
			Adams Bashforth	1.E-1 iss	1 (21 sec)
			Gear	1.E-1 iss	0.01
EXTEND	SNE-5 C1-15	Macintosh IIfx	Euler impr.	12000 ns / 10000 ns	1 (1 sec) / unstable
			Trapezoidal rule	30000 ns/ 20000 ns	2.3 / unstable
FSIMUL	SNE-1 C1-4	PC 80387 /25	AB 2o, vs	5.E-4 iss/ 1.E-3 iss	0.556 / unstable
			implicit Heun	5.E-4 ss/ 1.E-3 ss	0.973 / unstable
			RK4	5.E-4 iss/ 1.E-3 iss	1 (187 sec) / unstable
HYBSYS	SNE-2 C1-7	DECStation 3100/16	ABM	1.E-5 iss	1.983
			Euler	1.E-4 ss	1 (8.47 sec)
			RK 4	2.E-4 iss	1.099
IDAS	SNE- C1-25	Pentium 60mHz	Euler	minss=0.002 mss=0.01	1 (8 sec)
			Trapezoidal		1
I Think	SNE-5 C1-16	Macintosh IIfx	Euler	1.E-4 ss/ 1.E-3 ss	1 (420 sec) / unstable
			RK 2	1.E-4 ss/ 1.E-3 ss	1.286 / unstable
			RK 4	1.E-4 ss/ 1.E-3 ss	1.714 / unstable
MATLAB	SNE-3 C1-10	PC 80387 (PS/S80)	RKF 2/3	1.E-5 re	739 sec
			RKF 4/5	1.E-6 / 1.E-7 re	563 sec / 752 sec
MATRIXx	SNE-10 C1-19	PC 80486/33	Euler	1.E4 equ. time points	1 (90.3 sec)
			RK2 / RK4	1.E4 equ. time points	1.468 / 2.411
		Sun 4 /40	Euler	1.E4 equ. time points	1 (8.19 sec)
			RK2 / RK4	1.E4 equ. time points	1.442 / 2.322

LANGUAGE	SNE-NR C1-NR	COMPUTER	ALGORITHM	STEPSIZE ACCURACY	COMPUTATION TIME OTHERS
mosis	SNE-12 C1-22	PC 486/33	Euler	1.0E-3 ss	1 (2.3 sec)
			RK4	1.0E-3ss /1.0E-4 ss	1.783 / 17.957
			Adams Moulton	1.0E-4 ss,1.0E-8 mae	1.122
			Stiff Alg.	1.0E-4 ss,1.0E-8 mae	0.039
NAP2	SNE-1 C1-2	PC 80387 Norton CI 25,6	mod. Gear, vs,vo	1.E-5 iss	4.56 sec
POWERSIM	SNE 14 C1-25	PC 80486/66	Euler	1.0E-3 ss	1 (32 s)
			RK4	vs, 1.0E-3 iss	1.2
PROSIGN	SNE-3 C1-13	not given	Simpson 2o, vs	1.E-3 mss	1 (470 sec)
			AB 4o, vs	2.5.E-3 mss	0.434
SABER	SNE-11 C1-20	Sun	Gear 1o/Gear 2o	vs	1 (0.75 sec)/ 0.44
		SPARC10/402	Gear 2o/Gear 2o	5.E-4 ss/1.E-3 ss	1 (47.3 sec)/ 0.448
			Trapezoidal rule	vs	0.016
SIL	SNE-2 C1-9	PC 80387	Stiff alg., vs, vo	1.E-2 re/ 1.E-4 re	0.231 / 0.351
				1.E-6 re/ 1.E-10 re	0.49/ 1 (11.43 sec)
SIMNON	SNE-12 C1-23	PC 80386/25	Euler	1.0E-3	1 (23 sec)
			RKF23	vs, 1.E-6 re	0.913
			RKF45	vs, 1.E-6 re	0.652
SIMNON	SNE-11 C1-21	PC 80386/40	Euler	1.0E-3	1 (31 sec)
			RKF23	vs, 1.E-6 re	0.39
			RKF45	vs, 1.E-6 re	0.264
		PC 80486/66	Euler	1.0E-3	1 (9.8 sec)
			RKF23	vs, 1.E-6 re	0.398
			RKF45	vs, 1.E-6 re	0.276
SIMULINK	SNE-3 C1-11	Sun 4	RK 5, vs	1.E-2 re,1.E0-.E-4 ss	1 (10.4 sec)
			Gear	1.E-2 re,1.E0-.E-4 ss	0.034
			Linsim	.E-2 re,1.E0-1E-4 ss	0.018
SIMUL_R	SNE-1 C1-5	not given	Euler	1.E-3 ss, 1.E-5 re	1 (not given)
			RK 4	2.E-3 ss, 1.E-5 re	1.9
			Euler implicit	1.E-1 ss, 1.E-3 re	0.22
STEM	SNE-5 C1-18	PC 80287/20	RKF 1/2o, vs	1.E-6 re, 1.E-3 ae	1 (18.84 sec)
			RKF 4/5o, vs	1.E-6 re, 1.E-3 ae	0.574
			Gear, vs	1.E-6 re, 1.E-3 ae	0.027
TUTSIM	SNE- C1-24	PC 80387/16	Euler	5.E-4 mss	1 (44 sec)
			AB	5.E-4 mss	1.114
XANALOG	SNE-2 C1-6	PC 80287 /16	RK 4	1.E-3 ss / 2.5.E-3 ss	2.744 / 88 sec
			Euler	1.E-3 ss / 2..E-3 ss	1 (82 sec) / unstable
			mod. Euler	1.E-3 ss / 2..E-3 ss	1.439 / unstable

Legend: ss ... stepsize; iss ... initial ss; log (i)ss ... logarithmic (i)ss; mss ... max. ss; re ... relative error; ae ... absolute error; ns ... number of steps; f ... function evaluations, vs ... variable ss; vo ... variable order; 4o ... 4th order; etc.; RK4 ... classical Runge-Kutta; RKF ... Runge-Kutta-Fehlberg; AB(M) ... Adams-Bashforth(-Moulton); BDF ... Backwards Differential Formulas

Table 2: Results of task i): test and comparison of integration algorithms

3.3 Task iii): Steady state calculation:

The third task should check which languages offer features for steady state calculation. The model is simple enough to calculate the steady states analytically, so all results could be compared with the exact values ($p = 10000$; $f_s = 10$, $m_s = 10$, $r_s = 1000$; $p = 0$; $f_s = m_s = r_s = 0$).

Languages with steady state finder (column 3 of table 3 ,"trim command, iteration") calculated the results for both cases with sufficient accuracy. Usually the iterative solution of the steady state equations started with the initial values for f, m and r.

Languages without a steady state finder ("longterm simulation") simulated over a long period stopping when derivatives are nearly zero (approx. at $t = 100$), getting as accurate results as the steady state finders.

LANGUAGE	PARAMETER VARIATION	LOG.	STEADY STATE CALC.
ACSL	manual variation at runtime	standard	trim command, iteration
DESIRE	parameter loop in model description	manual	not given
DYNAST	manual variation in model description	standard	long term simulation
ESACAP	parameter loop in model description	standard	long term simulation
ESL	parameter loop in model description	standard	trim command, iteration
EXTEND	manual variation in graphic model description	standard	long term simulation
FSIMUL	parameter loop in graphic model description	standard	long term simulation
HYBSYS	parameter loop at runtime	standard	trim command, iteration
IDAS	manual variation in model description	standard	long term simulation
I Think	manual variation in graphic model description	standard	long term simulation
MATLAB	parameter loop in model description	standard	trim command, iteration
MATRIXx	manual variation in model description	standard	trim command, iteration
mosis	parameter loop at runtime	standard	trim command, iteration
NAP 2	manual variation in model description	standard	long term simulation
POWERSIM	parameter loop in model desrc.(co-models)	manual	not given
PROSIGN	parameter loop in graphic model description	standard	trim command, iteration
SABER	parameter loop in model description	standard	trim command, iteration
SIL	parameter loop at runtime	manual	trim command, iteration
SIMNON	parameter loop at runtime	manual	long term simulation
SIMULINK	manual variation in graphic model description	standard	trim command, iteration
SIMUL_R	parameter loop at runtime	standard	trim command, iteration
STEM	manual variation in model description	manual	trim command, iteration
TUTSIM	parameter loop at runtime	standard	long term simulation
XANALOG	parameter loop in graphic model description	standard	trim command, iteration

Table 3: Results of tasks ii) and iii): Parameter sweep and steady state calculation

4. TRENDS AND DEVELOPMENTS

The results of this comparison also allows a view on developments and trends of simulation languages and simulators. In the following some trends are listed, but also the problems which may arise:

Developments:

- Implicit model descriptions
- Submodel features
- Graphical model descriptions
- Graphical preprocessors
- Sophisticated integration algorithms
- State event handling
- New analysis methods (formula manipul.)
- Separation of model and experiment
- More powerful runtime interpreters
- Windows Implementations

Problems:

- Loss of input-output relations
- Conflicts with macro features
- Loss of segment structure
- Overhead in generated equations
- Overhead for about 80% of problems
- Dependent on modeling technique
- CSSL structure too weak
- Interpreters not powerful enough
- Documentation with model
- Loss of speed, esp. on PC

In general, it is interesting, that

- Big enterprises tend to develop their own language, which are marketed, too
- Universities and institutions develop also new languages, which partially are successfully marketed
- In continuous simulation on the one side CSSL standard - languages become a common denominator for modeling, on the other hand a block-oriented graphical description based on control technique is frequently used.

Results and Experiences derived from a Comparison between Simulation Systems

J. Klußmann[1], J. Krauth[1] and R. Splanemann[2]
Bremen Institute of Industrial Technology and Applied Work Science at the University of Bremen (BIBA)[1], Degussa AG[2]

0. Abstract

This paper investigates if different simulation tools produce the same results when applied to the same system. Two comparisons have been carried out using the same test model of an assembly system. First a number of researchers carried out simulation experiments independent from each other, only on the basis of a written model definition. The results varied considerably, partly due to unclear model definition. In a second step a smaller number of tools has been compared by the authors themselves, thus excluding misunderstandings of the model definition. In this comparison the tools produced identical results once the models had been made really identical. But it was difficult to produce exactly identical models using different tools. Each tool has its inherent assumptions on "normal" behaviour, and if these assumptions are not known to the modeller, he is likely to generate a model with slight errors.

1. Introduction

Nowadays a wide variety of simulation tools is available on the market, especially in the field of material flow simulation in manufacturing systems. They all claim to be precise, but they all claim to be different. Hence the question arises: If used to simulate the same system, will they produce the same results? This question actually implies two questions: First, is it possible to create identical models using different tools? Especially with the modern comfortable tools which require no more programming but offer ready-made building blocks, this seems to be a problem. Building blocks are certainly very comfortable for quick and easy modelling, but they limit flexibility. So it is not clear if different simulation tools allow to generate identical models at all. The second question then is: In case the models can be made identical, do the tools then produce the same simulation results? Apparently these questions are very critical for the credibility of simulation.

The answer to both questions is "Yes, but..." In principle we can trust simulation results, but we have to be careful. This result is not surprising, but we feel it is often ignored in practical applications.

The paper is structured as follows: The next section describes the test system which is a simplification of a real assembly system. Section 3 reports on the result of a "distributed" comparison carried out by a number of researchers who each had only the written definition of the test system. The last section 4 represents own experiences of modelling the test system with three, different simulation tools and draws some conclusions about it.

2. The test system

We published the following test system definition in the journal "Eurosim Simulation News" in 1992 [1] and asked all interested persons or institutes to send us their solutions.

The test system consists of 7 assembly stations and a load/unload station all linked by an automated flexible conveyor system. This system is sketched in figure 1. An inner rectangular conveyor circulates clockwise and transports pallets on which the products to be assembled are fixed. The inner conveyor connects 8 subsystems as shown in figure 2. Each subsystem comprises one of the eight stations (assembly or load/unload) Ax, a buffer conveyor B2 of variable lenght in front of the station, a one-place buffer behind it, a bypass conveyor B1, and two connecting elements Sx and Sy. Here B1 is part of the inner rectangle. A pallet coming from the left can either be shifted to B2 in Sx or move along on B1. It is shifted to B2 if the following two conditions are satisfied:

- the product on the pallet has not yet undergone the operation(s) carried out in station Ax,
- there is enough space on the conveyor B2 in front of station Ax.

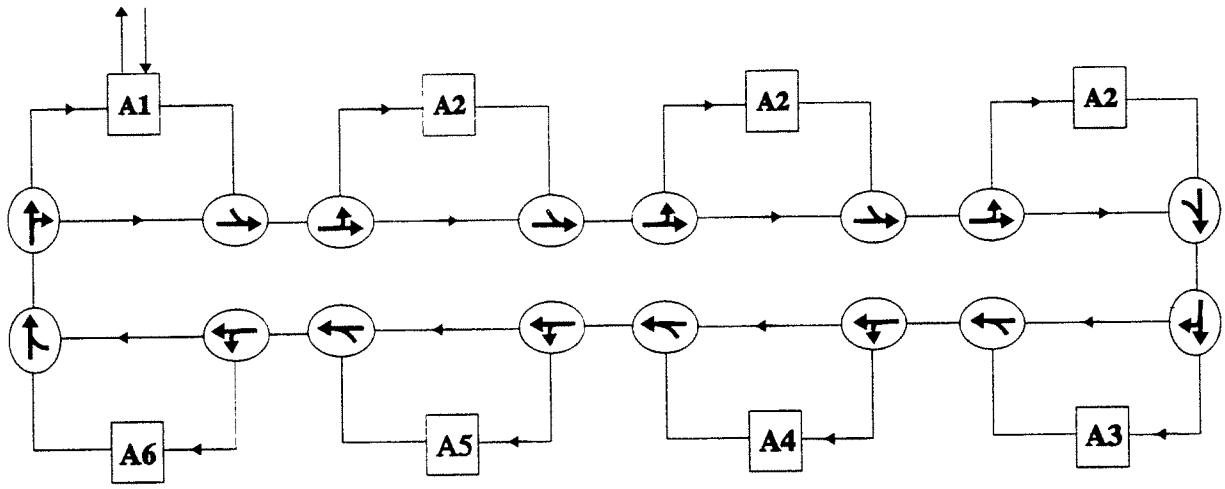


Figure 1: Flexible assembly line as test system

Finished products are taken from pallets and replaced by unprocessed parts in station A1. The sequence of operations the products undergo is arbitrary with the only exception that A2 has to be the first or the last station. All three stations A2 perform the same operations, hence only one of them has to process each product. Station A6 functions as a substitute of stations A3, A4, and A5. It performs all of the missing operations of these three whenever a product is being processed in A6.

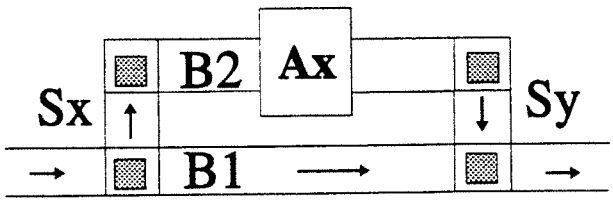


Figure 2: Subsystem of the flexible line

The processing times and the buffer sizes in front of all stations as well as the lenght of the respective bypass conveyors are given in table 1.

Station	Operation time (seconds)	Length of buffer in front of station (m)	Length of bypass conveyor (m)
A1	15	2.0	1.2
A2	60	1.6	0.8
A3	20	1.6	0.8
A4	20	1.6	0.8
A5	20	1.6	0.8
A6	30	2.0	1.2

Table 1: System parameters

After publishing this model definition we received a letter pointing out some ambiguous details of the definition. So we published a second, more precise definition in the following issue of Eurosim Simulation News [2] which we do not repeat here.

3. First Comparison

In the first comparison many scientists were involved, and their communication was very limited. The result has already been documented in Eurosim Simulation News [3], so we only repeat a summary here:

Table 2 (on the next page) gives the results obtained with 19 different simulation tools for a simulation time of eight hours. All tools have been applied by different researchers. The table shows a surprising diversity of results.

From this table it is impossible to tell to what degree the diversity of results is due to different understanding of the model definition or to errors in either the model implementation or the simulation software itself. From some researchers we received a note afterwards that they had misunderstood the model definition.

One ambiguous point is the question wether the time needed to feed a pallet into a station is part of the processing time or not. In case it is the bottleneck stations are A2, A3, A4, and A5, and the maximum number of products that can be processed in eight hours is 1440 because in this case every 20 sec. a product can be finished. In case it is not the bottleneck are stations A2 with a processing time of 61.3 sec., hence every 20.4 sec. a product can be finished, and therefore the maximum number of products is 1411. Most of the results obtained are close to one of these numbers.

Simulation system	Distributor	Author of test model	Number of assembled parts (with 20 pallets)
POSES	University of Chemnitz (D)	Ges. f. Prozeßautom. & Consult., Chemnitz (D)	1462
TAYLOR	F&H, Düsseldorf (D)	F&H, Düsseldorf (D)	1441
EXTEND	Imagine That, San Jose (USA)	University of Rostock	1440
SLAM II	Schröder GmbH, Düsseldorf (D)	AIC, Turin (I)	1440
SIMPLE-mac	AESOP, Stuttgart (D)	Unseld&Partner, Vienna (A)	1439
WITNESS	AT&T Istel, Düsseldorf (D)	BIBA, Bremen (D)	1439
DSIM	University of Vienna (A)	University of Vienna (A)	1425
CASSANDRA	KFKI, Budapest (H)	KFKI, Budapest (H)	1415
MICRO SAINT	Rapid Data, Worthing (GB)	Micro Analysis and Design, Boulder (USA)	1411
GPSS/H	Dr. Staedler GmbH, Nürnberg (D)	University of Michigan (USA)	1409
SIMFLEX/2	University of Kassel (D)	University of Kassel (D)	1409
DESMO	University of Hamburg (D)	University of Hamburg (D)	1408
DOSIMIS-3	SDZ, Dortmund (D)	IML, Dortmund (D)	1408
SIMUL_R	Simutech, Vienna (A)	Simutech, Vienna (A)	1405
EXAM	Russian Academy of Science, Moscow	Russian Academy of Science, Moscow	1404
PC SIMDIS	University of Magdeburg (D)	University of Magdeburg (D)	1384
MOSYS	IPK, Berlin (D)	IPK, Berlin (D)	1346
SIMAN	Dornier-System GmbH, Friedrichshafen (D)	CIMulation Centre, Chippenham (GB)	919
TOMAS	DVZ, Neubrandenburg (D)	DVZ, Neubrandenburg (D)	884

Table 2: Simulation results with different simulation systems

What was particularly remarkable about this first comparison was the little feeedback we received after having published the model definition the first time: Only one researcher asked for clarification of several ambiguous points. These are pointed out and clarified in [2]. All other researchers seemed to understand immediately what we meant. But the diversity of results proved that their understandings deviated from ours, and also from each other. When

we had published the clarification, nobody else asked for any more information, even though there were still many points unclear.

What does this mean? It is obviously very difficult to define a model in an unambiguous way. And it seems to be equally difficult to even notice where there are ambiguities. When one person defines something very precisely, and another person understands him perfectly well, it does not necessarily mean both have the same understanding. And it may take a very long time until they notice they have not. We believed our first - and even more our second - definition was clear enough to build a model, and the majority of researchers thought so, too. But what we defined and what they understood was not always the same. At least in some cases we definitely know differences in understanding the model definition.

With respect to simulation this implies there is always a risk of misunderstanding when a simulationist and an engineer cooperate and communicate about a model. This risk can of course be avoided when the engineer builds the model himself. But in order to enable him to do so, the tool must provide him with constructs he understands. Nowadays a considerable number of simulation tools provides such domain-specific building blocks as - in the case of manufacturing simulation - machines, buffers, conveyour, etc. Their dynamics are predefinded, so the user does not have to define them any more, he simply selects and combines them. But as we shall see below, this creates a new source of misunderstanding.

4. Experiences derived from a second Comparison

In the next step our aim was to exclude all sources of misunderstanding. Therefore we built the models on our own, using three different simulation systems: Dosimis-3 [4], Simple++ [5], and Witness [6]. These tools are frequently used in German manufacturing industry. They are particularly suited to model manufacturing and assembly systems, they support graphical modelling, and they provide the user with pre-defined domain-specific building blocks.

First simulation runs showed small differences between the results of the different simulation systems. A very detailed validation process proved that the three models were not identical. With each of the tools we had made some mistakes in modelling, mainly based on misunderstandings of the functionality and the behaviour of the pre-defined building blocks or modules the simulation tools provide. The detailed problems in modelling with the three simulation systems is published in [7]. In the following we represent the results and conclusions of this comparison.

The same results have been achieved with all three simulation systems. As well with all systems small mistakes first showed a little impact on the result. The mistakes happened mostly by modelling the distributing and connecting elements Sx and Sy. The reasons for all the mentioned mistakes are misunderstandings of the detailed behaviour of simulation system building blocks.

We assume the same results could also have been obtained using any of the other tools involved in the first comparison - or at least with the majority of them. It may be easy with some of them, and more tricky with others. But this does not mean some are good and some are bad. It only means they have primarily been designed for different purposes, by designers who had different perceptions of what a "normal" manufacturing system does.

The problem is that the user is quite often not aware of these differences in details. He himself has his own understanding of "normal" behaviour, and he tends to assume these comfortable modern simulation tools provide him with precisely the building blocks he expects.

Why should he think a "conveyor " block e.g. does not behave the way the conveyors he knows behave in reality? Unfortunately this assumption is often wrong. And - even worse - the exact description of the dynamics is often not available in the manual.

One solution of this problem is of course to make users aware of the potential diversity of builiding block behaviour, and to document precisely the behaviour of all building blocks in the manual. Another solution might be to provide the user with techniques to define his own building blocks. These techniques however have to be very simple, otherwise we would be back at simulation languages or even programming languages. Moreover the verification of user-defined blocks must be supported because he is likely to make mistakes, he will probably not test and validate them with sufficient rigour, and he is likely to use his own blocks again and again. And finally, we expect that user-defined building blocks will be documented even less, and therefore they will only be useful for the author himself, and after some months maybe not even for him. Hence simulation tools which allow the user to define his own building blocks have to provide solutions for these three subsequent problems of comfort and simplicity, of correctness, and of documentation.

A prototype of such an advanced simulation tool, allowing for user-defined building blocks and providing some techniques for rigorous verification, has been described in [8]. Petri nets have been used to define or modify application oriented building blocks. The mathematical theory of Petri nets allows for some rigorous testing of user-defined blocks, thus supporting their verification and validation to some extent. To our knowledge not much has been done since then to investigate further possibilities of validation support. However, more recent work towards tools which enable the user to define his own building blocks can be found in [9] and [10].

References:

[1] J. Krauth: Comparison 2: Flexible Assembly System. Eurosim Simulation News 1, March 1991, p.28.

[2] J. Krauth: Comparison 2: Flexible Assembly System. Eurosim Simulation News 2, July 1991, pp.25-26.

[3] J. Krauth: Comparison 2: Preliminary Evaluation. Eurosim Simulation News 4, March 1992, p.31.

[4] DOSIMIS-3, Benutzerhandbuch, SDZ GmbH, Dortmund (D).

[5] SIMPLE++, Referenzhandbuch, AESOP GmbH, Stuttgart (D).

[6] WITNESS User Manual, AT&T ISTEL Visual Interactive Systems Ltd., Highfield (UK).

[7] J. Krauth, R. Meyer: Comparison of simulation systems based on a test model. In: Systems Analysis, Modelling and Simulation, 17 (1995), pp. 1-12

[8] J. Krauth: Modellierung und Simulation flexibler Montagesysteme mit Petri-Netzen. OR-Spektrum (1990)12, pp 239-248

[9] J. Pirron: Simple++: The revolution in object-oriented simulation. Proceedings SIM94, Waseda University Tokyo, July 1994

[10] Schürholz, A: CREATE! Entwicklungsstand und Funktionsumfang einer objektorientierten Simulatorentwicklungsumgebung. In: Tavangarian, D. (Hrsg.): Fortschritte in der Simulationstechnik, Band 4, Simulationstechnik, 7. Symposium, Hagen, 1991, pp. 10-14

Systems Simulation with Nonlinear Supervisor

D. Popescu[1], A. Cheruy[2] and F. Doboga[1,2]

[1]Faculty of Control and Computer Science, U.P.B., Splaiul Independentei 313, Bucuresti 6, Romania
[2]Laboratoire d'Automatique du Grenoble, I.N.P.G., BP 46, 38402 St.Martin d'Heres,France

Author to be contacted: D.Popescu - mail: popescu@indinf.pub.ro

These pages suggest a control architecture with integrated supervisor for homogeneous or hybrid hierarchical structures. This approach is more and more frequently used to increase the efficiency in industrial plant exploitation. The supervisor is outlook like an intelligent system which takes the best decision even for a process intensely disturbed or with significant uncertainty. A control structure with two hierarchical levels is proposed. The inferior level is represented by a dynamic process connected at a multivariable controller. The upper level, the supervisor, assures an efficient exploitation regime for the process.

1.INTRODUCTION

The development in control area has been fueled by three major needs: the need to deal with increasingly complex systems, the need to accomplish increasingly demanding design requirements and the need to attain these requirements with less precise knowledge of the plant and its environment. Increasingly complex dynamic systems with significant uncertainty have forced system designers to turn away from conventional control methods.

The capacity of control structures with integrated supervisor to take decisions for complex processes intensely disturbed with a large degree of uncertainty suggests that these structures may be good candidates for realize the real time adaptive control of large-scale dynamic systems.

The supervisor is reprezented by a nonlinear multivariable model

$$\hat{z} = f(\hat{\Theta}, y) \tag{1}$$

This model is used for the construction of quality condition I(y)

$$I(y) = T[\hat{z}(\hat{\Theta}, y)] \tag{2}$$

where T is an algebraic transformation. The optimization problem associated is

$$max_{y \in D_{adm}}\{I(y)\} \tag{3}$$

with D_{adm} is the admisible domain of process[2].

The supervisor is a control system used for designing of optimal decisions. The decision is transfered in a reference computed forme ($r^*=y^*$).

2.DESIGNING OF OPTIMAL SUPERVISOR

2.1.Designing of Nonlinear Supervisor

Generally, for industrial applications, the supervisor could be considerated by a nonlinear model:

$$\hat{z}(t) = \hat{\Theta}^T f(y) = \hat{\theta}_1 f_1(y) + ... + \hat{\theta}_n f_n(y) \tag{4}$$

with $f_i(y)$ for i=1:n nonlinear functions in y. $\hat{\Theta}$ could be estimated by least-squares method

$$\hat{\Theta} = (f^T f)^{-1} f^T z \tag{5}$$

For $\hat{z}(\hat{\Theta},y)$ estimated the optimization problem is

$$max_{y \in D_{adm}} \{I(y) = T[\hat{z}(\hat{\Theta}, y)]\} \tag{6}$$

Generally, this problem is solved by algorithms which use a recurrent relation

$$y_{k+1} = y_(k) + \alpha_k d_k \quad y_0 \in R^n \tag{7}$$

The quality of every optimization method is related with the evaluation method of d_k. To facilitate the supervisor designing, a simple strategy of computing d_k is proposed.

In a Hilbert space, the distance between two points, y and y_k, is defined by

$$d^2(y, y_k) = (y - y_k)^T A (y - y_k) \tag{8}$$

where A is the *metric matrix*. The problem is to find y on the surface (8) which maximizes the condition I(y). To solve this problem I(y) is approximated by a first order Taylor series for the point y_k

$$I(y) \cong I(y_k) + (\Delta y)^T GI(y_k) \tag{9}$$

where $\Delta y = y - y_k$ and $GI(y_k)$is the gradient of optimization condition I(y) evaluated for y_k.

The optimization problem given by

$$max_y \{I(y) \cong I(y_k) + (\Delta y)^T GI(y_k) \mid \Delta y^T A \Delta y = d^2\} \tag{10}$$

could be solved for the stationarity conditions imposed by the Lagrange's function

$$L(y, \lambda) = \Delta y^T GI(y_k) - \lambda(\Delta y^T A \Delta y - d^2) \tag{11}$$

The derivatives of $L(y,\lambda)$ with respect to the y and λ are zero in the extreme points

$$\frac{\partial L(y, \lambda)}{\partial y} = GI(y_k) - 2\lambda A \Delta y = 0 \tag{12}$$

$$\frac{\partial L(y, \lambda)}{\partial \lambda} = \Delta y^T A \Delta y - d^2 = 0 \tag{13}$$

With (12) it can be written

$$\Delta y = y_{k+1} - y_k = \frac{1}{2\lambda} A^{-1} GI(y_k) \tag{14}$$

and from this

$$y_{k+1} = y_k + \frac{1}{2\lambda} d_k \ , \quad d_k = A^{-1} GI(y_k) \tag{15}$$

For a good choice of *metric* A all the gradient numerical optimization methods are obtained. In performances termes, the choice of weighting factor for d_k is realized in a optimal manner if α_k results like a solution to

$$max_\alpha \{I(y_{k+1}) = I(y_k + \alpha d_k)\} \tag{16}$$

The stop condition for the optimization algorithm (when $k \in Z_+$) with respect to the condition I(y) round about y^*, is

$$\|y_{k+1} - y_k\| < \varepsilon_1 \tag{17}$$

or

$$| \ I(y_{k-1}) - I(y_k) \ | < \varepsilon_2 \tag{18}$$

or

$$\|GI(y_{k+1})\| \preceq \varepsilon_3 \tag{19}$$

For the presented methods a remarkable quality is the convergence speed which could be appreciated by

$$\lim_{k \to +\infty} \frac{\|y_{k+1} - y^*\|}{\|y_k - y^*\|^p} = c \tag{20}$$

With respect to the parameter p, two cases are considered:

$$c \in (O,1) \ , \ p = 1 \Longrightarrow \textit{linear convergence}$$
$$c \in (0,1) \ , \ p > 1 \Longrightarrow \textit{superlinear convergence} \tag{21}$$

In conclusion, all gradient methods are standed out by the evaluation modality for d_k. It is possible to considerate a different approach which express the optimization condition in a canonical form by utilization of simple mathematical operations and maintains the most efficacious search direction (the direction of gradient).

If the condition is expressed in a quadratic form and it is centered in the principal axes of the space R^n then a possible research strategy is given by

$$y_{k+1} = y_k + \lambda_k^{-1} GI(y_k) \tag{22}$$

where λ_k represents hessian's eigenvalues of function I(y), with $\lambda_k > 0$.

The length of step α_k is established by $\alpha_k = \lambda_k^{-1}$. It is possible to prove that the search by the gradient GI(y) direction is continued until the intersection of principal axes.

2.2.Designing of Stochastic Nonlinear Supervisor

For the stochastic case, the problem (6) is given by

$$max_{y \in D_{adm}(\omega)} \{I(y) = \hat{\Theta}^T(\omega) f(y)\} \tag{23}$$

where $\hat{\Theta}(\omega)$ and D_{adm} are defined for all realizations of random variable ω in the events space Ω. $y^*(\omega_i)$ represents the solution of problem (23) for the realization ω_i and $I^*(\omega_i)$ is the value of condition I(y) for $y=y^*(\omega_i)$. It is possible to prove that $y^*(\omega_i)$ and $I^*(\omega_i)$ are random variables (i=1:m with m dimension of events space Ω). Although, the admissible

solutions set is considered being non-empty for the realisations $\omega_i \in \Omega$.

Therefore, it is possible to write

$$D_{adm}(\omega) = \cap_{\omega_i \in \Omega}\{D(\omega_i)\} \neq \emptyset \tag{24}$$

where $D_{adm}(\omega_i)$ is the process operating domain for $\omega_i \in \Omega$. D_{adm} could be estimated by

$$D_{adm}(\omega) = \{y \mid Ay \preceq b(\omega)\ ;\ y \succeq 0\} \tag{25}$$

with A constant matrix and b(ω) a random variable, or

$$D_{adm}(\omega) = \{y \mid A(\omega)y \preceq b(\omega)\ ;\ y \succeq 0\} \tag{26}$$

with (A,b) discrete random variables with the values $\{A_i,b_i\}_{i=1:m}$. The problem (23) admits a solution just for an equivalent reformulating in determinist terms. For this, two from the most usual techniques, *optimality stochastique conditions*, are presented:

Minimal Risk Condition

The problem is

$$min\{\alpha\} = min_{y \in D_{adm}(\omega)}\{p(I(y) = \hat{\Theta}^T(\omega)f(y) < I_0\} \tag{27}$$

The equation (27) represents a problem which minimizes the risk α for I(y) having values larger than a imposed value of condition, I_0.

The determinist equivalent for (27) is obtained from a procedure which is presented below. The mean value of variable $\hat{\Theta}(\omega)$ realisations is:

$$m^T = [E\{\theta_i\}]^T \tag{28}$$

and the covariance matrix is

$$V = \{v_{ij}\}\ ,\ v_{ij} = E\{(\theta_i - m_i)(\theta_j - m_j)\} \tag{29}$$

The problem

$$max_{y \in D_{adm}(\omega)}\{I_m(y) = m^T f(y)\} \tag{30}$$

had an optimal value in $I_m^*(y) = m^T f(y^*)$ where y^* is the solution of problem (30).

Imposing $I(y)<I_0$ with $I_0=I_m^*$, it obtains

$$\alpha = p\{\hat{\Theta}^T(\omega)f(y) < I_0\} = p\left\{\frac{\hat{\Theta}^T(\omega)f(y) - m^T f(y)}{\sqrt{y^T V y}} < \frac{I_0 - m^T f(y)}{\sqrt{y^T V y}}\right\} = \\ = \Phi\left\{\frac{I_0 - m^T f(y)}{\sqrt{y^T V y}}\right\} \tag{31}$$

where

$$\Phi(t) = \sqrt{\frac{2}{\pi}}\int_0^t e^{-\frac{z^2}{2}}dt \tag{32}$$

is the distribution function for z.

For Φ an increasing function and $V>0$ the problem (31) is reformulated to

$$max_{y \in D_{adm}(\omega)}\left\{\frac{m^T f(y) - I_0}{\sqrt{y^T V y}}\right\} \tag{33}$$

which could be solved by well-known methods (gradient methods).

Imposed Risk Condition

This condition is defined by

$$max_{y\in D_{adm}(\omega)}\{I'(y)\}\ \ p\{I(y)=\hat{\Theta}^T(\omega)f(y)<I'\}=\alpha_0 \tag{34}$$

In this case, the risk is fixed at the known value α_0 and it is desired the determination of value y* (which maximizes the condition I'(y)). So, it is possible to write

$$\alpha_0 = p\{I(y)<I'\} = p\left\{\frac{\hat{\Theta}^T(\omega)f(y)-m^Tf(y)}{\sqrt{y^TVy}} < \frac{I'-m^Tf(y)}{\sqrt{y^TVy}}\right\} = \Phi\left\{\frac{I'-m^Tf(y)}{\sqrt{y^TVy}}\right\} \tag{35}$$

$$\frac{I'-m^Tf(y)}{\sqrt{y^TVy}} = d \implies I' = m^Tf(y)+d\sqrt{y^TVy} \tag{36}$$

Now the problem (34) is written

$$max_{y\in D_{adm}(\omega)}\left\{I' = m^Tf(y)+d\sqrt{y^TVy}\right\} \tag{37}$$

The problem (37) is determinist. It's solution could be obtained by utilization of gradient optimization methods.

3.STUDY OF A REAL CASE

3.1.Implementation Possibilities

Two implementation possibilities for an intelligent supervisor exists [3],[4].

1. The OFF-LINE implementation where the supervisor and the operating system are disconnected. The process information are introduced by the human operator and the decision will be transfered in the same manner.

2. The real-time implementation where the supervisor and the operating system are interconnected. In this case, the process information and the decisions of supervisor are manipulated during the process operating.

3.2.Supervised Control of a Pyrolise Reactor

A chemical reactor for ethylene fabrication by pyrolise of oil in continuous mode is considered. The oil and the water vapors are the reactants. The pyrolise reaction is extremely complex and is developed in high temperature (840°C) and low pressure (4 bars) conditions. The result of reaction is a multiconstituents mixture which contains the useful products (ethylene)[1].

The control parameters for the lower level, represented by the ensemble of control systems, are: y_1-oil flow, y_2-water flow, y_3-reaction pressure and y_4-reaction temperature. The existent control devices assures the nominal functioning mode at the reference values r_1=1400 m^3/h, r_2=450 m^3/h, r_3=4 bar and r_4=800 °C. The control objective is to maximize the ethylene production an hour.

The qualitative parameter z is the ethylene concentration in the mixture at the output of reactor. The most important disturbance (ω) of reaction is the oil quality.

It is possible to imagine an information acquisition materialized by a measured experimental values collection $\{(y_i, z_i)\}_{i=1:M}$ for the control model identification. The acceptable structure of control model is

$$\hat{z}(\hat{\Theta}, y) = \hat{\theta}_0 + \hat{\theta}_1 y_1 + \hat{\theta}_2 y_2^{-1} + \hat{\theta}_3 y_3 + \hat{\theta}_4 y_4 + \hat{\theta}_5 y_4^2 \tag{38}$$

Using the IDE part of program IDEOPT (a software for control model estimation and optimization developed by us) and the least-squares identification method, the obtained model is

$$\hat{z}(\hat{\Theta}, y) = 0.138*10^2 + 0.573*10^{-3} y_1 + 0.129*10^{-2} y_2^{-1} - 0.365*10^{-2} y_3 + + 0.365*10^2 y_4 + 0.29*10^{-3} y_4^2 \tag{39}$$

The quality condition is given by the ethylene production an hour

$$I(y) = y_1 \hat{z}(\hat{\Theta}, y) \tag{40}$$

Using the OPT part of program IDEOPT the optimization problem is solved:

$$max_{y \in D_{adm}} \{I(y) = y_1 \hat{z}(\hat{\Theta}, y)\} \tag{41}$$

The result is

$$y_1^* = 1569.7\ m^3/h\ ,\ \ y_2^* = 443.7\ m^3/h\ ,\ \ y_3^* = 4.49\ barr\ ,\ \ y_4^* = 825.6\ ^\circ C. \tag{42}$$

The ethylene production an hour has the maximum value I(y*)=427.79 m^3/h.

Considering T(y) the mean computed value of ethylene production an hour (T(y)=409.69 m^3/h) it is possible to evaluate the relative increasing of production

$$\eta = \frac{I(y^*) - I_{med}(y)}{I_{med}(y)} = 0.044 \tag{43}$$

So, the supervisor had improved the process performances with 4.4 % comparatively with a normal exploitation.

REFERENCES

1. Bloch, Th., A. Haglund, J. F. Aubry, F. Simonet (1994), Modélisation des systèmes hybrides de production chimique, *ADPM 94 Bruxelles.*

2. S. Calin, M. Tertisco, D. Popescu, C. Popeea (1980), Process Control Optimization, *Ed. Tehnica, Bucuresti*

3. G. Beverige and G. Schecter (1972), Optimization Theory and Practice, *McGraw-Hill Book Company, New York.*

4. D. Himmelblau (1972), Applied Nonlinear Programming, *McGraw-Hill Book Company, New York.*

Conservative Agents in GA-deceptive Games

H. Dawid * A. Mehlmann

In this paper we analyze the learning behavior of genetic algorithms (GAs) in a special class of evolutionary games. We restrict our attention to situations where the equilibrium strategies have comparatively small fitness values in the case of random populations. Due to this deceptive game structure adaptive systems like a GA will have problems with attaining equilibrium states. We name this class of evolutionary games "GA-deceptive". Various simulations in GA-deceptive games are presented and discussed. Finally we demonstrate that the convergence problems of GAs in GA-deceptive games may be overcome by inserting conservative agents into the population.

1. Introduction

As the Nash equilibrium concept is not able to explain why the players will coordinate on a certain equilibrium, many game theorists have shifted their attention to the field of evolutionary game theory. A large number of different models of learning have been proposed and their convergence behavior has been analyzed. In this paper we restrict our attention to models, where each player represents a whole population of individuals competing against the members of another population. In contrary to the standard learning models of evolutionary game theory which are in general mean value models we represent each member of the population explicitly by a binary string and model the learning with a simple genetic algorithm (see Goldberg [1]).

GAs have been used in several models to describe interacting populations in economic or game theoretic situations. We refer to Dawid and Mehlmann [2,3], Dawid [4] or Holland and Miller [5] for an extensive review of papers dealing with the behavior of GAs in economic systems. Many of these models have shown that GAs are well suited to describe the evolution of a population of interacting adaptive economic agents. Besides this empirical work also some analytical results have been derived to characterize the behavior of GAs in economic systems (Dawid [6], Dawid and Hornik [7]).

All these results, empirical and analytical, deal with situations where the payoff of an agent is determined by the interaction with other agents in his own population. In this paper we restrict our attention to situations where two populations interact with each other but no interaction happens within the populations. We model such a situation by assuming that each agent plays an evolutionary game against the mean strategy of the other population. A similar setup was used in Dawid and Mehlmann [3]. Unfortunately the theoretical results are not directly applicable in this case and therefore we have to rely on simulation results. As demonstrated in Dawid and Mehlmann [2] for one popula-

*This research was supported by the Austrian Science Foundation under Contract No. P9112-Soz.

tion contests we introduce agents with differing adaptation strategies and show in which situations a heterogeneous population structure improves the learning in two population contests.

The paper is organized as follows. In Section 2 we describe the used algorithm, in Section 3 we present simulations with different population structures and we give some concluding remarks in Section 4.

2. A Genetic Model of an Adaptive Population

We consider a situation where two populations play an evolutionary game against each other. The game is defined by the set of pure strategies I with $|I| = m$ and the payoff matrix $A = [a_{ij}]_{i,j \in I}$. We denote the two populations by P^1 and P^2. We assume that both populations consist of n agents. Each agent plays a mixed strategy $s \in \Delta^m$, where $\Delta^m = \{s \in \mathbb{R}^m : s_i \geq 0, \sum_{i \in I} s_i = 1\}$ is the simplex in the $\mathbb{R}^m$. Each agent is represented by a binary string of length l which encodes the mixed strategy of the agent. Let Ω be the set of all binary strings with length l then there is a unique mapping $s : \Omega \to \Delta^m$ such that $s(k)$ is the mixed strategy of an agent represented by a string $k \in \Omega$. Let further $\phi^i \in \Delta^r$, $r = 2^l$, $i = 1, 2$ be the state of P^i (i.e. ϕ^i_k is the relative frequency of the string $k \in \Omega$ in P^i). We call the strategy $s^i \in \Delta^m$ with

$$\bar{s}^i = \sum_{k \in \Omega} \phi^i_k s(k) \tag{1}$$

the population strategy of P^i.

The fitness of the binary string $k \in P^i$ is given by the payoff an agent using strategy $s(k)$ receives when playing against the population strategy s^j with $j \neq i$. The fitness of a string k in P^i depends thus on the state of P^j and we get the following formula for the fitness functions f^1, f^2:

$$f^i_k(\phi^j) = s(k)' A s^j \qquad k \in \Omega,\ i, j \in \{1, 2\},\ i \neq j. \tag{2}$$

Consider a scenario where the same game is repeatedly played between the members of the two populations. The adaptive economic agents (or technically speaking the binary strings representing the agents) will probably change their strategies from period to period in order to react to obtained information such as the own payoff, the payoff of other agents, strategy recommendations from other agents or just new ideas. We model this changing of strategies by applying the three standard operators (see Goldberg [1]) proportional selection, one point crossover and mutation to the binary population The two populations are initialized randomly and afterwards the three operators are applied again and again until a given number of iterations has been reached. Note that the fitness values of all the strings have to be calculated anew in each period as the state of the opposed population changes in general from period to period.

Two different arguments may be used to justify this procedure. A technical justification lies in the fact that GAs have proven to be quite successful in solving complex optimization problems and therefore it may be assumed that they are well suited to find good "solutions" of such interactive problems too. On the other hand each operator may be interpreted in an economic way and the whole algorithm may therefore be seen as

an economic model of adaptive learning. The implicit rationality assumpfions which are implied by this approach are discussed in some detail in Dawid [4].

3. Conservative Agents

One of the main advantages of the use of population learning models like genetic algorithms in contrast to "mean value" models lies in the fact that we are able to consider situations where the different agents in the populations have different learning rules. A very simple differentiated learning behavior is the case where some of the agents do not adapt their strategies at all whereas all the other agents update their strategy according to the procedure described in section 2. We call the agents who use their initial strategy also in all following periods "conservative agents". Technically speaking a conservative agent corresponds to a string which is directly transferred from the population at time t to the population at time $t+1$ without any application of genetic operators. Nevertheless this string may be selected into the mating pool and thus a copy of the string may enter the "ordinary" population. With other words, a conservative agent is unaffected by his surroundings. However he himself can influence the other agents in his population.

Conservative agents have already been used in simulations where a so called "GA-deceptive" game is played within one population (see Dawid and Mehlmann [2]). A game is called "GA-deceptive" if some equilibrium strategies have a very low fitness compared to some non equilibrium strategy whenever matched with a random population. In such games a GA without mutations will have considerable problems in learning an equilibrium. The mixed strategies which attach a high weight to the initially low earning equilibrium strategies are extincted due to selection effects and can not be retained afterwards when they become high fitness strategies. However in Dawid and Mehlmann [2] it is shown that the insertion of only one conservative agent for each pure strategy will overcome the problem and lead to the convergence of the GA to an equilibrium. In the one population setup the introduction of a small number of conservatives has an effect similar to the introduction of a small mutation probability. The economic interpretation is however quite different. A high number of conservative agents characterizes a population with a small propensity to adapt to new circumstances, whereas a positive mutation probability means that the population is innovative and that some agents use strategies whose effects are not known to them.

In the present paper we deal with two population contests. We will use again a GA-deceptive game to illustrate the effect of the conservative agents in this setup. Consider a symmetric 3×3 normal form game with payoff matrix

$$A = \begin{pmatrix} 2 & 0 & 1.5 \\ 0 & 6 & 1 \\ 0 & 36 & 0 \end{pmatrix}. \tag{3}$$

The unique Nash equilibrium of this game is the symmetric equilibrium $((1,0,0),(1,0,0))$. In our simulations we use the following assumptions. Both populations consist of 100 binary strings of length $l = 12$. The first, second and last 4 bits of a binary string k are the binary representations of three integers k_1, k_2, k_3. The components of the mixed

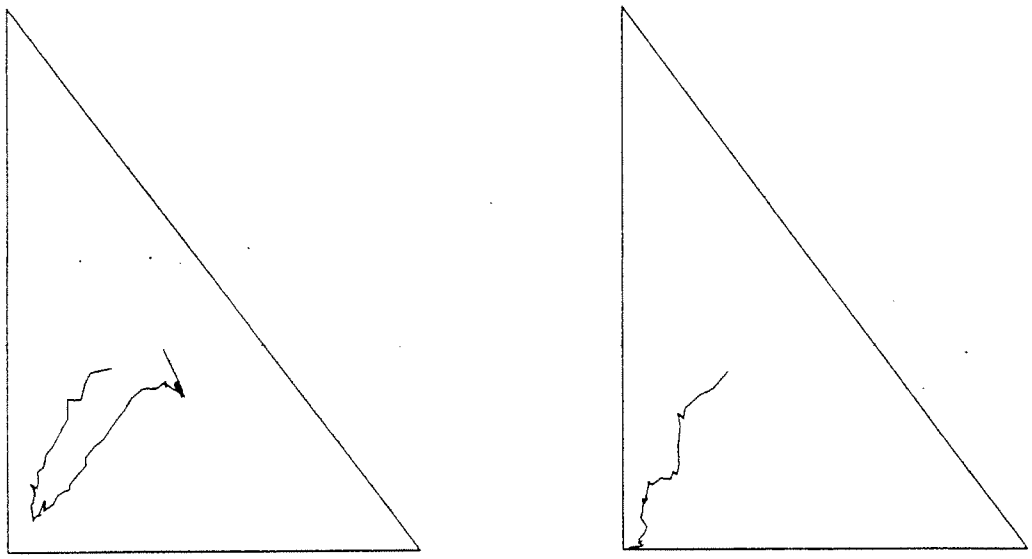


Figure 1. The population strategies $\bar{s}^1$ and $\bar{s}^2$ in a simulation with a GA with parameter values $n = 100$, $l = 12$, $\lambda = 1$ and $\mu = 0$.

strategy encoded by the string k is given by

$$s_i(k) = \frac{k_i + .001}{k_1 + k_2 + k_3 + 0.003}.$$

Obviously only a finite subset of Δ^3 can be represented with this kind of coding. If we initialize a population of strings randomly the expected population strategy is $m = \frac{1}{3}(1, 1, 1)$. Therefore the second pure strategy has a fitness of 12 which is much higher than the fitness 1.167 of the first pure strategy. Thus the considered game is GA-deceptive.

In figure 1 we show the trajectories of the two population strategies $\bar{s}_t^1$ and $\bar{s}_t^2$. It can be seen quite clearly that the equilibrium is not reached. The population strategy of P^2 approaches the third pure strategy more rapidly than P^1. When $\bar{s}^2$ gets close to $(0, 0, 1)$ the strings in P^1 which attach more weight to the first strategy get a higher fitness and $\bar{s}^1$ "turns around". There are however no more strings in the population who encode a mixed strategy which puts a high weight to the first pure strategy. Thus $\bar{s}^1$ gets stuck at $\bar{s}^1 = (0.375, 0.375, .25)$. Note that $(0, 0, 1)$ is the best reply to this mixed strategy, which means that the second population acts optimal under the given conditions. We get a very asymmetric situation where the agents in the first population get a mean payoff of 0.7 whereas the mean payoff in the second population is 13.5.

If we add now one conservative agent for each pure strategy to both populations the situation changes completely and the asymmetry disappears. We show such a simulation in figure 2. Similar to the case of one population contests the conservative agents play the role of a constant memory and prevent the complete loss of certain genetic material. In this simulation, population 1 approaches the third pure strategy more rapidly than population 2. When $\bar{s}^1$ is near $(0, 0, 1)$ the first pure strategy is again the strategy with the highest fitness in P^2. The conservative agent playing this pure strategy spreads now rapidly in P^2 and due to crossover effects also some mixed strategies with a high weight on

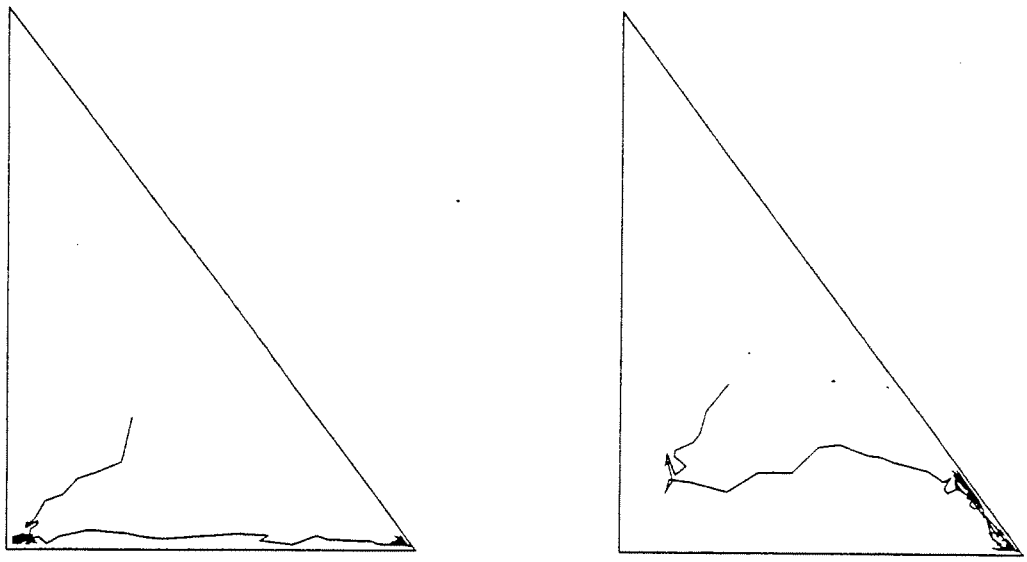


Figure 2. The population strategies $\bar{s}^1$ and $\bar{s}^2$ in a simulation with a GA with parameter values $n = 100$, $l = 12$, $\lambda = 1$, $\mu = 0$ and one conservative agent for each pure strategy in both populations.

the first pure strategy appear. Finally s^2 approaches $(1, 0, 0)$ and stays near this strategy. As $(1, 0, 0)$ is a best reply to itself the first pure strategy is now also the strategy with the highest fitness in P^1 and $\bar{s}^1$ approaches $(1, 0, 0)$ as well. The system is now near the unique Nash equilibrium where all agents act optimal. We like to emphasize that in this model the addition of mutations did in some cases not lead to the convergence towards the equilibrium. Due to space constraints we are however not able to compare the two approaches in more detail.

Let us finally consider a case where conservative agents are present only in one of the two populations. In figure 3 we show a simulation where only P^2 contains conservative agents. Again both population strategies evolve towards the third pure strategy in the beginning, however the second population is able to adapt to the changing environment as the first pure strategy becomes optimal. The first population has completely lost all the genetic material needed to get to $(1, 0, 0)$ and converges to $\bar{s}^1 = (0.33, 0, 0.67)$. The conservatives help the agents in the second population to get a mean payoff of about 1.65, whereas the mean payoff of the first population is only 0.66.

4. Conclusions

In this paper we have used a simple genetic algorithm to model two populations of adaptive economic agents who receive some payoff by interacting with agents in the other population. Our results may indicate that conservative agents are helpful for a population and improve the learning behavior of all the agents in the population. However, we like to stress that conservative agents may have quite unintended effects in non GA-deceptive games, where also a regular GA without mutations may find an equilibrium. In such a case the addition of conservative agents may lead to oscillatory and rather chaotic looking

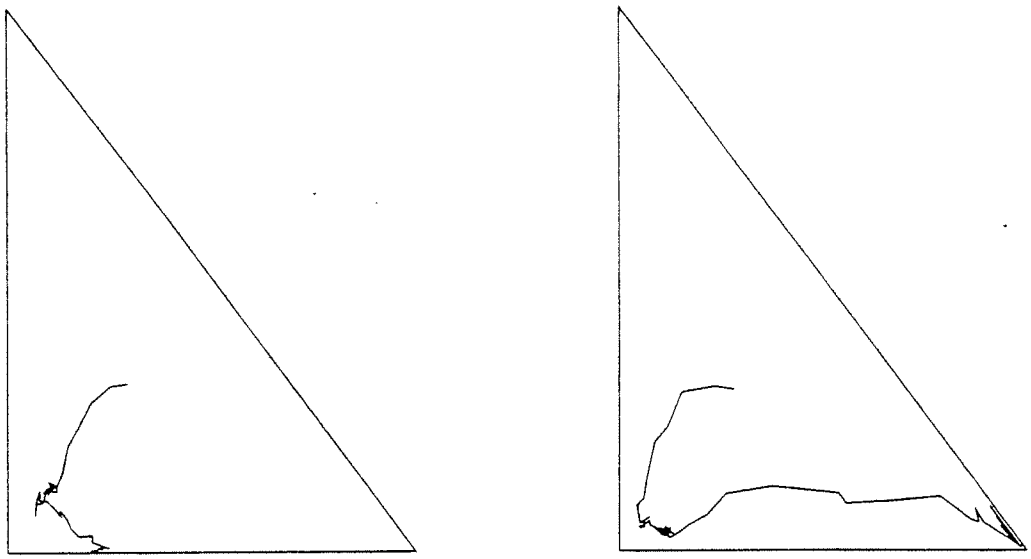


Figure 3. The population strategies s^1 and s^2 in a simulation with a GA with parameter values $n = 100$, $l = 12$, $\lambda = 1$, $\mu = 0$ and one conservative agent for each pure strategy in the second population.

behavior of both population strategies. The effect of the conservative agents depends crucially on the game which is played and it should in our opinion be an interesting task to look for economic explanations of the success and failure of this kind of learning rule in different environments.

REFERENCES

1. D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley (1989).
2. H. Dawid and A. Mehlmann, "Genetic Learning in Strategic Form Games" Research Report, Vienna University of Technology (1994).
3. H. Dawid and A. Mehlmann, "Two Population Contests and the Language of Genetics", to appear in G. Paun (ed.), Mathematical Linguistics and Related Topics, Editura Academiei, Bucharest, 1995.
4. H. Dawid, Genetic Learning in Economic Systems, Analytical Results and Applications, Ph.D. Thesis, Vienna University of Technology (1995).
5. J.H. Holland and J.H. Miller, American Economic Review: Papers and Proceedings of the 103rd Annual Meeting of the American Economic Association, (1991) 335.
6. H. Dawid, "A Markov Chain Analysis of Genetic Algorithms with a State Dependent Fitness Function", to appear in Complex Systems (1995).
7. H. Dawid and K. Hornik, "The Dynamics of Genetic Algorithms in Interactive Environments", to appear in Journal of Microcomputing Applications (1995).

The strings that tie simulators together: the Message Passing Paradigm as instantiated by PVM

W. Weisz

Vienna University of Technology, Computing Services, Dept. for High Performance Computing,
Wiedner Hauptstrasse 8-10, A-1040 Wien, Austria

The distribution of cpu-intensive programs over a multitude of processing elements is becoming more and more popular as the price of single- and multiprocessors is decreasing. On the other hand the systems modelled by simulation involve generally many activities that go on in parallel in real life. So it is quite natural that simulation problems should be ported to clusters of computers.

1. INTRODUCTION

In order to solve a single simulation problem 3 different scenarios for using multiple processors can be envisaged:

1. a master distributing the data to several instantiations of the same program and collecting the results for final manipulation (Single Program Multiple Data),
2. differing tasks generated by one master are mapped onto the available processors,
3. tasks started independently on different computers (some may reside on the same computer) make a rendez-vous and create links over which they communicate (initial values, results, states, etc.), one of those tasks may assume a coordinating or data-collection task.

All three models may be programmed using the message passing paradigm as provided by PVM.

1.1 A short introduction to PVM

The portable message-passing programming system PVM (Parallel Virtual Machine) (1) was designed to allow the linkage of separate host computers of the same or different architectures running some Unix implementation and connected over a communication network like Internet. The machines may be on the same desk or at different sites spread over the whole earth globe. A single Parallel Virtual Machine can be composed of a selection of workstations, vectorprocessors and multiprocessors. Any PVM process on any of those computers can communicate with any other one, as can processes running on a single host. The whole constitutes therefore a single *Virtual Machine* which the user programs at the application level in FORTRAN, C or C++ using language-level calls to the message passing library.

A Parallel Virtual Machine is created by starting a single daemon (*pvmd*) on each of the participating hosts as a user process. The daemons control the communication

between the "productive" processes, the startup of processes on remote computers and the inclusion of running processes into the Virtual Machine. The individual processes (including the *pvmds*) are called *tasks*, and each task is attributed a *task-id* (*tid*) that is unique within a Parallel Virtual Machine.

In the following description verbs in *italics* indicate actions that have an equivalent function in the PVM library. A process becomes a task when it is *spawned* from within a Parallel Virtual Machine either from a PVM console (a user callable process that allows the user to dynamically define and eliminate the participating hosts) or from user-written tasks. Processes initiated from the Unix environment attach themselves to the Parallel Virtual Machine, and, if successful, are returned the new task's *tid*. If a program is *spawned* by another PVM program, the parent gets back the *tid* of the child, and the child can retrieve ist parent's *tid*. The parent can then communicate it to its other children using messages.

Tasks communicate by *packing* their information into messages that are *sent* to a task with a specified *tid* or *broadcast* to a list of tasks or all the tasks that belong to a *group*. The *receive*ing task retrieves the message and unpacks it. Finally the task *exits* the Parallel Virtual Machine, or leaves it by just vanishing but this is bad programming style and may produce some unnecessary overhead for the local *pvmd*.

2. DISTRIBUTED SIMULATIONS

With the tools supplied by PVM it becomes straightforward to use message passing as a backplane to distribute a single simulation task to multiple hosts. Writing a PVM program from scratch doesn't constitute a problem.

Most simulation packages are available only as binaries which do not allow to apply any modifications to the code. But many of them allow the call to user-written subroutines from the problem description script. If this user code can be a FORTRAN, C or C++ program it can contain the PVM functions.

2.1 Parallelizing a single simulation algorithm

As described in detail in (4) a simulation program like MATLAB can call in its script a function that *spawns* multiple identical slave processes. These may be a copy of the parent process or a different program. Since PVM provides the tools to recognize whether a program has a parent, it is easy to write a single program that can decide whether it is the master or a slave.

The master program then divides the data or the computing interval and sends to each child its share. The messages are received and processed by the slave processes which finally return the resulting data to the master, which then does the statistics or distributes new data to the child processes.

Some problems require the exchange of information between the computing instances. In this case the master has to inform each slave of the *tid* of any of the other slaves. Then there can be a direct exchange of messages between the slave tasks, without the

intervention of the master. The latter will only collect results necessary for computing the statistics.

The master may also serve as the man-machine interface providing interactive monitoring and intervention. The changed data are then forwarded to the respective slave process.

2.2 Distributing a problem to differing simulators

There are problems that cannot be solved using a single simulator. For instance the simulation of the behaviour of a system consisting of mechanical and electrical parts (e.g. ABS breaks) needs the coupling of 2 different simulators (one based on ACSL and the other on Simulink) which can run in parallel, each on its own host.

The theoretical concept behind this approach is the *Model Interconnection Concept (MIC)* (2) which describes the development of a simulation model by using independent submodels that interact in the course of the computation. This approach can be directly mapped to the cooperative work of independent processes which from time to time get in touch with each other and exchange information. The vehicle of the transmission of information from one submodel to the other can easily be implemented by using a message passing system.

In order to get through the whole problem in a reasonable time it is necessary that the submodels are not to tightly coupled. If there is too much message exchange the communication will dominate over the computing time, and it is several order of magnitudes slower than the processor cycles. If there is much communication between the submodels another paradigm is better suited: such a simulation model should be implemented on a shared memory multiprocessor.

The Model Interconnection Concept served as the basis for the development of the simulation package **mosis** (3) which has been implemented using PVM for inter-processor communication, and can thus be used on workstation clusters.

In this model there is still a master that spawns the submodels to some predetermined processor and sends it the data as well as the *tid*s of the other slave tasks it needs to get in touch with. The submodels can be distributed to the hardware that is most appropriate to its software implementation. Some submodels may even need hosts with special features like interfaces to external systems (e.g. Hardware in the Loop or Man in the Loop).

2.3 Coupling independent simulators

The most general view of the Model Interconnection Concept is to map the submodels to really independent processes that start up independently and only in the course of the computation try to connect to other simulators running somewhere else.

Such an approach as awkward as it is at first glance may present important advantages. First it allows to start submodels that need some time to build up their database before they can meaningfully communicate with other tasks. Such a process can be started on an available host even so there are not yet enough resources to start

the whole model. By the time the initial database is created the other tasks may have had the chance to start.

Whenever the appropriate time has come each simulator will enter the Parallel Virtual Machine. Then it signals its presence and its *tid* to the others which may in turn communicate their whereabout to the newcomer. Finally the exchange of data between any two of them or through broadcasting can start.

Another scenario where such an approach makes sense is the case of hosts distributed over a large distance. Since in large networks the chances that intermediate segments may be down are not so low, programs that can be self-sufficient over a certain time period may run independently, and only contact the Parallel Virtual Machine when they need to communicate. Thus failures in the network may have a much lower impact on the overall problem solving, as if the complete Virtual Machine must be up during the whole run.

The feature of PVM that allows this approach is the existence of Dynamic Process Groups. An independent process can hook to the Parallel Virtual Machine by requesting its *tid*. If this is successful, i.e. there is a *pvmd* running on his host, it may then *join a group*. A group is defined by a name, i.e. a character string, attached to it. If the current task is the first one making a request for this group, the group is created. Tasks belonging to the same group may get in touch with each other, without knowing the ones *tid* beforehand nor having any common ancestor who might pass the information down. Any task that is a member of a group can broadcast a message to all the other tasks in the group without knowing their *tid*. In this message it can include its *tid* but even this is not necessary because the retrieving tasks may get this information from the PVM system. Tasks that have retrieved the message can then update their table of available simulators and may thus get in touch with any of them whenever needed.

3. CONCLUSION

Using PVM the embarassingly parallel nature of most simulation models can be mapped in a straightforward way onto programs that run on clusters of hosts, thus reducing the time to solve a problem drastically.

BIBLIOGRAPHY

1. Al Geist, et al.: PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing, The MIT Press, Cambridge, Massachusetts (1994).
2. G. Schuster, F. Breitenecker, I. Husinsky: The Model Interconnection Concept and ist Effects on Parallelisation of Simulation Models, Proc. ESS 92 - European Simulation Symposium Dresden, SCS Publication San Diego (1992), 93-97.
3. G. Schuster, F. Breitenecker: Coupling Simulators with the Model Interconnection Concept and PVM, Proceedings of the EUROSIM 95 Congress, Vienna, Austria, Elsevier Science B.V. (1995) 321-326.
4. S. Pawletta, T. Pawletta and W. Drewelow: Distributed and parallel simulation in an interactive environment, Proceedings of the EUROSIM 95 Congress, Vienna, Austria, Elsevier Science B.V. (1995) 345-350.

The Real-time Simulation of Multiframe System on Multiprocessors

Du tieta[a], Huang Kedi[b], Zha Yabing[b]

[a]Dept. of Computer Science, [b]Dept. of Automatic Control
National University of Defense Technology
Changsha, Hunan, P.R. of China (410073)

Abstract

The principles of processor allocation for the paralell simulation of multiframe systems on multiprocessors are discussed in this paper. It is pointed that in order to decrease the overhead of interprocessor communication, each subsystem of multiframe system should be parceled into different processor(s) to be solved concurrently. The control flow of multiframe system simulation is described in this paper.

key words:
multiframe system, simulation, processor allocation, numerical integration

In the applications of the real-time simulation of continuous systems, since the varying rates with time of the subsystems of the physical system are different each other, sometimes they differ very much, so different integration step sizes are usually used for different subsystems to meet the requirements of the speed and accuracy and etc. during simulation. Therefore, the simulation models with multiple frame rates ("multiframe system" for short) are built. The problems of the parallel simulation of the systems with single frame rate on multiprocessors were discussed in the other papers[1], here we are going to discuss some problems about the parallel simulation of the multiframe systems.

1. The control flow of multiframe system simulation

Let M is the model of a multiframe system described by state equations as follows:

$$\begin{cases} M: \{t, U, X, Y, f, g\} \\ X' = f(t, X, U) \\ Y = g(X, U) \end{cases}$$

where t is the independent variable, U is a set of inputs, $X=\{x_1, \cdots, x_n\}$ is a set of state variables, $f=\{f_1, \cdots, f_n\}$ are state transfer functions, Y is a set of outputs, g are output functions.

For convenience, we assume that there are only two subsystems included in M. Namely, system M can be decomposed into two subsystems: one is

subsystem F that varies fast with time, the other is subsystem S that varies slowly with time. It might be assumed that F has the state variables $X_F=\{x_1, \cdots, x_i\}$ $(1<i<n)$, the state equations are

$$X'_F=f_F(t, X, U)$$

and S has the state variables $X_S=\{x_{i+1}, \cdots, x_n\}$, the state equations are

$$X'_S=f_S(t, X, U)$$

Therefore, M can also be denoted by

$$\begin{cases} M: \{t, U, X, Y, f, g\} \\ X'_F=f_F(t, X, U) \\ X'_s=f_s(t, X, U) \\ Y=g(X, U) \end{cases}$$

Let the integration step size chosen for F be t_F when simulation, and the step size for S be t_S. $R=t_S/t_F$ is usually an integer which is called frame-ratio.

Following is the algorithm to solve M by numerical integration on single processor computers[2].

Algorithm 1

```
/* t_o is the starting time of simulation, t_max is the finish time,
   X(0) is the initial conditions of X at time t_o,
   X(N) is the values of X at time t_N, t_N=t_o+N*t_S (N=0,1,2,...) */
for t=t_o to t_max step t_S
  begin
    calculat-X_S(X(N), X_S(N+1), t_S) /* calculate X_S(N+1) via X(N) */
    calculat-X_F(X(N), X_F(N+1/R), t_F) /* calculat X_F(N+1/R) via X(N) */
    for i=1 to R-1 step 1
      begin
        interpolation(X_S(N), X_S(N+1), i, X̄_S(N+i/R))
          /* interpolation between X_S(N) and X_S(N+1) to obtain the
             approximate value X̄_S(N+i/R) of X_S at time t_{N+i/R} */
        calculat-X_F(X̄_S(N+i/R) ⋃ X_F(N+i/R), X_F(N+(i+1)/R), t_F)
      end
  end
```

In algorithm 1, different interpolation algorithms can be used. For example, a quadratic interpolation based on X(N), X(N+1) and X'(N) can be used.

2. The control flow of multiframe system simulation on multiprocessors

For multiprocessors, the algorithm 1 can also be used to solve M theoretically. Given m processors, all the m processors are used to solve the subsystem S at first, namely calculating $X_S(N+1)$ via X(N); and then the m processors are used to solve the subsystem F, namely to calculate $X_F(N+i/R)$, $i=1, 2, \cdots, R$. In few words, compute the integral of S with the specified numerical integration method one time at first, and then compute the integral of F R times.

There is, however, too many overhead of interprocessor communication if the two subsystems are solved by all the m processors successively. That's

because:

1. It's shown from researches[1] that the times of interprocessor communication is directly proportional to m when a task are parceled into m processors to be executed concurrently. It might be described as following formula:

$$C=k(m-1) \qquad (2\text{-}1)$$

where C is the times of communication, k is a coefficient ($k>0$). k is relational to the number of equations in model and the data dependency among the equations. C is a direct factor of the communication overhead.

2. Generally speaking, the data dependency is relatively strong within each subsystem, whereas there is no sharing algebraic variable (non-state variables) between subsystems except for state variables. In fact, even if there are shared algebraic variables, they will also be processed like state variable, computedin a subsystem, and referenced in the other subsystem by means of interpolation. Since the values to be referenced for state variables is their initial values or last values, therefore, there is no data dependency between subsystems when the right-hand functions of S and F are computed respectively. If they are computed on two groups of processors respectively, no intergroup communication exists.

It's not difficult to draw a conclusion from the above two facts that the overhead of communication will be less if the fast and slow subsystems are allocated into two different groups of processors to be solved respectively. In fact, if the number of processors allocated to the fast subsystem is m_1, and to the slow subsystem is $m_2=m-m_1$ ($m_1, m_2>0$), then, from equation (2-1) we have

$$Cp=k_1(m_1-1)+k_2(m-m_1-1)$$
$$Cs=k_1(m-1)+k_2(m-1)$$

where, Cp is the times of communication for the case of the two subsystems being solved on two groups of processors respectively, Cs is the times of communication for the case of the two subsystems being solved on the all processors successively. And we have

$$\frac{Cs}{Cp} = \frac{(k_1+k_2)(m-1)}{k_1(m_1-1)+k_2(m-m_1-1)}$$

Considering the case of $k_1=k_2$, we have

$$\frac{Cs}{Cp} = \frac{2(m-1)}{m-2} > 2$$

So we have, $Cp<Cs/2$ when $k_1=k_2$. Especially, $Cp=0$ whereas $Cs>0$ when $m=2$.

If $k_1=\alpha k_2$ (α is a constant), the similar conclusion can be drawn, namely $Cp<Cs$.

So far, we can obtain the conclusion: to decrease the overhead of interprocessor communication, the two subsystems of the multiframe system should be parceled into two groups of processors. to be solved respectively.

Certainly, F and S must be solved concurrently on different processor groups, otherwise the goal to speedup simulation by means of the techniques of parallel processing will not be achieved. Here, algorithm 1 is not suitable, because in order to compute $X_F(N+i/R)$ by means of interpolation, $X_S(N+1)$ must be calculated in advance, that means

the slow subsystem must be integrated one time at first, and then the fast subsystem are integrated R times. That is the sequence introduced by algorithm 1. To solve F and S concurrently, $X_F(N+i/R)$ $(i=1,2,\cdots,R)$ must be calculated on the other group of processors simultaneously as $X_S(N+1)$ are computed on a group of processors.

The values of $X_S(N+i/R)$ $(i=1,2,\cdots,R-1)$ invoked in the calculation of $X_F(N+(i+1)/R)$ must be obtained by extrapolations base on the values of state of X_S before t_N rather than by interpolations. For examples, the linear extrapolators based on $X_S(N-1)$ and $X_S(N)$ or $X'_S(N)$ and $X_S(N)$ can be used. Following is the parallel algorithm to solve M on multiprocessors.

Algorithm 2

```
/* The means of notations in the algorithm are identical with
  those in algorithm 1 */
for t=t_o to t_max step t_S
  par-begin /* solve the two subsystem concurrently */
    begin
      calculat-X_S(X(N), X_S(N+1), t_S) /* computer X_S(N+1) via X(N) */
    end
    begin
      for i=0 to R-1 step 1
        begin
         extrapolation(X'_S(N), X_S(N), i, X_S(N+i/R))
            /* extrapolation based on X'_S(N) and X_S(N) to obtain the
               approximate value X_S(N+i/R) */
         calculat-X_F(X_S(N+i/R) ∪ X_F(N+i/R), X_F(N+(i+1)/R), t_F)
        end
    end
  par-end
```

3. The allocation of processors in multiframe system

In the above discussions, m_1 and m_2 were implicitly assumed as integers. In practical applications, however, it is not always true if the source of m processors are shared equally by the two subsystems in accordance with their amounts of computation, (denoted by Q_F and Q_S respectively), then the numbers of processors to be allocated for the two subsystems are respectively as follows:

$$m_F = m*Q_F/(Q_F+Q_S),\ m_S = m*Q_S/(Q_F+Q_S).$$

Here, m_F and m_S may not be integers. In such case, how to allocate processors for each subsystem?

Let $m_S=m_1+\alpha$, $m_1=[m_S]$, $0<\alpha<1$, then

$m_F=m_2+(1-\alpha)$, $m_2=[m_F]$

In following discussions, we assume that m=4 for all illustrations. For example, let $m_S=4/3$, $m_F=8/3$. Then the source of processor allocated for S and F can be illustrated in Fig. 1. The dark part denotes S, andthe light part denotes F.

It's shown from algorithm 2 that as S is integrated one time, F will

be integrated R times. The identical computation tasks are performed each time for subsystem F. Therefore, for convenience on control, the configuration of processors should be identical. For instance, $2\frac{2}{3}$ processors are allocated to F each time. Otherwise, it will be more difficult for processor scheduling. The sketch maps of processor allocation for R=2 and R=4 are given in Fig. 2.

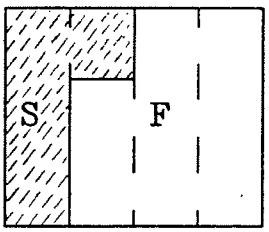
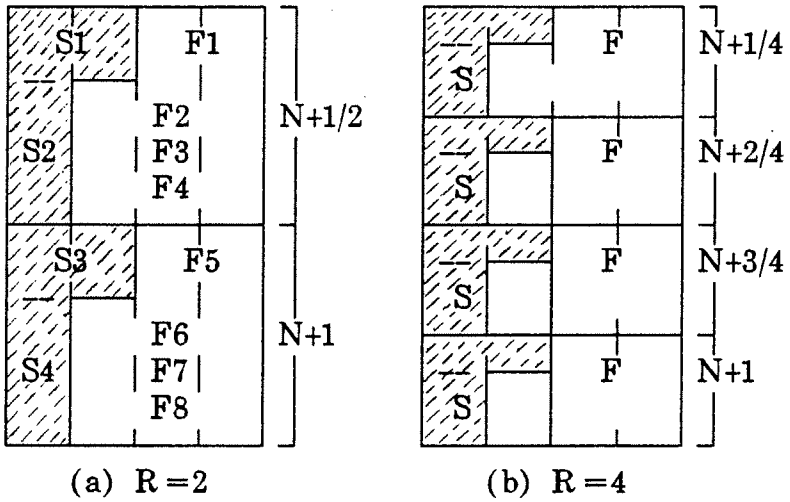


Fig. 1



(a) R=2 (b) R=4

Fig. 2

Let's explain the meaning of Fig. 2 for the case R=2.

S is divided equally into 4 parts (S1, S2, S3, S4) according to its execution order, and F into 8 parts(F1, F2, ···, F8). The codes of F5, F6, F7, F8 are identical with those for F1, F2, F3, F4 respectively. The execution flow of S is: S1 is executed on P1 and P2 (Pi is the ith processor) concurrently at first, secondly, S2 is executed on P1, and then S3 is executed on P1 and P2, at last S4 is executed on P1.

The execution flow of F is: F1 is executed on P3 and P4 concurrently at first, secondly, f2, f3, f4 are successively executed on P2, P3, P4 simultaneously, and then F5 is executed on P3 and P4, at last F6, F7, F8 are executed on P2, P3, P4.

Both some code of S and some code of F are executed on P2 alternately, therefore the processor is called "crossed processor" vividly. When R=4, some code of S and some code of F will be executed on the "crossed processor" alternately four times. The R pieces of codes of F to be executed on the "crossed processor" are identical each other, but the R pieces of codes of S are different from each other.

It seems that the loads on the processors will be more balanced if a "crossed processor" is introduced, it brings, however, many other extra problems, including:

* the control will becomes much more complicated which will brings the extra overhead,
* the times of communication will increase considerably,
* implementation becomes much more difficult.

Now we discuss how much source of processors is lost, If the "crossed processor" not be introduced to avoid the problems mentioned above.

Not introducing "crossed processor" means m_F and m_S must be integers. Following algorithm can be used to determine m_F and m_S.

Algorithm 3

$m_F = m*Q_F/(Q_F+Q_S)$, $m_S = m*Q_S/(Q_F+Q_S)$
if $m_S<1$ then
  $m_S=1$, $m_F=m-1$
else if $m_S>m-1$ then
  $m_S=m-1$, $m_F=1$
 end if
else
 $m_S=INT(m_S+0.5)$, $m_F=m-m_S$
end if

If the processors are allocated according to algorithm 3, the situation of the lost processor source is listed in Table 1.

Table 1

	m–unused–p	m–lost–s	a–unused–p	a–lost–s
$m_S<1$ or $m_F<1$	1	1/m	1/2	1/2m
$m_S>1$ & $m_F>1$	1/2	1/2m	1/4	1/4m

m–unused–p: the maximum number of the unused processors
a–unused–p: the number of unused processors on an average
m–lost–s: the maximum lost source of processor
a–lost–s: the lost source of processor on an average

It is usually not true for the case of $Q_S<<Q_F$ or $Q_F<<Q_S$, therefore, the possibilities of case 1 are relatively small. So, generally speaking, only 1/4m of the processor source is lost if the "crossed processor" not introduced. When m is relatively large, say, m=4, 8, 16 and etc., this sort of loss can be ignored, compared with the inherent unbalancing of the loads during task partitioning, and with the problems mentioned above. If m=2, however, it is suggested to use algorithm 1 for double–frame system simulation.

4. Conclusion

To sum up, for the multiframe system simulation on multiprocessors, algorithm 1 is suitable if m=2; when m>2, algorithm 2 is suitable, and two different groups of processors should be allocated to the two subsystems respectively, and the "crossed processor" should not be introduced.

The above discussion are set off on double–frame systems, the conclusion, however, are suitable for multiframe system (the number of subsystems with different frame rate is larger than 2). The algorithm mentioned above were applied in PARSIM, a software tool for paralell continuous system simulation on multiprocessors[1].

Reference

1. Du Tieta, *Study and Implementation in the Architecture of Homogeneous Parallel Simulation Computer and Its Software Supporting Environment*, Ph.D. dissertation, Aug. 1990.
2. Applied Dynamical International, *ADSIM Reference Manual*, 1989.

Efficiency of Parallel Strategies in Simulation - Results of the EUROSIM Parallel Comparison

F. Breitenecker[a], I. Husinsky[b], and G. Schuster[c]

[a] Dept. Simulation Techniques, fbreiten@email.tuwien.ac.at
[b] Computing Services, husinsky@edvz.tuwien.ac.at
[c] ARGESIM (ARGE Simulation News), guenter@osiris.tuwien.ac.at
Technical University Vienna, Wiedner Hauptstraße 8-10, A - 1040 Vienna

This contribution deals with the "EUROSIM Comparison on Parallel Simulation Techniques" ("Parallel Comparison"). First an overview about the "EUROSIM Comparisons" (published in the journal **EUROSIM Simulation News Europe**) is given, then the Parallel Comparison is introduced in detail. In the following the results of solutions sent in up to now are discussed. The paper concludes with a summary on the efficiency of the parallelization techniques used in the solutions.

1. THE EUROSIM COMPARISONS

EUROSIM, the Federation of European Simulation Societies, started in 1990 the publication of the journal **EUROSIM Simulation News Europe (SNE)**, a newsletter distributed to all members of the European simulation societies under **EUROSIM's** umbrella and to people and institutions interested in simulation. **SNE** is also part of **Simulation Practice and Theory (SIMPRA)**, the scientific journal of **EUROSIM**.
The idea of the journal **SNE** (circulation 2500; edited by F. Breitenecker and I. Husinsky, ARGE Simulation News (**ARGESIM**) , Technical University of Vienna, Austria; three issues per year) is to to dissemination information related to all aspects of modeling and simulation. The contents of **SNE** are news in simulation, simulation society information, industry news, calendar of events, essays on new developments, conference announcements, simulation in the European Community, introduction of simulation centers and comparison of simulation software, simulators and (parallel) simulation techniques.
The series on comparisons of simulation software has been very successful. Based on simple, easily comprehensible models the software comparisons compare special features of modeling and experimentation within simulation languages:

- modeling technique
- event handling
- submodel features
- numerical integration
- steady-state calculation
- frequency domain
- plot features
- parameter sweep
- postprocessing
- statistical features
- statistical processors
- control strategies
- optimization
- random numbers
- animation, etc.

Seven Software Comparisons, four continuous ones and three discrete ones (a fourth discrete comparison is in preparation) have been set up. Furthermore, a second type of comparisons, the Parallel Comparison has been initiated.
The continuous comparisons are: Comparison 1 (C1; Lithium-Cluster Dynamics under Electron Bombardment, November 1990) deals with a stiff system; Comparison 3 (C3; Analysis of a

Generalized Class-E Amplifier, July 1991) focusses on simulation of electronic circuits and eigenvalue analysis; Comparison 5 (C5; Two State Model, March 1992) requires very high accuracy computation; Comparison 7 (C7; Constrained Pendulum, March 1993) deals with state events.
The discrete comparisons are: Comparison 2 (C2; Flexible Assembly System, March 1991) gives insight into flexible structures of discrete simulators; Comparison 4 (C4; Dining Philosophers, November 1991) involves not only simulation but also different modeling techniques (e.g. Petri nets); Comparison 6 (C6; Emergency Department - Follow-up Treatment, November 1992) deals with complex control strategies; Comparison 8 (C8, locks on channels) will deal with variance reduction methods.
Up to now, 100 solutions have been sent in. Table 1 shows the number of solutions for the Software Comparisons as well as for the Parallel Comparison. The series will be continued

	C1	C2	C3	C4	C5	C6	C7	CP
SNE 0	Def							
SNE 1	5	Def						
SNE 2	4	4	Def					
SNE 3	4	3	3	Def				
SNE 4	1	5	5	3	Def			
SNE 5	4	-	1	1	2			
SNE 6	-	2	-	2	1	Def		
SNE 7	1	2	1	2	-	1	Def	
SNE 8	-	1	-	-	-	1	3	
SNE 9	-	-	-	-	-	2	3	
SNE 10	1	2	-	-	-	2	2	Def / 1
SNE 11	2	2	1	-	1	-	-	2
SNE 12	1	-	1			-	2	3
SNE 13	-	-	-			-	3	1
SNE 14	3	-	1		-	-	2	-
Total	**26**	**21**	**13**	**8**	**4**	**6**	**15**	7

Table 1: EUROSIM Comparisons, publication of solutions

2. THE EUROSIM COMPARISON ON PARALLEL SIMULATION TECHNIQUES

SNE 10 introduced a new type of comparison dealing with the benefits of distributed and parallel computation for simulation tasks. Three test examples have been chosen to investigate the types of parallelization techniques best suited to particular types of simulation tasks.
Each test example should be first solved in a serial fashion to provide a reference for the investigation of speed-up factors. The examples should then be tested using the parallel facilities (software and hardware) available. Performance should be assessed in terms of a numerical value found by dividing the time for serial solution by the time for the parallel solution (speed-up factor *f*). Wherever appropriate, serial solutions should be based on the same environment. Measurements of time should be in terms of the total elapsed time for running the task. Information must be provided about the method of parallelization or distribution of subtasks. If of interest, more than one solution for a particular test example may be offered. Furthermore, a rough indication should be provided for the program preparation time, especially for the parallel solution.
This new type of comparison addresses users of all types of parallel and distributed facilities. The spectrum may range from simulation languages, via general purpose programming languages, to

special parallel languages and from networks of workstations, via special parallel computers, to very high performance computers.
The objective is to make comparisons of different types of problems and of methods for the parallelization of simulation tasks. It is not intended that this should involve direct comparisons of the (hardware) performance of parallel facilities.

TE-1 Monte Carlo Study. This first test example deals with damped second order mass-spring system described by the equation

$$m\,d^2x(t)/dt^2 + k\,x(t) + d\,dx(t)/dt = 0,\ x(0)=0.1,\ \dot{x}(0)=0,\ m=450,\ k=9000$$

The task is to perform 1000 simulation runs and to calculate and store the average responses over the time interval [0, 2] for the motion for subsequent plotting.
Figure 1 shows some solutions, figure 2 shows the hierarchical structure of this task (the tasks can be distributed and simulated independently).

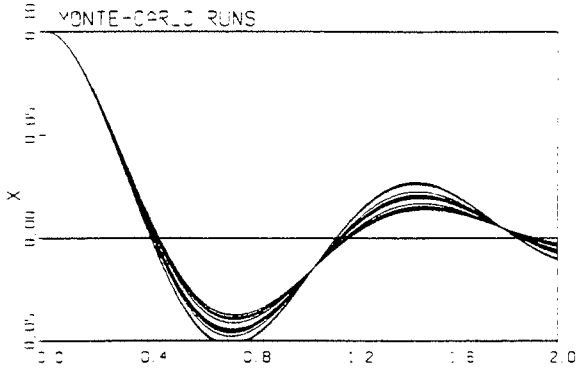


Fig.1: TE-1: Results of Monte Carlo Study

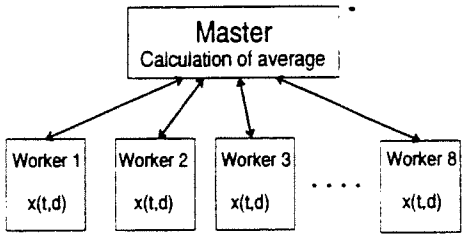


Fig 2: TE-1: Hierachical structure of tasks

TE-2: Predator-Prey Dynamics. The second test example is concerned with coupled predator-prey population models. Five predator-prey populations (v_1, v_2), (w_1, w_2), (x_1, x_2), (y_1, y_2) and (z_1, z_2) are interacting. The model equations are (all initial populations normalized to 1):

$$dv_1/dt = a_v v_1 - b_v v_1 v_2 - c_v v_1^2$$
$$dv_2/dt = -d_v v_2 + e_v v_1 v_2 - f_v v_2^2 + r_v,\quad r_v = v_2 (g_v w_1 + h_v x_1 + j_v y_1 + k_v z_1)$$
$$dw_1/dt = a_w w_1 - b_w w_1 w_2 - c_w w_1^2 + r_w,\quad r_w = w_1 (-g_w v_2 + h_w x_2),\ dw_2/dt = \ldots, \text{ etc.}$$

The task is to solve the system within the time interval *[0, 100]* in a serial fashion and in an appropriate parallel fashion and to provide the terminal values of each population.
Figure 3 shows some results in the phase plane, figure 4 shows the strongly coupled structure of the tasks, if different processors calculate the five populations. It is expected that with this example little or no improvement may be found through parallelization. Negative results are of considerable interest and should not be discarded.

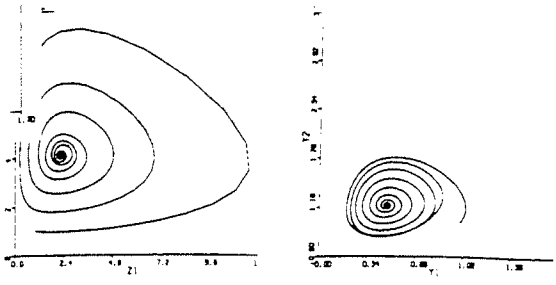


Fig.3: TE-2: Some simulation results

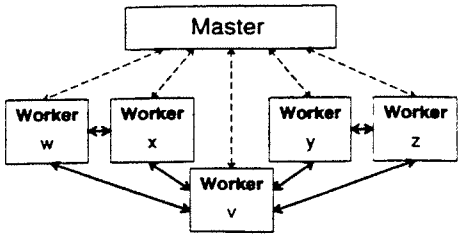


Fig 4: TE-2: Strongly coupled task structure

TE-3: Discretized PDE. The third test example is based on a second order partial differential equation describing a swinging rope with length L fixed at one end and forced at the other.

$$u_{xx}(t,x) = a\, u_{tt}(x,t), \quad u(0,t) = 0,\ u(L,t) = b\, exp(-d\, t)\, sin(w\, t),\ u(x,0) = u_x(x,0) = 0$$

Using discretization with the method of lines results in a weakly coupled system of equations:

$$k^2\, a\, d^2 u_i(t) / dt^2 = u_{i-1}(t) - 2\, u_i(t) + u_{i+1}(t),\ i = 1, ..., N-1;\ u_i(0) = du_i(0)/dt = 0$$

$$u_0(t) = u(0,t) = 0,\ u_N(t) = u(L,t) = b\, exp(-d\, t)\, sin(w\, t),\ L=10,\ a=2,\ b=1,\ d=0.2,\ w = 1,\ k = L/N$$

The task is to solve the system of equations with a discretisation $N = 800$ or more lines within the time horizon [0, 30] in a serial and in an appropriate parallel fashion. As result the lines at $x=9L/10$, $x=3L/4$, $x=L/2$, $x=L/4$ and $x=L/10$ should be stored for subsequent plotting.
Figure 5 shows results for some lines, figure 6 shows the weakly coupled strukture of parallel tasks, if M different processors calculate N/M subsequent lines.

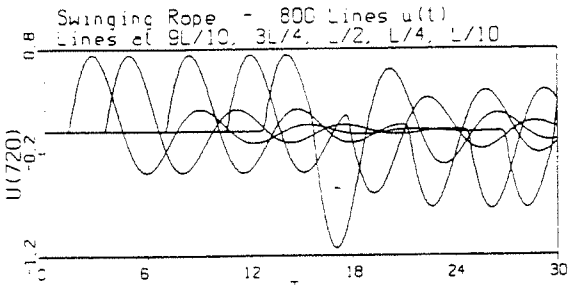


Fig.5: TE-3: Results for some lines

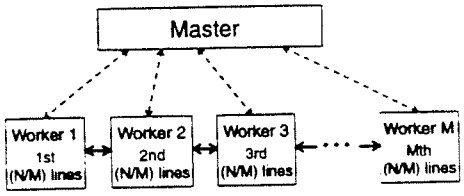


Fig 6: TE-3: Weakly coupled task structure

3. RESULTS OF THE PARALLEL COMPARISON

Up to now seven solutions have been sent in, briefly sketched in the following. Table 2 summarizes the results quantitatively.

CP-1: Workstation Cluster / PVM, FORTRAN, C. The first solution was published already in SNE 10 as a sample solution and was programmed directly in the programming languages FORTRAN and „C", using the message passing system „PVM". The programs were developed on a IBM RS6000-cluster (9 workstations) connected by a Token-Ring network and using PVM version 3.2.6. TE-1 resulted in almost linear speed-up, also TE-3 shows significantly speed-up (but only until to a certain number of processors); TE-2 gave negative results (typical for compiling systems).

CP-2: Parsytec Cluster / SLIM. The second solution (University of Glasgow) used the continuous system simulation tool SLIM (interpreter language) and a Parsytec Supercluster (Transputers working with the PARIX operating system). By now, only TE-1 could be provided (Monte Carlo simulation, master-slave approach, linear speed-up). The „parallel features" (sending messages from master to the slaves) had to be implemented „by hand" as the SLIM system did not provide communication between different models.

CP-3: Cogent XTM / mosis. The third solution again came from TU Vienna: The three tasks were implemented within the parallel simulation language **„mosis"** developed there. The hardware used was the 20-transputer system Cogent XTM with operating system QIX and communication system „Kernel Linda". **mosis** itself provides the the communication between the processors (simulation tasks) and can work with different operating/communication systems. (Linda, PVM, PC's etc.). The models in **mosis** are compiled to „C" and linked to the run time system.

CP-4: Heterogeneous MP- System / FSIMUL_P. The fourth solution from Ruhr-Universität Bochum shows an interesting heterogeneous multiprocessor system, consisting of UNIX-based workstations and PC-based workplaces (connected by means of a TCP/IP network), and of a digital signal processor (DSP TMS320C40) and of transputers T800 (plug-in cards for PC - bus). As software the simulation package FSIMUL_P, a supplement of the block oriented simulation language FSIMUL, was used. The simulations were performed on the fast processors of the system. The results were quite fast, also for test example 2 (non-compiling system).

CP-5: Cogent XTM / Linda, C. The fifth solution allows to compare parallelization on the level of a simulation language with parallelization in a programming language. This solution was performed on the 20-transputer system Cogent XTM with operating system QIX and communication system Kernel Linda (as in CP-3), but as software C and the Linda System was used (instead of the simulation systems mosis, see CP-3). This directly programmed solution is slightly better than the solution with a simulation system, but the implementation took much more time.

<table>
<tr><th>SOLUTION</th><th>MONTE-CARLO-STUDY (TE-1)</th><th>PREDATOR-PREY SYSTEM (TE-2)</th><th>PARTIAL DIFFERENTIAL EQUATION (TE-3)</th></tr>
<tr><td>Workstation Cluster / PVM with FORTAN, C)</td><td>P = 8 (9), S=2,
f = 4,9</td><td><table><tr><td>P=5 (6)</td><td>h</td><td>5h</td><td>10h</td><td>20h</td></tr><tr><td>S=2</td><td>0.05</td><td>0.21</td><td>0.39</td><td>0.77</td></tr></table></td><td><table><tr><td>P = 8 (9)</td><td>h</td><td>2h</td><td>4h</td><td>8h</td></tr><tr><td>N=600,S=125</td><td>0.59</td><td>1.10</td><td>1.9</td><td>3.1</td></tr><tr><td>N=800,S=200</td><td>0.72</td><td>1.37</td><td>2.05</td><td>3.82</td></tr><tr><td>N=1000,S=250</td><td>0.93</td><td>2.05</td><td>3.10</td><td>4.30</td></tr></table></td></tr>
<tr><td>Pasytec Cluster / SLIM</td><td><table><tr><td>P</td><td>2 (3)</td><td>8 (9)</td><td>16 (17)</td></tr><tr><td>f</td><td>1.99</td><td>7.71</td><td>14.57</td></tr></table></td><td></td><td></td></tr>
<tr><td>Cogent XTM / mosis</td><td>P = 8 (9), S=2,
f = 4,4 (6,2)</td><td><table><tr><td>P=5 (6)</td><td>h</td><td>5h</td><td>10h</td><td>20h</td></tr><tr><td>S=2</td><td>0.06</td><td>0.19</td><td>0.30</td><td>0,61</td></tr></table></td><td><table><tr><td>P = 8 (9)</td><td>h</td><td>2h</td><td>4h</td><td>8h</td></tr><tr><td>N=800,S=200</td><td>4.33</td><td>4.56</td><td>5.64</td><td>6.04</td></tr></table></td></tr>
<tr><td>Heterogenous MP-System / FSIMUL_P</td><td>P = 4 (5), S = 2
f = 3,67</td><td><table><tr><td>P</td><td>S</td><td>h</td><td>2h</td><td>5h</td><td>10h</td></tr><tr><td>5 (6)</td><td>2</td><td>0.38</td><td>0.38</td><td>0.83</td><td>1.32</td></tr><tr><td>3 (4)</td><td>2 / 4</td><td>0.14</td><td>0.27</td><td>0.62</td><td>1.06</td></tr></table></td><td>P = 4 (5), N = 800
f = 3,85</td></tr>
<tr><td>Cogent XTM / Linda („C“)</td><td>P = 8 (9), S=2,
f = 7,8</td><td><table><tr><td>P=5 (6)</td><td>h</td><td>5h</td><td>10h</td><td>20h</td></tr><tr><td>S=2</td><td>0.08</td><td>0.29</td><td>0.60</td><td>1.20</td></tr></table></td><td><table><tr><td>P = 8 (9)</td><td>h</td><td>2h</td><td>4h</td><td>8h</td></tr><tr><td>N=600,S=125</td><td>6.78</td><td>7.12</td><td>7.54</td><td>7.61</td></tr><tr><td>N=800,S=200</td><td>6.88</td><td>7.21</td><td>7.62</td><td>7.70</td></tr></table></td></tr>
<tr><td>Cogent XTM / SIMUL_R PARALLEL</td><td><table><tr><td>P</td><td>2 (3)</td><td>8 (9)</td><td>16 (17)</td></tr><tr><td>f</td><td>1.93</td><td>3.57</td><td>9.99</td></tr></table></td><td>P = 5 (6), S = 2
f = 0,04</td><td><table><tr><td>P</td><td>1 (2)</td><td>2 (3)</td><td>4 (5)</td><td>8 (9)</td></tr><tr><td>N=800,S=200</td><td>1</td><td>1.79</td><td>2.75</td><td>2.35</td></tr></table></td></tr>
<tr><td>Workstation Cluster / MATLAB - PSI</td><td><table><tr><td>P</td><td>2 (3)</td><td>5 (6)</td><td>10 (11)</td></tr><tr><td>f</td><td>2.00</td><td>4.99</td><td>9.92</td></tr></table></td><td><table><tr><td>P=5 (6)</td><td>h</td><td>2h</td><td>5h</td><td>10h</td></tr><tr><td>S=2</td><td>0.70</td><td>0.98</td><td>1.90</td><td>2.77</td></tr></table></td><td><table><tr><td>P = 8 (9)</td><td>h</td><td>2h</td><td>4h</td><td>8h</td></tr><tr><td>N=800,S=200</td><td>5.01</td><td>5.71</td><td>6.24</td><td>6.54</td></tr></table></td></tr>
</table>

Table 2: Results of seven solutions for the Parallel Comparisons (P - number of processors, S - number of states on each processor, h - stepsize, N - discretization of PDE, f - speed-up factor)

CP-6: Cogent XTM / SIMUL _R _PARALLEL. The sixth solution again was performed on the 20-transputer system Cogent XTM with operating system QIX and communication system Kernel Linda (as CP-3 and CP-5). As software the C-based CSSL-type simulation language SIMUL_R (with parallel extensions) was used. The results show the typical behaviour of compiling languages: TE-1 allows linear speed-up with the number of processors, TE-3 shows significant speed-up until to a certain number of processors (then decrease of speed-up), TE-2 gives negative results, because of the too fine granularity of the models and because of the very fast serial simulation of the compiled models.

CP-7: Workstation Cluster / MATLAB - PSI. The seventh solution shows that parallelization is also possible within classical and commercial simulation tools. The authors developed and used a C++ class library for transport independent interprocess communication between UNIX workstations in order to implement the parallelized test examples in MATLAB. The simulations were performed on a cluster of SUN classic workstations, connected via Ethernet, all running MATLAB. TE-1 and TE-3 yielded similar speed-up results like the other solutions, but also TE-2 showed a speed-up: the reason is the relatively slow interpreting processing time in MATLAB.

It has to be noted that the three test examples are are structurally different, they show hierarchical structures, weakly and strongly coupled structures, and different granularity of substructures. In the following the results of the solutions sent in are compared by means of qualitative criteria, by means of classifying influence factors.

In general, the following factors influence the success of a parallelization:

- processor hardware (F1)
- communication hardware (F2)
- communication software (F3)
- programming language (F5)
- modelling software (F5)
- experimentation software (F6)

Table 3 tries to qualify these factors in terms of „**s**" - slow, „**a**" - average, and „**f**"-fast, for the seven solutions sent in.

SOLUTION	F1	F2	F3	F4	F5	F6
Workstation Cluster / PVM, FORTRAN, C	a	s	a	f	-	-
Parsytec Cluster / SLIM	s	f	s	-	s	a
Cogent XTM / mosis	s	f	s	-	f	f
Heterogeneous MP- System / FSIMUL_P	a	s	s	-	a	a
Cogent XTM / Linda, C	s	f	a	f	-	-
Cogent XTM / SIMUL _R _PARALLEL	s	f	s	-	f	f
Workstation Cluster / MATLAB - PSI	a	s	a		s	s

Table3: Qualification of parallelization factors of seven solutions

4. REMARK

Together with the solutions to the comparisons, or independently, also software demos and test versions of simulation software and related tools were sent to the editors. These demos and test versions are available from the software server

`<URL: ftp://simserv.tuwien.ac.at>`

which is run by the ***ARGE Simulation News* (ARGESIM).** ARGESIM is a non-profit working group at the Technical University of Vienna. The group offers the infra-structure for the editorial office for **SNE**, for simulation courses, and publishes the **ARGESIM Reports**. etc.
ARGESIM also runs the **EUROSIM** WWW - server

`<URL: http://eurosim.tuwien.ac.at>`

This server offers information on EUROSIM, ARGESIM and their activities (conferences, development, projects, etc). Recent EUROSIM society news may be found there (from **SNE**), and information about the Comparisons and their solutions.

Real-Time Simulation and Control of an Uncertain Railway Bogie

T. Gajdár [a] and G. Kolumbán[b]

[a] Department of Railway Vehicles, Faculty of Transportation Engineering,
[b] Department of Measurements and Automation, Faculty of Electrical Engineering,
TU of Budapest, H-1521

ABSTRACT

Railway vehicles using conventional forms of wheelsets have limited performance, but the application of active controllers to the wheelset guidance systems offers the possibility of improved performance and stability. The authors investigated the usefulness of H_∞ active control design applied to the wheelset guidance systems of a railway bogie having parameter uncertainties. The obtained results from the linear computations are applied on a fully nonlinear bogie system. After the computation took place the performance and stability results of conventional and controlled bogie systems are compared.

INTRODUCTION

In the achievement of high speed railroad traffic the loss of stability, hunting motion and inadequate riding comfort are usual phenomena. To reduce passenger discomfort and to increase the safety aspects of railroad traffic many solutions are considered. One of the ways is the use of special tracks, can be seen at high speed lines in France and in Japan, although this solution is very costly and can only be done when new tracks are built. The other way to increase safety and ride comfort is to use new advanced solutions in the construction area of bogies, although this success is limited due to the already high utilization of technology. Although by the application of active or semi-active controllers in the vehicles [1,2,3,4,5,6] is promising. At the present paper the authors goal is to investigate the usefulness of active controllers supplied at the bogie in parallel to the longitudinal guidance of wheelsets. Concerning the safety and comfort needs, the parameter uncertainties play an important role, because these are highly influence the dynamic properties of vehicles. These uncertainties are in present due to production failures, inadequate maintenance and parameter changes during operation. In real systems such as railroad bogies, many nonlinearities exist, such as the nonlinear creep terms and geometrical nonlinearities, representing the real curvature of wheel and rail profile. These parameters can also be concerned as uncertain within an upper and lower limit, therefore its effect can be built into the control part.
The afore-mentioned phenomenon results in the need of robust control design, called as Robust Linear Quadratic Regulator design (RLQR). The control problem can be solved as a direct linear full or limited state-feedback problem, assuming that the required state variables are available directly. Under real operational conditions the vehicle speed isn't constant, because of the effect of traffic conditions, signalling equipment, central train direction, as it can be seen on Figure 1. The reason of the application of controllers is also the feasibility of practical utilization. Although the question remains, namely does the cost of installation cover the benefits gained ?
To answer this question one must think about the complexity of railroad traffic, such as

- interaction of train and track, energy consumption, impact on environment and maintenance costs.

Due to the afore-mentioned criteria the design specifications were

- to increase passenger comfort, which means reduced lateral acceleration, displacements, relative velocities of the wheelsets,
- to take into consideration the effect of parameter uncertainties, such as to incorporate the effect of contact geometry, and to assure the system against parameter deviations of the wheelset guidance parameters. This consideration results in the application of Robust Control Design.

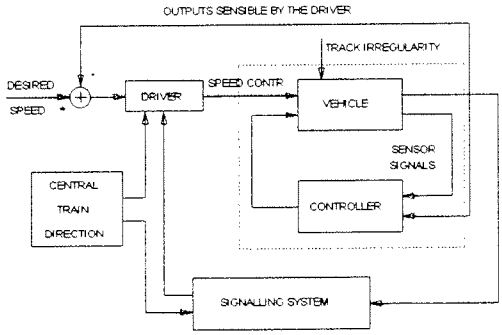


Figure 1. Complex model of vehicle-controller-driver and its environment

In order to make a comparison of the controlled and conventional systems numerical analysis is carried out. In the controlling phase the more practical limited state feedback method proposed by Venhovens is applied, aiming less sophisticated measurements of state variables. As the result of numerical calculations the system dynamic behaviour is evaluated on the basis of nonlinear methods, such as by investigating the time histories, phase plane plots, etc. In the calculations it was proven that the controlled bogie always gives better results in comparison to the conventional system.

2. EQUATIONS OF MOTION OF AN UNCERTAIN BOGIE MODEL

A simplified widely used 6 DOF bogie model (Figure 2) is used to design various control strategies proposed in this study. To model the wheel/rail contact mechanics Kalker's linear creep theorem is used. The generized equations of motion of a linear wheelset and bogie model without to going into details can be found in literature [4,6].

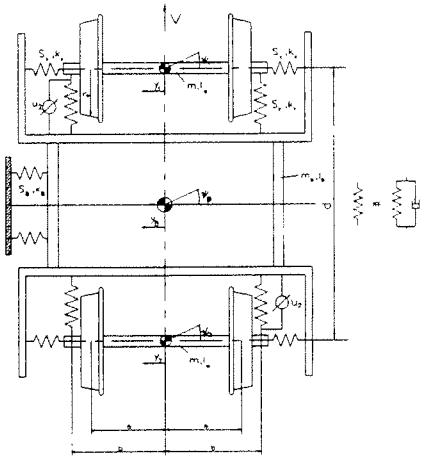


Figure 2. Linear railroad bogie model

The linear equations of motion of bogie in a general form expressed as follows (for details on matrices, see [6]):

$$M\ddot{x}(t) = -K\dot{x}(t) - Sx(t) + D_{\delta_1}\delta_1(t) + D_{\delta_2}\delta_2(t - \frac{d}{V}) + Cu_2, \quad (1)$$

The state vector, x, of the bogie model can be expressed as follows:

$$x^T = [y_1 \; y_2 \; \psi_1 \; \psi_2 \; y_B \; \psi_B \; \dot{y}_1 \; \dot{y}_2 \; \dot{\psi}_1 \; \dot{\psi}_2 \; \dot{y}_B \; \dot{\psi}_B].$$

Equation 1 can be rearranged into state space form, which is more suitable for control purposes:

$$\dot{x}(t) = Ax(t) + B_1u_1(t) + B_2u_2(t). \quad (2)$$

The measurable outputs of the bogie system are the lateral accelerations, lateral and yaw relative velocities and displacements of wheelsets, which can be expressed as follows:

$$y_1(t) = C_1x(t) + D_{11}u_1(t) + D_{12}u_2(t). \quad (3)$$

2.1. Modeling parameter uncertainties

The representation of uncertainties is based on the additive property of matrices, where the state matrix **A** in Equation 1, can be separated into the nominal and the uncertain parts as follows:

$$A = A_o + \sum_{i=1}^{n} q_iE_i, \quad (4)$$

Since the uncertain matrices in Equation 1 are all of rank 1, Equation 1 can be written as

$$A = A_o + L\Delta N^T, \quad (5)$$

where the matrices are given in [6]. Now the state space equation of the uncertain system can be written as:

$$\dot{x} = A_ox + L\Delta N^Tx + B_1u_1 + B_2u_2., \quad (6)$$

where B_1 and B_2 vectors are the disturbance and control vectors, respectively [4,5,6].
The goal of the controller design is to minimize the performance index and to make the controller robust against parameter uncertainties of bogie parameters.

The performance index of the robust control problem is given by:

$$J(u_1^*, d^*, u_2^*, \lambda, \gamma) = \lim_{T\to\infty} \frac{1}{T}\int_0^T [x^T(Q_o + \lambda NN^T)x + u_2^TR_ou_2 - \gamma^2u_1^Tu_1 - \lambda d^Td]dt \quad (7)$$

The optimal robust controller, which minimizes the performance index given by Equation 7 concerning worst case disturbance can be written as:

$$u_2(t, \gamma, \lambda) = -R_o^{-1}B_2^TX_{H_\infty/RLQR}(\gamma, \lambda)x(t), \quad (8)$$

where $X_{H_\infty/RLQR}$ is the positive semi-definite solution of the following ARE:

$$\begin{aligned} &X_{H_\infty/RLQR}A_o + A_o^TX_{H_\infty/RLQR} + (Q_o + \lambda NN^T) + \\ &+X_{H_\infty/RLQR}(\frac{1}{\gamma^2}B_1B_1^T + \frac{1}{\lambda}LL^T - B_2R_o^{-1}B_2^T)X_{H_\infty/RLQR} = 0. \end{aligned} \quad (9)$$

3. NONLINEARITIES OF BOGIE MODEL

In the upper part of paper the RLQR controller design procedure is tackled based on linear modeling. The effect of possible nonlinearities is reckoned with, by means of changing wheel conicity given by its minimal and maximal boundary values.
To model the nonlinear contact mechanics, a lot of consideration must be taken into account,

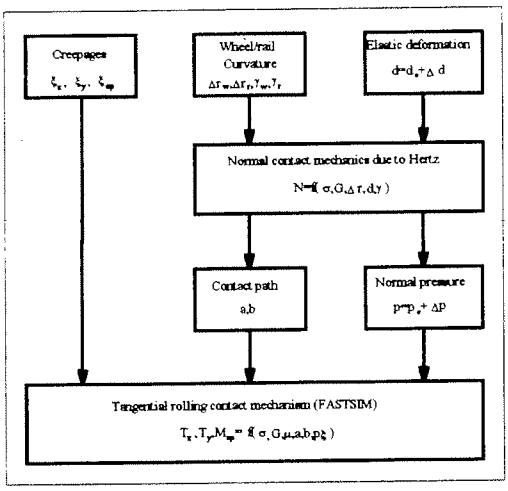


Figure 4. Nonlinear contact mechanics

such as the nonlinear creep saturation curves and wheel/rail contact geometry, see Figure 4. The sizes of contact ellipses are calculated according to the actual position of wheelsets, normal loads and wheel/rail geometry [8]. The creep forces and moment are defined as the function of creepages, contact path and maximal pressure between wheel and rail. To model the wheel profile the Hungarian K5 profile is selected, while to model the rail profile the UIC 54 profile is used.

3. METHOD OF INVESTIGATION

Based on numerical results obtained for a linear wheelset and bogie model [4,6] two feedback gains are obtained for the LQR and H_∞ / *RLQR* controllers, after the two iteration procedure for γ and λ (γ=50 and λ=850). Due to the rather complicated nature of state parameter measurement the controller was designed by only supposing the direct availability of the two DOF wheelsets state parameters. The 9-10 rows of obtained feedback gain matrix for the limited state feedback case by only taking into consideration the 2 DOF wheelset state parameters at V=60 m/s, thus:

$$K_{LQR} = \begin{bmatrix} -2.6751 & 0 & -7.4188 & 0 & 0 & 0 & 0.0048 & 0 & -0.1622 & 0 & 0 & 0 \\ 0 & -2.6751 & 0 & -7.4188 & 0 & 0 & 0 & 0.0048 & 0 & -0.1622 & 0 & 0 \end{bmatrix} \times 10^6,$$

$$K_{RLQR} = \begin{bmatrix} -3.4828 & 0 & -9.9226 & 0 & 0 & 0 & 0.0043 & 0 & -0.2243 & 0 & 0 & 0 \\ 0 & -3.4828 & 0 & -9.9226 & 0 & 0 & 0 & 0.0043 & 0 & -0.2243 & 0 & 0 \end{bmatrix} \times 10^6.$$

After the controlled bogie model based on linear analysis is set up the linear controller applied on a fully nonlinear model. The model contains kinematical and geometrical nonlinearities while the guidance parameters are linear. The parameter uncertainties are embodied as changing contact parameters. Other unmodelled uncertainties can be considered, such as the layer physics of contacting bodies, see Figure 4. After the numerical computations carried out

the evaluation of computed data can be done by means of PSD functions, time series, phase plane plots, etc. (see Figure 4).

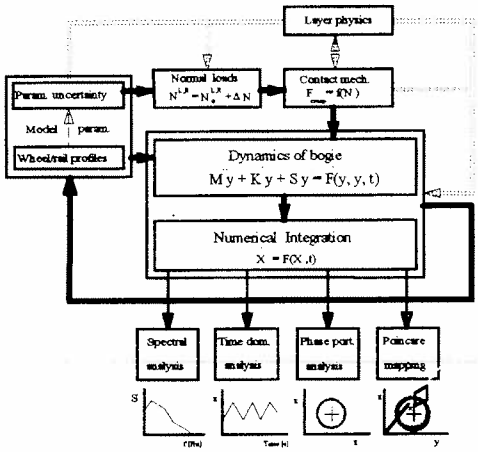


Figure 5. Method of dynamic analysis

4. NUMERICAL RESULTS AND DISCUSSION

As the aim to apply controllers was to suppress the vibrations and accelerations of wheelsets and bogie frame. Numerical simulations were carried out at V=190 and V=300 km/h-s for both conventional and controlled bogies. As it is feasible from Figure 6., where the lateral accelerations of front and rear wheelsets and the controlled front wheelset are shown, that the response of controlled wheelset is mush smaller and no wheel/rail contact occur.

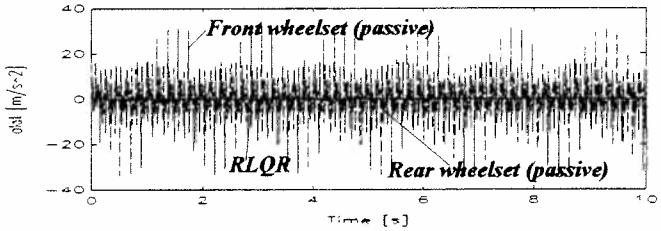


Figure 6. Lateral accelerations of the wheelsets of bogie

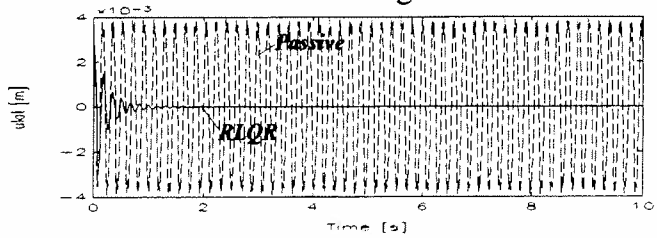


Figure 7. Lateral displacements of front wheelset of bogie

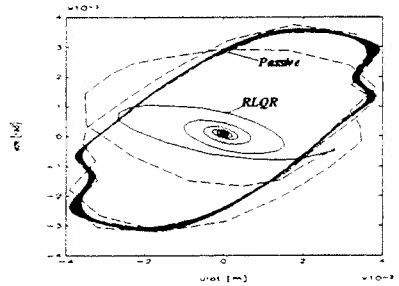


Figure 8. Phase plane plot of front wheelset at V=190 km/h

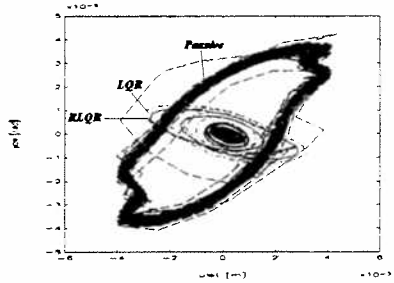


Figure 9. Phase plane plot of front wheelset at V=300 km/h

In Figure 7 the lateral displacement of the conventional and RLQR controlled front wheelsets are shown. In Figure 8 and 9 the same wheelset phase plane plots are shown at V=190 and 300 km/h. It is feasible from the Figures that in case of the conventional system hunting motion develops, which as the speed increases going to be a chaotic motion, while the controlled systems behave in a stable way approaching a stable focus or a small amplitude limit cycle.

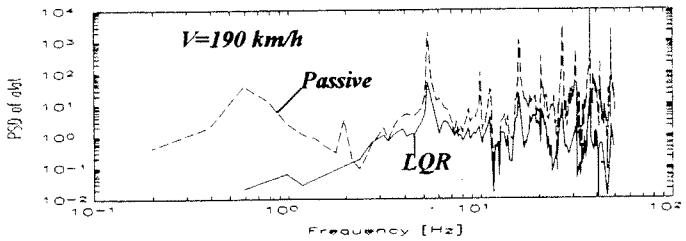


Figure 10. Lateral acceleration spectrum of first wheelset

5. CONCLUSION AND FUTURE WORK

Based on the LQR and RLQR control strategies applied to a linear bogie model the obtained feedback gains are used on a nonlinear bogie model. The optimal state-feedback controller was determined by minimizing the an appropriate performance index. The linear controller applied on a nonlinear bogie system gives satisfactory results in all velocity of travel, also in very high speeds such as 300 km/h. For both controllers the system remains stable, the amplitudes of motion are reduced and wheel/rail contact is avoided which is also feasible from the PSD functions. In the future the modeling must be extended for a whole vehicle model and the curving behaviour of vehicle with controllers must be also considered.

ACKNOWLEDGEMENT

Research was supported by the Hungarian OTKA Foundation under Grant no.: F016922.

REFERENCES

1. W.Kortüm: Introduction to system dynamics of ground vehicles, Proceedings of the third ICTS Seminar, Amalfi, Italy,pp.1-36,1986.
2. R.A.Williams: Active suspensions classical or optimal? Proceedings of 9th IAVSD Symposium, Linköping, Sweden, June 24-28, pp.607-620, 1985.
3. A.Shimura, K.Yoshida: H^{∞} Vibration Control of Active Suspension for High-Speed Train, 2nd Int. Conf. on Motion and Vibration Control, Yokohama, Aug.30-Sept.3, 1994.
4. T.Gajdár, L.Oroszváry: Active Wheelset Design in the Presence of Physical Parametric Uncertainties, Proc. 4th Mini Conf. on Vehicle System Dynamics, Budapest, Nov.7-9, 1994. (to appear).
5. J.Bokor, A.Keresztes, L.Palkovics, P.Várlaki: Design of Active Suspension System in the Presence of Physical Parameter Uncertainties, FISITA, 1994.
6. T.Gajdár, L.Oroszváry: Active H_{∞} Controller Design for Conventional Railway Bogie, Int. J. of Vehicle Design, (to appear).
7. Paul J.Th., Venhovens: Optimal Control of Vehicle Suspensions, Ph.D. Thesis, 1994.
8. A.Szabó, T.Gajdár, Gy.Sostarics, I.Zobory: On Numerical Simulation of the Wheel Wear Process under Specified Operational Conditions - Possibility of Wheelset Guidance Optimization, Contact Mechanics and Wear of Rail/Wheel Systems, 4th Int. Conf., Vancouver, 24-28 July, 1994, (to appear).
9. Ching, R.Y.,Safanov, M.G.: Robust Control TOOLBOX, User's Guide, 1992.

3D-FINITE ELEMENTS FOR THE ANALYSIS OF PROGRESSIVE FAILURE IN LAMINATED COMPOSITE BEAMS

R. Luciano*, R. Zinno**

*Università di Cassino - Via Zamosh, 43 - Cassino - Italy
** ICITE-CNR - Via Tiburtina, 770 - Rome - Italy

ABSTRACT

Many models have been proposed to predict the first occurence of failure, but few methods are available to analyze the subsequent behaviour of the structure in composite laminates. In this paper a micromechanical approach is used to model the progressive damage of composite beams. It is characterized by two steps: individuation of the cracks in the damaged material by considering a quadratic polynomial criterion of first failure, and evaluation of the actual stiffness matrix by using a self-consistent micromechanical model. In order to model the progressive failure inside the laminated composite structure, in this paper a finite element code is developed by using a three-dimensional layer-wise constant shear elements (3DLCS). The formulation of the proposed elements is presented and a validation of its numerical performance is provided by comparisons with experimental data. The software here developed seems to be a promising tool for simulation of failure of more complex structures by a simple model problem.

1. INTRODUCTION

Many experiments have been shown that the loss of stiffness in a composite material is mainly due to the presence of microcracks. Several authors used the self-consistent method to study the loss of stiffness for a brittle material with an anisotropic distribution of cracks [1 ÷ 3]. Laws and Brockenbrough [2] developed several expressions for the evaluation of the change in the compliance matrix of an orthotropic material in presence of different types of open cracks, in the special cases of the penny-shaped and the slit cracks. In this paper a method is presented in order to determine the orientation and the shape of the cracks in a layer of a laminated composite structure, and the expressions developed by Laws and Brockenbrough are used to compute the stiffness of the corresponding damaged material. The creation of a new crack is detected by using one of the classic failure criteria [4].

Further, this methodology is implemented in a numerical code where a three-dimensional layer-wise constant shear element (3DLCS) is used to obtain the stress state inside the composite material. This element, as shown in ref. [5], makes possible the analysis of a composite laminate overcoming the implementation difficulties linked with the 2D-LWCS theories, but retaining their precise stress calculation and, as the conventional 3D-continuum elements, gives a very simple interpretation of the degrees of freedom (dof) and the stress resultants. In particular, the incompressibility constraint makes the element capable to overcome the problem of the ill-conditioning shown by Ahmad [6].

When the structure is characterized by the presence of several layers, if we choose to use one layer of 3DLCS elements to model each lamina, the number of dof of the entire structure will increase. Thus, in this work, it is proposed a method based on the concept of the cluster of laminae to model many layers, along the thickness direction, by only one layer of 3DLCS elements. Finally, some numerical comparisons with experimental data obtained by Greif and Chapon [7] are developed to validate the proposed method.

2. MICROMECHANICAL MODEL

The reduction of the stiffness in each point P of an orthotropic material which characterizes each layer of the laminated composite structure is evaluated in function of the density, the shape and the orientation of the cracks in the volume around P. For a cracked orthotropic material, a closed form expression for the determination of the compliance matrix have been proposed by Laws and Brockenbrough [2], by using the *Self-consistent* method. In particular if we denote by Γ^i the compliance change of an orthotropic solid with initial compliance S^0 due to a unit density distribution of cracks of type i, we can write the final compliance of the damaged material with many similar cracks in the following way:

$$[S] = [S^0] + f_i[\Gamma^i(S)]; \tag{1}$$

where f_i is the dimensionless crack density parameter. Since Γ^i is evaluated in function of the geometry of the crack and the final compliance of the damaged material, the eqn. (1) represents a nonlinear equation in S. As shown in [2], it is possible to approximate the computations to a single iteration (e.g.: Taylor method), so the expression (1) becomes formula.

In order to detect, in each point, the presence of a crack, a quadratic polynomial criterion can be used. It can be written in function of the stress vector (σ_i, $i = 1,...6$) as follows:

$$F_1\sigma_1 + F_2\sigma_2 + 2F_{12}\sigma_1\sigma_2 + F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + F_{44}\sigma_4^2 + F_{55}\sigma_5^2 + F_{66}\sigma_6^2 = f(\sigma); \tag{2}$$

where if $f(\sigma) < 1$ there is no failure and if $f(\sigma) \geq 1$ there is failure. In eqn. (2) the stress components must be written with reference to the material orthogonal directions (1,2,3), (1) being the fiber direction and (3) the thickness direction. In this expression F_i, F_{ij} and F_{ijk} are the components of the 2nd, 4th and 6th rank strength tensors, respectively. In eqn. (2) only the 2nd and 4th rank strength tensor components (F_i and F_{ij}) were taken into account. The strength terms associated with the shear stresses σ_4 ($=\sigma_{23}$), σ_5 ($=\sigma_{13}$) and σ_6 ($=\sigma_{12}$) F_4, F_5 and F_6 are taken to be equal to zero, since the shear strengths are the same for positive and negative shear stresses. It is also assumed that there is no interaction between shear stresses and normal stresses, thus F_{16}, F_{26}, etc. become zeros. The expressions of the coefficients F_i and F_{ij} for the Tsai-Wu and the maximum stress failure criteria used in this work, are functions of the composite material strengths (X_T, X_C, Y_T, Y_C, R, S and T). They can be found in [4].

In the following we will consider only the following types of cracks: (a) cracks in the direction perpendicular to the fibers, and (b) cracks parallel to the fibers and growing in the matrix. From experimental results the shape of the first kind of cracks is penny-shaped, while in the other case the cracks have slit-aligned shape. In order to determine the orientation of the cracks, the principal tensile direction of the stress field in any point in which is detected the presence of failure will be considered. In particular, if the principal tensile direction mismatches less than 10^0 from the fiber direction, the crack will grow parallel to the fibers, otherwise it will be considered orthogonal to them. For each uncracked orthotropic ply, costituent the composite laminate structure, the Cauchy stress tensor $\{\sigma\}$ and the infinitesimal strain tensor $\{\varepsilon\}$, written with respect to the material axes (1,2,3), are related by the compliance matrix [S] as follows:

$$\{\varepsilon\} = [\mathbf{S}]\{\sigma\}; \tag{3}$$

where the components of [S] can be written, as usual, in terms of the engineering constants.

Then, from eqn. (3) the Total Potential Energy of an uncracked body can be written, by neglecting the force per unit volume, as:

$$\Pi = \frac{1}{2}\int_V \{\sigma\}\{\varepsilon\} - \int_{S_b} \{t\}\{u\}dS_b = -\frac{1}{2}V\{\overline{\sigma}\}[\mathbf{S}]\{\overline{\sigma}\}; \tag{4}$$

where $\{t\}$ and $\{u\}$ are the forces and the displacements on the body surface S_b.

On the other side, the total potential energy in a cracked solid is:

$$\Pi_c = -\frac{1}{2}V\{\overline{\sigma}\}[\mathbf{S}(\overline{\sigma})]\{\overline{\sigma}\}; \tag{5}$$

where $[\mathbf{S}(\overline{\sigma})]$ is the effective final compliance tensor of the cracked solid and it depends on the value of the actual stress field $\{\overline{\sigma}\}$. Thus the energy released by the system of cracks is the difference between the Total Potential Energies of the uncracked and the cracked body. The final compliance and the actual energy released E_r, can be evaluated from the expression:

$$\Pi_c - \Pi = E_r = \frac{1}{2}f_i\{\overline{\sigma}\}\Gamma^i\{\overline{\sigma}\}\ . \tag{6}$$

In the case of slit cracks, the effective compliance tensor of the cracked body is [2]:

$$[\mathbf{S}(\overline{\sigma})] = [\mathbf{S}] + \pi\beta[\Gamma^S] = [\mathbf{S}] + f_s[\Gamma^S]; \tag{7}$$

where β is the dimensionless crack density parameter, the superscript S indicates slit cracks, and the non-zero components of $[\Gamma^S]$ are:

$$\Gamma^S_{22} = \frac{S_{22}S_{11} - S_{21}^2}{S_{11}} \cdot (\alpha_1^{1/2} + \alpha_2^{1/2}); \qquad \Gamma^S_{44} = (S_{44}S_{55})^{1/2}; \tag{8a,b}$$

$$\Gamma^S_{66} = \frac{(S_{22}S_{11} - S_{21}^2)^{1/2}(S_{33}S_{11} - S_{13}^2)^{1/2}}{S_{11}} \cdot (\alpha_1^{1/2} + \alpha_2^{1/2}); \tag{8c}$$

where α_1 and α_2 are the roots of the following equation:

$$(S_{22}S_{11} - S_{21}^2)\alpha^2 - [S_{11}S_{66} + 2(S_{23}S_{11} - S_{13}S_{21})]\alpha + S_{11}S_{33} - S_{13}^2 = 0. \tag{9}$$

In the case of penny-shaped cracks, the effective compliance tensor of the cracked body becomes [2]:

$$[\mathbf{S}(\overline{\sigma})] = [\mathbf{S}] + \frac{4}{3}\pi\delta[\Gamma^P] = [\mathbf{S}] + f_p[\Gamma^P]; \tag{10}$$

where the superscript P indicates penny-shaped cracks and δ is the dimensionless crack density parameter. The non-zero components of $[\Gamma^P]$ can be obtained by the following relations:

$$\Gamma^P_{11} = \frac{2\gamma_1\gamma_2(\gamma_1+\gamma_2)(S_{33} - S_{32}^2)}{\pi S_{33}}; \qquad \Gamma^P_{44} = \frac{4(\gamma_1+\gamma_2)(S_{33}^2 - S_{32}^2)(2S_{44})^{1/2}}{\pi[S_{33}(2S_{44})^{1/2} + (\gamma_1\gamma_2)(S_{33}+S_{23})(S_{33}-S_{32})^{1/2}]}; \tag{11a,b}$$

where $\Gamma^P_{44} = \Gamma^P_{55}$, while γ_1^2 and γ_2^2 are the roots of

$$(S_{33}^2 - S_{32}^2)x^2 - [S_{33}S_{44} + 2S_{13}(S_{33} - S_{32})]x + S_{11}S_{33} - S_{13}^2 = 0. \tag{12}$$

It is worth to note that, in eqns. (7) $\div$ (12), the matrix [S] is the initial compliance matrix in according to the Taylor method. Since the problem is highly non linear, a suitable iterative finite element code will be developed to reach the solution.

3. FINITE ELEMENT MODEL AND SOLUTION PROCEDURE

Each layer, or cluster of laminae, is discretized by 3-Dimensional Layer-wise Constant Shear (3DLCS) elements. In this element the displacements and the coordinates referred to the (x, y, z) global axes are interpolated as follows:

$$\{\mathbf{u}\} = [\mathbf{N}]\{\mathbf{u}_i^e\}; \qquad \{\mathbf{x}\} = [\mathbf{N}]\{\mathbf{x}_i^e\}; \tag{13}$$

where $[\mathbf{N}]$ is the interpolation functions matrix, and $\{\mathbf{u}_i^e\}$ and $\{\mathbf{x}_i^e\}$ are the nodal displacements and coordinates vectors of each element. Quadratic interpolation functions are used for both (x, y) and (u, v), while linear variation is adopted in the thickness direction (see Fig. 1). Therefore the

element has 18 nodes, nine nodes with 3 dof (u, v, w) and the remaining with 2 dof (u, v). This is due to the incompressibility condition made along the thickness coordinate.

Then, the equilibrium of the structure can be obtained through the determination of the stationary point of the total potential energy of the uncracked (eqn. 4) or the cracked (eqn. 5) body. By discretizing these equilibrium equations, we can obtain:

$$\int_V [\mathbf{B}]^T \{\sigma\} dV = \int_{S_b} [\mathbf{N}]^T \{t\} dS_b; \qquad \{\varepsilon\} = [\mathbf{B}]\{\delta\}; \qquad (14a,b)$$

where $[\mathbf{B}]$ is the linear strain-displacement matrix which connects the strains $\{\varepsilon\}$ to the nodal displacement vector $\{\delta\}$, which is the collection of the nodal $\{\delta_i\}$. Then the stress field $\{\sigma\}$ can be evaluated from the strain field by using the inverse of the constitutive equation (3) for the uncracked body and eqns. (7) and (10) for the body damaged by slit-aligned cracks or penny-shaped cracks, respectively. This produces:

$$\{\overline{\sigma}_{123}\} = [\mathbf{C}_{123}(\overline{\sigma})]\{\varepsilon_{123}\}; \qquad \{\overline{\sigma}_{xyz}\} = [\mathbf{C}_{xyz}(\overline{\sigma})]\{\varepsilon_{xyz}\}; \qquad (15a,b)$$

where the passage from eqn. (15a), expressed in material coordinate, to eqn. (15b), expressed in global coordinate, it is easily achieved by using a suitable rotation matrix [5].

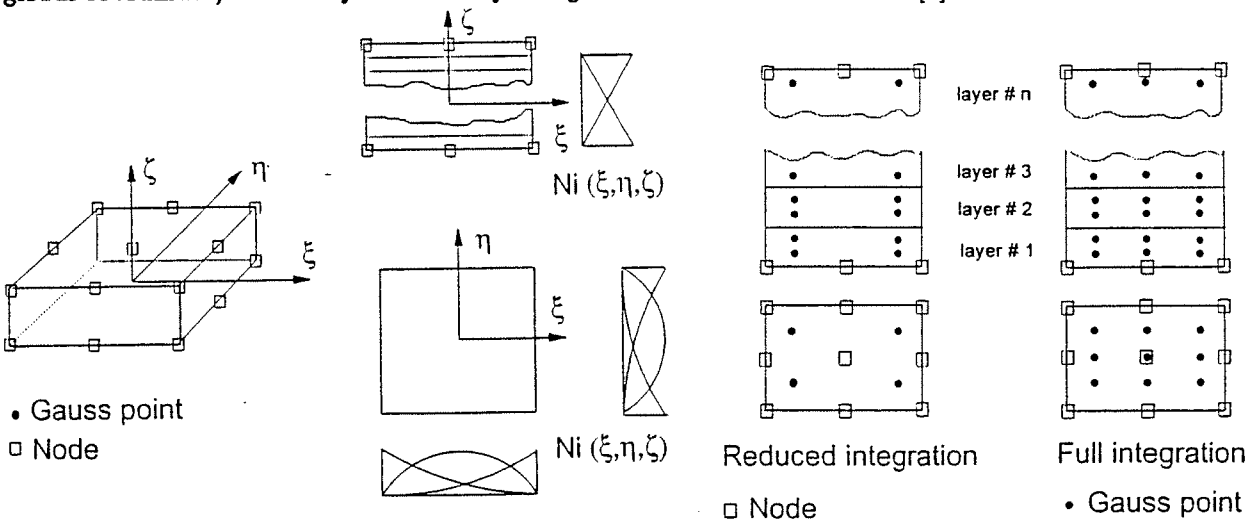


Fig. 1 - 3DLCS element and position of the Gauss points in a cluster of laminae.

From eqn. (14a) and using the constitutive equation (15b), it is possible to obtain the standard element stiffness matrix:

$$[\mathbf{K}^e] = \int_{V_e} [\mathbf{B}]^T [C_{xyz}(\overline{\sigma})][\mathbf{B}] dV_e. \qquad (16)$$

By using one element for each layer (or cluster of laminae) we will omit the statement that normals remain straigth after deformation reproducing the situation of the layer-wise models.

Further, the use of the cluster of laminae reduces significantly the number of the total dof and, if an intelligent choice of the group of layers inside the clusters is made (for example layers with the same fiber orientation), the precision of the stress calculation is not significantly reduced. As shown in Fig. 1, taking into account that through the thickness direction the interpolation functions are linear, two Gauss points for each layer are used in the thickness direction. In each layer the position of the two Gauss points is obtained using the well known two-points Gauss rule and the Gaussian weight is obtained by taking into account the dimension of each layer with respect to the other laminae. Because of the incompressibility, two nodes aligned through the thickness have the same w and we choose the master nodes on the top or on the bottom surface of the element. For this reason, it is possible to divide the nodal displacement vector into two parts. The first $\{\delta^1\}$ collects the independent dof (u and v of all the nodes and w of the master nodes), while the second $\{\delta^2\}$ collects the remaining dependent nodal displacements. The condition of incompressibility through the thickness can be introduced as a constraint equation by using a suitable matrix $[\mathbf{A}]$, and rewriting the element stiffness matrix and the force vector as follows [5]:

$$\{\delta\} = \begin{Bmatrix} \{\delta^1\} \\ \{\delta^2\} \end{Bmatrix} = [\mathbf{A}]\{\delta^1\}; \quad [\tilde{\mathbf{K}}^e(\bar{\sigma})] = [\mathbf{A}]^T[\overline{\mathbf{K}}^e(\bar{\sigma})][\mathbf{A}]; \quad \{\tilde{F}^e(\bar{\sigma})\} = [\mathbf{A}]^T\{\overline{F}^e(\bar{\sigma})\}; \quad \text{(17a, b, c)}$$

where $[\overline{\mathbf{K}}^e(\bar{\sigma})]$ and $\{\overline{F}^e(\bar{\sigma})\}$ are obtained from the $[\mathbf{K}^e(\bar{\sigma})]$ and $\{F^e(\bar{\sigma})\}$ moving opportunely their elements. If more than one layer is present in the laminate, another condensation procedure must be done for each vertical connecting all the w-displacements to only one master node.

The nonlinearity, produced by the dependence of the compliance tensor on the stress field, is taken into account by adopting the following updated direct method:

1. increase the applied loads or displacements of an increment $\{\delta F\}^n$ or $\{\delta u\}^n$;
2. determine the actual displacement field at the given load or displacement level. In particular the compliance matrix is calculated by taking into account if the failure is occurred and the shape of the crack in each Gauss point;
3. compute the strains and the stresses at each Gauss point and transfom the stresses with respect to the material coordinate system of reference;
4. evaluate the Tsai-Wu failure polynomial in each Gauss point. If the failure is occurred, calculate the maximum tensile principal stress and its direction with respect to the material coordinate;
5. if in every Gauss point there is no failure or if the difference between the last and the previous solution is less than a fixed tollerance (here $1 \cdot 10^{-6}$) go to step 1, otherwise go to step 2.

It is necessary to specify that when a Gauss point experiences the failure it remains cracked for the following steps and will be skipped in the failure polynomial evaluation.

4. NUMERICAL RESULTS AND EXPERIMENTAL VALIDATION

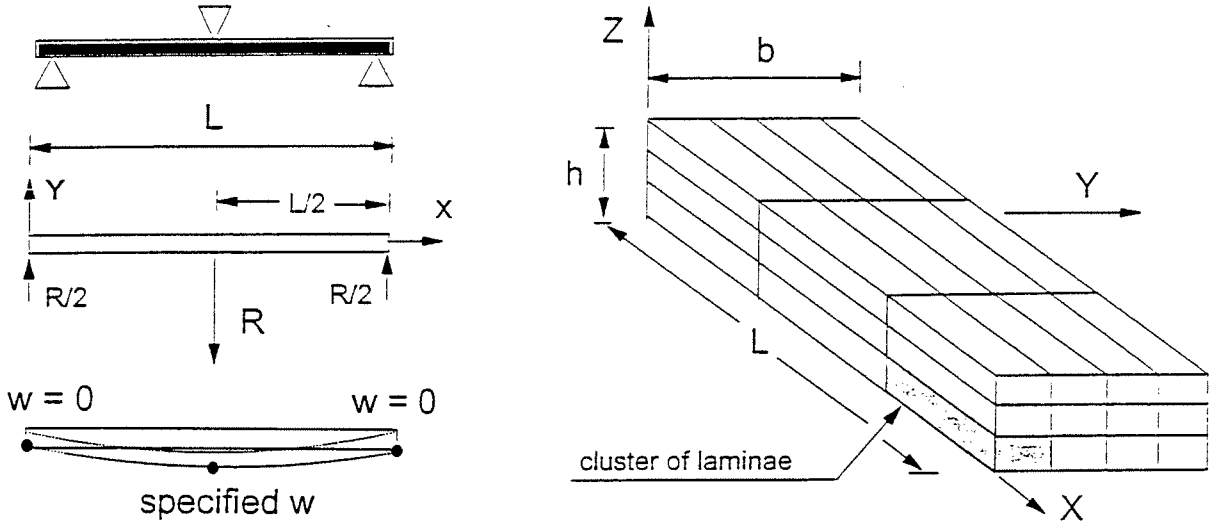


Fig. 2: Three-point bending scheme and typical 3D-finite element mesh

In this section some comparisons with experimental data available in literature are presented. In particular, specimens subjected to three-point bending were analyzed and the results are compared to the experimental data furnished in [7]. The beams are composed by layers having the same thickness and made in AS4/3502 Graphite-Epoxy pre-preg tape with the following mechanical parameters:

$E_1 = 15.0\ Msi$; $E_2 = E_3 = 1.1\ Msi$; $\nu_{12} = \nu_{13} = \nu_{23} = 0.30$; $G_{12} = G_{13} = 0.5723\ Msi$;
$G_{23} = 0.42307\ Msi$; $X_T = 0.27\ Msi$; $X_C = 0.215\ Msi$; $Y_T = 0.0075\ Msi$;
$Y_C = 0.03\ Msi$; $R = S = T = 0.0094\ Msi$.

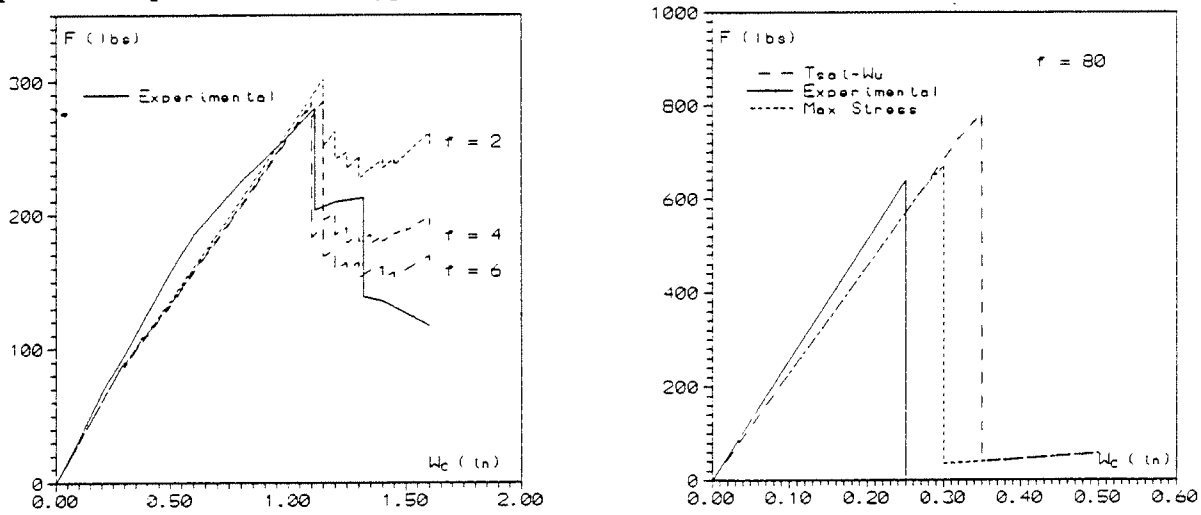
First a specimen with span of 6 inches, width of 1.01 inches and total thickness of the laminate equal to 0.179 inches, with a $[\ 90_8/0_8\ ]_S$ layup was analyzed. The load configuration and the boundary condition adopted are schematized in Fig. 2. In particular the beam was discretized by a 5x2x4 3DLCS elements. The Tsai-Wu and the maximum stress failure criteria were adopted.

The results show that the values of the crack density parameters do not influence the numerical estimate of the ultimate failure, but they become very important in the post-ultimate failure analysis.

In the case considered for the computations, the parameters $f = f_s = f_p = 4$ seems to be the best choice in order to fit the experimental data. Further, it is possible to note that the qualitative post-ultimate failure behaviour is not catastrophic for any value of f, because when the external 90^0 layers are broken the internal 0^0 layers continue to carry load.

The opposite conclusion can be reached by observing the results for a beam, subjected to a three-point bending, characterized by a lamination sequence of $[0_8 \ / \ 90_8]_S$ with span of 5 inches, width of 0.95 inches and total thickness of 0.181 inches. This example is discretized by the same finite element mesh of the previous case and subjected to the same boundary conditions.

From the previous example, it is possible to conclude that the proposed method appears stable in the determination of the ultimate failure which is very important in the design of a laminated composite structure. Moreover, the method is able to define the qualitative behaviour of the structure after the occurence of the ultimate failure. Thus the proposed progressive failure analysis makes possible the prediction of the type of the global failure as ductile or brittle.



a) case of $[90_8/0_8]_S$ *b) case of* $[0_8/90_8]_S$

Figg. 8a, b - Comparisons between numerical and experimental data for laminated composite beams

Acknowledgements

The authors wish to thank Prof. Franco Maceri for the valuable comments on this paper. The financial supports of the Italian National Research Council (CNR) and of the Italian Ministry of University and Researches (MURST) are gratefully acknowledged.

REFERENCES

[1] N. Laws, G.J. Dvorak and M. Hejazi - *Stiffness changes in unidirectional composites caused by crack systems* - Mech. Mater., 2, 123-137, 1983.

[2] N. Laws and J.R. Brockenbrough - *The effect of micro-crack system on the loss of stiffness of brittle solids* - Int. J. Sol. Struct., 23, 9, 1247-1268, 1987.

[3] N. Laws and G.J. Dvorak - *The effect of fiber breaks and aligned penny-shaped cracks on the stiffness and energy release rates in unidirectional composites* - Int. J. Sol. and Struct., 23, 1269-1283, 1987.

[4] D. Bruno, G. Spadea and R. Zinno - *First-ply failure of laminated composite plates* - Theor. Appl. Fract. Mech., 19(1), 29-48, 1993.

[5] R. Zinno and E.J. Barbero - *A three-dimensional layer-wise constant shear element for general anisotropic shell-type structures* - Int. J. Num. Meth. Engng., 37, 2445-2470, 1994.

[6] S. Ahmad, B. M. Irons and O.C. Zienkiewicz - *Analysis of thick and thin shell structures by curved finite elements* - Int. J. Num. Meth. Engng., 2, 419-451, 1970.

[7] R. Greif and E. Chapon - *Investigation of successive failure modes in graphite/epoxy laminated composite beams* - J. of Reinf. Plastics and Comp., 12, 1993.

FINITE ELEMENT ANALYSIS OF FAILURE IN LAMINATES

D. Bruno *, G. Porco *, R. Zinno **

* Department of Structural Engineering - University of Calabria - Cosenza - Italy
** ICITE-CNR - Via Tiburtina, 770 - Rome - Italy

ABSTRACT

It is important to accurately predict the strength and load carrying capacity of a composite material, which are typically directionally dependent, for design. Reliable application of such structures requires a knowledge of their stress-strain and failure behaviour. In this work an analysis of failure in laminates is developed, in particular the first occurrence of failure in a layer (first-ply failure) is taken into account for laminates. Moreover, the delamination failure process produced by the buckling of a debonded area is studied. Finally, some mechanical results are obtained for different laminate thicknesses and stacking sequences, which illustrate the influence of geometric and mechanical characteristics on failure behaviour and on delamination.

1. INTRODUCTION

Interlayer slip is one of the most common indices of degradation in composite laminated structures. Thus it is important to improve the theories available to obtain a better evaluation of the stresses, especially those of interlayer connection. The range of applicability of the classic plate theory has been well established by Pagano [1]. In particular, Pagano has shown the importance of incorporating the effect of transverse shear deformations in order to accurately estimate plate lateral deflections. Taking into account these considerations higher-order and layer-wise theories were proposed, but their implementation is not simple, because they imply a large number of degrees of freedom [2]. Improvements in the prediction of the stresses in laminated plates are also made possible by assuming a layer-wise zig-zag variation of the inplane displacements [3,4].

Failure criteria for composite materials are more difficult to postulate with respect to the analogous for the isotropic material, because they are more involved in terms of structural and material complexity. In 1967 Hoffman [5] and in 1971 Tsai and Wu [6] proposed two different theories, the first for brittle materials and the second appropriate to general anisotropic materials. Another aspect of failure is the delamination phenomenon, related to the presence of interlaminar defects.

In this work an analysis of failure in laminates is developed by using a first-order shear deformation plate model in the finite element development of a zig-zag layer-wise model of a multilayered structure. Moreover, delamination growth in laminates is analyzed taking into account fracture modes involved in the delamination failure process. Numerical results of delamination behaviour are also given, particularly regarding the maximum load carrying capacity of laminates loaded in compression which exhibit delamination buckling of layers.

2. STRESSES IN COMPOSITE PLATES

2.1. Plate theory and constitutive relationships

A laminated plate is composed with "n" orthotropic layers, the total thickness is "h" and the global system of reference (x, y, z) lies on the top surface with the z-axis directed in the thickness direction, positive downward and clockwise with x and y. In each layer lies a material system of

reference (1,2,3), where direction 1 is parallel to the fibres and direction 3 is coincident with the z-axis. The angle between the directions 1 and x will define the position of the fibres.

The displacement field is considered as follow [4]:

$$u = u^0 + z(\gamma_x - w^0_{,x}) + \sum_{k=1}^{n-1} \psi_k(z - z_k)Y(z - z_k);$$

$$v = v^0 + z(\gamma_y - w^0_{,y}) + \sum_{k=1}^{n-1} \chi_k(z - z_k)Y(z - z_k); \qquad w = w^0; \tag{1}$$

where: u, v, w are the displacements along the x, y and z axes, respectively;
u^0, v^0, w^0 are the displacements along the x, y and z directions, respectively, on the top surface Ω;
γ_x, γ_y are the shear rotations in the (x,z) and (y,z) planes of line normal to Ω;
z_k is the z-coordinate at the interface between the k-th and the $(k+1)$-th layers;
$Y(z - z_k)$ is the Heaviside unit function (0 for $z < z_k$ and 1 for $z \geq z_k$);
$\psi_k(x,y)$, $\chi_k(x,y)$ are functions determined in the following by satisfying the contact conditions on the transverse shearing stresses at the k-th interface between two generic layers:

i. Geometric continuity: $u(z_k^-) = u(z_k^+); \quad v(z_k^-) = v(z_k^+); \quad w(z_k^-) = w(z_k^+);$ (2)

ii. Stress continuity: $\sigma_{xz}(z_k^-) = \sigma_{xz}(z_k^+); \quad \sigma_{yz}(z_k^-) = \sigma_{yz}(z_k^+); \quad \sigma_{zz}(z_k^-) = \sigma_{zz}(z_k^+);$ (3)

iii. Strain continuity: $\varepsilon_{xx}(z_k^-) = \varepsilon_{xx}(z_k^+); \quad \varepsilon_{yy}(z_k^-) = \varepsilon_{yy}(z_k^+); \quad \varepsilon_{xy}(z_k^-) = \varepsilon_{xy}(z_k^+).$ (4)

As usual, the normal stress σ_{zz} is assumed negligible. For each layer the constitutive relation can be written with respect to the material coordinate system or to the global coordinate system, as follow:

$$\{\sigma_{123}\} = [C_{123}]\{\varepsilon_{123}\}; \qquad \{\sigma_{xyz}\} = [C_{xyz}]\{\varepsilon_{xyz}\}. \tag{5}$$

In equations (1) the coefficients ψ_k and χ_k appear, they can be obtained writing the constitutive equations for the transverse shear stresses at each interface and satisfying the stress continuity conditions (3). This procedure leads to the following recursive formulae:

$$\psi_k = a_k\gamma_x + c_k\gamma_y; \qquad \chi_k = d_k\gamma_x + b_k\gamma_y; \tag{6}$$

where a_k, b_k, c_k and d_k depend on the elastic constants of the layer materials [7].

2.2. Series solution

For the particular case of cross-ply laminates, a series solution is presented. In this case the coefficients $C_{16}, C_{26}, C_{36}, C_{45}$ of each layer are zeros, consequentely the coefficients c_k and d_k in eqns. (6) are also zeros. The strain energy per unit area can be written in the following way:

$$\Psi = \frac{1}{2}\{\delta\}^T[E]\{\delta\}; \tag{7}$$

where $\{\delta\}^T = \{u^0{}_{,x}\,; v^0{}_{,x}\,; u^0{}_{,y}\,; v^0{}_{,y}\,; \gamma_x; \gamma_y; \gamma_{x,x}\,; \gamma_{y,x}\,; \gamma_{x,y}\,; \gamma_{y,y}\,; -w^0{}_{,xx}\,; -w^0{}_{,yy}\,; -2w^0{}_{,xy}\}$ and the symmetric matrix $[E]$ can be written by using the mechanical parameters of the materials in which the layers are made [7]. The applied distributed pressure is considered as:

$$q = \sum_{m,n=1}^{\infty} Q_{mn}\, sin(\alpha x) sin(\beta y); \qquad \alpha = \frac{m\pi}{a}; \quad \beta = \frac{n\pi}{b}. \tag{8}$$

For a sinusoidal loading is $Q_{mn} = q_0$ and for a uniformly transverse load is $Q_{mn} = 16q_0/(\pi^2 mn)$ such that $m, n = 1, 3, 5, \ldots$. The boundary conditions are the following:
$u^0(x,0) = u^0(x,b) = v^0(0,y) = v^0(a,y) = w^0(x,0) = w^0(x,b) = w^0(0,y) = w^0(a,y) = 0;$
$\psi_x(x,0) = \psi_x(x,b) = \psi_y(0,y) = \psi_y(a,y) = 0;$
$N_y(x,0) = N_y(x,b) = N_x(0,y) = N_x(a,y) = M_2(x,0) = M_2(x,b) = M_1(0,y) = M_1(a,y) = 0;$ (9)
where a, b are the plate's dimensions in the x and y direction, respectively. The displacement field solution can be expressed by:

$$u^0 = \sum_{m,n=1}^{\infty} U_{mn}\, c(\alpha x) s(\beta y); \qquad v^0 = \sum_{m,n=1}^{\infty} V_{mn}\, s(\alpha x) c(\beta y); \qquad w^0 = \sum_{m,n=1}^{\infty} W_{mn}\, s(\alpha x) s(\beta y);$$

$$\psi_x = \sum_{m,n=1}^{\infty} X_{mn} c(\alpha x) s(\beta y); \qquad \psi_y = \sum_{m,n=1}^{\infty} Y_{mn} s(\alpha x) c(\beta y). \tag{10}$$

Where $c = cos$, $s = sin$ and the unknown amplitudes $U_{mn}, V_{mn}, \ldots$ must be determined. They can be obtained solving the following:

$$[S]\{\Delta_c\} = \{0,0,0,0,-Q_{mn}\}^T = \{F\}; \tag{11}$$

where $\{\Delta_c\}$ is the unknown amplitudes vector, $\{F\}$ is the column vector of the forces, and the coefficients of the symmetric matrix $[S]$ are:

$S_{1,1} = -m^2\alpha^2 E_{1,1} - n^2\beta^2 E_{3,3}$; $S_{12} = -mn\alpha\beta(E_{1,4} + E_{2,3})$; $S_{1,3} = -m^2\alpha^2 E_{1,7} - n^2\beta^2 E_{3,9}$;
$S_{1,5} = m^3\alpha^3 E_{1,11} + mn^2\alpha\beta^2(E_{1,12} + 2E_{3,13})$; $S_{2,2} = -m^2\alpha^2 E_{2,2} - n^2\beta^2 E_{4,4}$;
$S_{2,3} = -mn\alpha\beta(E_{2,9} + E_{4,7})$; $S_{2,4} = -m^2\alpha^2 E_{2,8} - n^2\beta^2 E_{4,10}$; $S_{1,4} = -mn\alpha\beta(E_{1,10} + E_{3,8})$;
$S_{2,5} = m^2\alpha^2 n\beta(2E_{2,13} + E_{4,11}) + n^3\beta^3 E_{4,12}$; $S_{3,3} = -E_{5,5} - m^2\alpha^2 E_{7,7} - n^2\beta^2 E_{9,9}$;
$S_{3,4} = -mn\alpha\beta(E_{7,10} + E_{8,9})$; $S_{3,5} = m^3\alpha^3 E_{7,11} + mn^2\alpha\beta^2(E_{7,12} + 2E_{9,13})$;
$S_{4,4} = -E_{6,6} - m^2\alpha^2 E_{8,8} - n^2\beta^2 E_{10,10}$; $S_{4,5} = m^2\alpha^2 n\beta(2E_{8,13} + E_{10,11}) + n^3\beta^3 E_{10,12}$;
$S_{5,5} = -m^4\alpha^4 E_{11,11} - m^2n^2\alpha^2\beta^2(2E_{11,12} + 4E_{13,13}) - m^4\beta^4 E_{12,12}$. (12)

2.3. Finite element development

The displacement field $\{\mathbf{u}\}^T = \{u^0, v^0, \gamma_x, \gamma_y, w^0\}$ inside an element can be written as:

$$\{\mathbf{u}\} = [\mathbf{N}]\{\mathbf{u}^e\}; \tag{13}$$

where $\{\mathbf{u}^e\}$ is the collection of the nodal degrees of freedom $\{\mathbf{u}_i^e\}$ which is in this formulation $\{\mathbf{u}_i^e\}^T = \{u_i^0, v_i^0, \gamma_{xi}, \gamma_{yi}, w_i^0, w_{,xi}^0, w_{yi}^0, w_{xyi}^0\}$. In particular $u^0, v^0, \gamma_x, \gamma_y$ are approximated by Lagrangian interpolation functions, while for w^0 Hermitian interpolation functions are used. Both of the interpolation functions are defined on a four node two-dimensional finite element. The vector $\{\delta\}$ of eqn. (7) can be correlate to the $\{\mathbf{u}^e\}$ using the following relation:

$$\{\delta\}_{13} = [\mathbf{R}]_{13x32}\{\mathbf{u}^e\}_{32}. \tag{14}$$

Using eqn. (14) it is possible to reach the classic expression of the the plate element stiffness matrix in the Finite Element Method without increasing the total degrees of freedom:

$$[\mathbf{K}^e] = \int_\Omega [\mathbf{R}]^T[\mathbf{E}][\mathbf{R}]d\Omega. \tag{15}$$

3. DELAMINATION BUCKLING

In this section a one-dimensional problem corresponding to a narrow plate is presented. With reference to Fig. 1, the thickness parameters of the plate are such that $t/T << 1$.

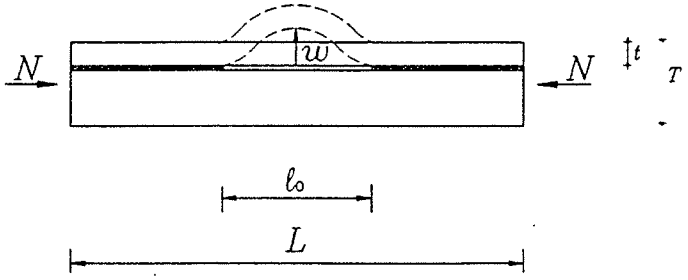


Fig. 1: Buckled narrow plate

In particular the stiffness parameters k_1 and k_2 can be expressed as:

$$k_1 = E_1 Bt; \qquad k_2 = E_2 B(T - t); \tag{16}$$

in which $k_1/k_2 << 1$ and E_1, E_2 are the Young moduli of the two plates, and B is the plate's width. An axial compression load N is applied until the upper layer begins to buckle and the delaminated area may subsequently spread.

The force-displacement relations for the buckled plate can be expressed [8,9]:

$$\frac{\sigma}{\sigma_c} = 1 + \left[\frac{\pi^2}{8} + \frac{3}{4}\left(\frac{\ell}{t}\right)^2 \frac{k_2}{k_1+k_2}\right]\xi^2; \qquad \frac{u_L}{L} = \left(\frac{\pi^2}{3}\right)\left(\frac{t}{\ell}\right)^2 \frac{\sigma}{\sigma_c} + \frac{\pi^2}{4}\left(\frac{\ell}{L}\right)\frac{k_1}{k_1+k_2}\xi^2; \quad (17)$$

where $\sigma = N/BT$ and $\sigma_c = N_c/BT = \pi^2 E_1 t^3 (k_1 + k_2)/(3\ell^2 k_1 T)$. The energy release rate G_T is given by:

$$G_T = -\frac{1}{2}\frac{\partial}{\partial \ell}\left[(N - N_c)\, u_2 \xi^2\right] = \frac{\pi^4 E_1 B t^3}{12\ell^2}\left\{2\xi^2 + \left[\frac{3\pi^2}{16} + \frac{3}{8}\left(\frac{\ell}{t}\right)^2\left(\frac{k_2}{k_1+k_2}\right)\right]\xi^4\right\}. \quad (18)$$

In the relation (18) the two contributions of mode I and II are involved. They result as follows [8,9]:

$$G_I = \frac{\pi^4 E_1 B t^3}{12\ell^2}\left[2\xi^2 + \frac{3\pi^2}{16}\xi^4\right]; \qquad G_{II} = \frac{\pi^4 E_1 B t^3}{12\ell^2}\left[\frac{3}{8}\left(\frac{\ell}{t}\right)^2\left(\frac{k_2}{k_1+k_2}\right)\xi^4\right]. \quad (19)$$

By using eqn. (19) and an appropriate delamination criterion it is possible to study the delamination growth and the equilibrium force-displacement path of the structure.

4. FIRST-PLY FAILURE AND DELAMINATION CRITERIA

In the analysis of laminates, a laminate may be assumed to fail when in one of the laminae the assumed strength criterion displays failure. Particularly the failure theories are cases generating from the general form proposed by Tsai & Wu [6]:

$$F_1\sigma_1 + F_2\sigma_2 + 2F_{12}\sigma_1\sigma_2 + F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + F_{44}\sigma_4^2 + F_{55}\sigma_5^2 + F_{66}\sigma_6^2 = f(\sigma). \quad (20)$$

where if $f(\sigma) \geq 1$ there is failure. Writing eqn. (20) the assumptions demonstrated by Wu [6] are used and the stress σ_3 is considered negligible. The coefficients F_i and F_{ij} for the Tsai-Wu failure criterion can be found in [6].

Another aspect of failure is the delamination phenomenon, in this case the maximum load capacity and delamination growth are analyzed by an energy criterion based on the fracture mechanics results. Delamination growth is related to the stress state of the crack zone, with the stress intensity factors K_I, K_{II} and K_{III}, or with the strain energy release rates G_I, G_{II} and G_{III}. In plates subjected to axial compression loads, the buckling of the layers is associated with delamination related to Mode I and Mode II. The mixed mode delamination growth is not observed to follow a single propagation law, thus various laws have been used. A more appropriate interaction relation to describe delamination growth is considered as [10]:

$$\left(\frac{G_I}{G_{IC}}\right)^m + \left(\frac{G_{II}}{G_{IIC}}\right)^n = 1; \quad (21)$$

where the values of the exponents m and n have been found in [10] and can be fixed both to the unit resulting in good agreement with the experimental data. Assume as:

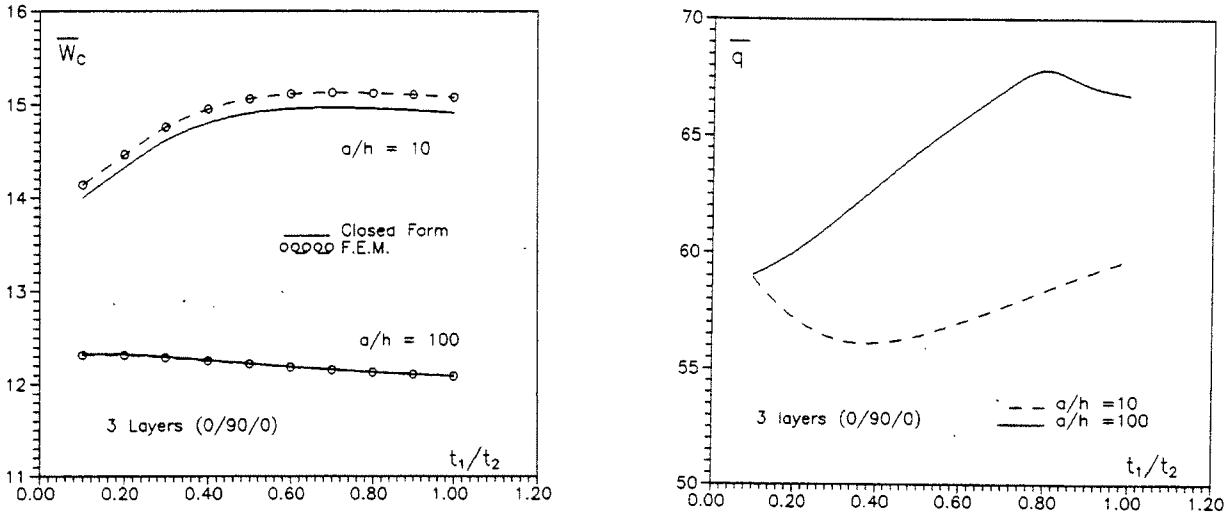
$$G_{IC} = \Gamma B; \quad G_{IIC} = \gamma \Gamma B; \quad m = n = 1; \quad \textit{and} \quad \gamma \geq 1; \quad (22)$$

where Γ is the surface adhesion energy per unit area of opening, and using Eqn. (21) delamination is assumed to occur when:

$$G_I + \frac{G_{II}}{\gamma} = \Gamma B; \rightarrow 2\xi^2 + \frac{3\pi^2}{16}\xi^4 + \frac{1}{\gamma}\left[\frac{3}{8}\left(\frac{\ell}{t}\right)^2\left(\frac{k_2}{k_1+k_2}\right)\xi^4\right] = \frac{12\Gamma\ell^2}{\pi^4 E_1 t^3} = \alpha. \quad (23)$$

5. NUMERICAL RESULTS AND CONCLUSIONS

A square cross-ply plate subjected to a transverse uniformly distributed load (eqn. 8) and simply-supported on all edges is analyzed.



a) Central deflections; *b) First-ply failure loads*

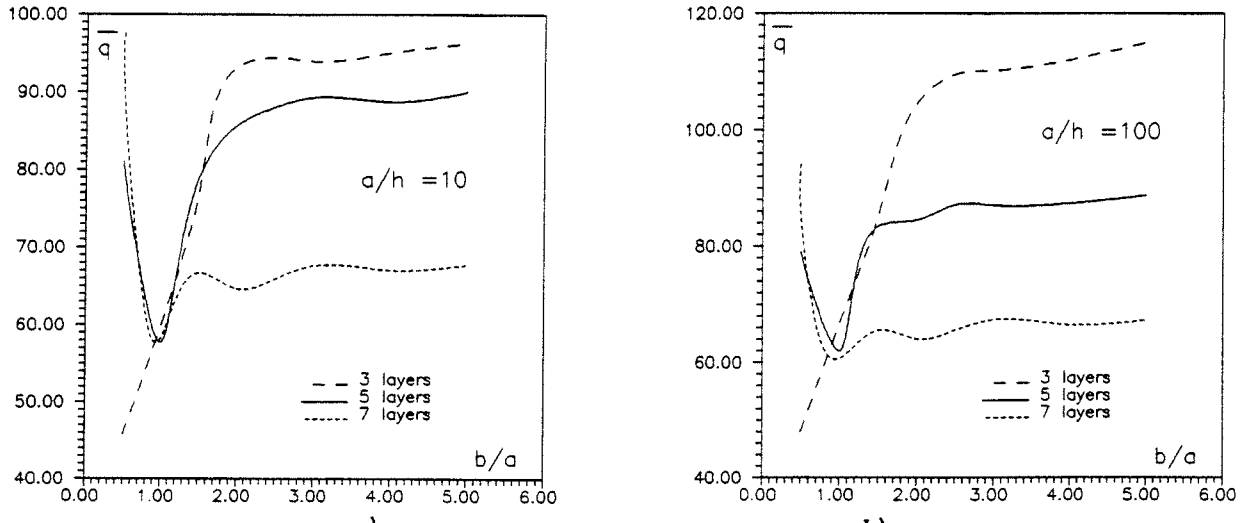
Figs. 2 - Cross-ply plates with different a/h and t_1/t_2

The laminate is composed of three layers $(0^0/90^0/0^0)$ made in Graphite-Epoxy T300/5208. The material has the following mechanical properties:

$E_1 = 19.2 \cdot 10^6\ psi$; $E_2 = 1.56 \cdot 10^6\ psi$; $G_{12} = G_{13} = 0.82 \cdot 10^6\ psi$; $G_{23} = 0.49 \cdot 10^6\ psi$;
$\nu_{12} = 0.24$; $X_T = 219.5 \cdot 10^3\ psi$; $X_C = 246.0 \cdot 10^3\ psi$; $Y_T = Z_T = 6.35 \cdot 10^4\ psi$;
$Y_C = Z_C = 6.35 \cdot 10^6\ psi$; $R = 9.8 \cdot 10^3\ psi$; $S = T = 12.6 \cdot 10^3\ psi$.

When the series solution is adopted the displacements are expressed as in eqns. (10) so the differential equilibrium equations are satisfied. A quarter of the same plate was also discretized by a 5x5 mesh of the proposed two-dimensional finite elements, because of the symmetry both in load and in geometry.

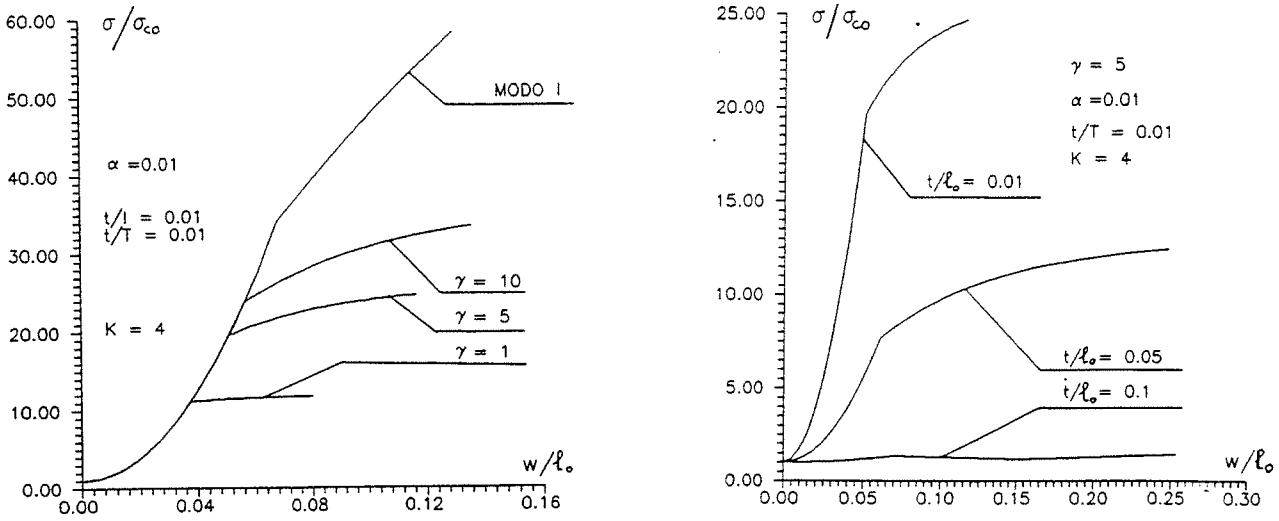
In Fig. 2a the finite element and the series solution results of the adimensionalised central transverse displacement $\overline{W}_c = ((W_c E_2 h^3)/(q_0 a^4)) \cdot 10^3$ v/s the t_1/t_2 ratio, t_1 being the outer layers thicknesses and t_2 the inner layer thickness, are compared. The results show that the error in the finite element approximation with respect to the series solution is acceptable. In Fig. 2b, for the same plates as the previous example, the first-ply failure loads are determined using the Tsai-Wu failure criterion and the series solution. Also in this case the effect of the shear deformability on the adimensionalised failure loads $\overline{q} = ((q_0/E_2) \cdot (a/h)^2) \cdot 10^3$ is very clear which will be overestimated if the shear effects are neglected.



a) *b)*

Figs. 3a, b - First-ply failure load v/s b/a ratio for $a/h = 10$ and $a/h = 100$

In Fig. 3a and 3b, the effect of the layering (3, 5 and 7 layers) and the b/a ratio (b = side parallel to the y axis, a = side parallel to the x axis) is considered. The layers are made by Graphite-Epoxy T300/5208 and the results are obtained using the series solution and the Tsai-Wu failure criterion. In this case the shear effects are less evident than in the previous examples.
In Fig. 4a a plot of σ/σ_0 versus w/ℓ_0 where the adhesion energy parameter is $\alpha_0 = 12\ell_0^2\Gamma/(\pi^4 E_1 t^3)$ is given. In particular the figure shows the influence of the parameter γ on the bearing capacity. It is evident that for Mode I ($\gamma = 0$) behaviour is better than the condition in the mixed mode ($\gamma = 1$). The effects produced by the ratio t/ℓ_0 on the solution are plotted in Fig. 4b. In this figure, the adhesion energy parameter was fixed to a value of $\alpha_0 = 0.01$, while the penalty parameter γ was fixed to a value of 5. The curves σ/σ_{c0} and w/ℓ_0 are drawn for different values of t/ℓ_0. Observing the numerical results, a less stable behaviour when the t/ℓ_0 ratio increases is apparent. This because of the decrease in the energy contribution from Mode II, when the t/ℓ_0 ratio increases.



a) Influence of γ on the bearing capacity; b) Effect of the defect size parameter t/ℓ_0
Figs. 4a, b - Delamination failure ($K = E_1/E_2$)

REFERENCES

1. Pagano N. J. - *Exact solution for composite laminates in cylindrical bending* - J. Comp. Mater., 3, 398-411, 1969.
2. R. Zinno, E. J. Barbero - *A three-dimensional layer-wise constant shear element for general anisotropic shell-type structures* - Int. J. Num. Meth. Engng., 37, 2445-2470, 1994.
3. Murakami M. - *Laminated composite plates theories with improved inplane responses* - Proc. ASME pressure vessels and piping conference, 257-263, 1984.
4. Di Sciuva M. - *An improved shear deformation theory for moderately thick multilayered anisotropic shells and plates* - J. Appl. Mech., 54, 589-596, 1987.
5. Hoffman O. - *The brittle strength of orthotropic materials* - J. Comp. Mater., 1, 200-206, 1967.
6. Tsai S. W., Wu E. M. - *A general theory of strength for anisotropic materials* - J. Comp. Mater., 5, 58-80, 1971.
7. Bruno D., G. Porco, R. Zinno - *Failure of multilayered anisotropic plates analysed by a layer-wise zig-zag theory* - Report 163, Dept. of Struct. Engng., Univ. of Calabria, Cosenza (Italy), 1994.
8. Bruno D., Porco G. - *Delaminazione nelle piastre laminate composite con difetti interlaminari* - XXII AIAS National Conference, Forlì, Italy, 6-9 Oct., 1993.
9. Bruno D., Porco G. - *Delaminazione in buckling di piastre laminate* - XXIII AIAS National Conference, Rende (CS), Italy, 21-26 Sept., 1994.
10. Amar Garg C. - *Delamination - a damage mode in composite structures* - Engng. Fract. Mech., 29, 5, 557-584, 1988.

DESIGN MODEL FOR BOILER HEAT TRANSFER

Bo Lampenius

Laboratory Department, Imatran Voima Oy
P.O.BOX 112, Vantaa, Finland

1. ABSTRACT

Technical as well as economical reasons support improved design of power plants. Correct design of heat transfer and combustion gives direct money savings in O&M costs and is also favourable to reduction of harmful emission. In plant retrofits and modifications, which aim at better performance and lower emissions, operation normally changes significantly. This happens for an example, when old burners are changed to low-NOx burners or pulverised fuel combustion is replaced by fluidised bed grate. When improvements are planned for the burning process in the furnace, it is important to know how it affects the heat transfer of the boiler as a whole. In order to predict the changes in the heat transfer while sizing the modification a heat transfer model is needed. The model has to be accurate, but also easy to use and fast, so that the design costs will remain reasonable.

This paper describes a new boiler heat transfer model, which is an integrated part of a simulator named SOLVO®. The development started in 1994, when it was found that the old boiler model of SOLVO® could not sufficiently predict changes in boilers with high radiative heat transfer. The method which was chosen to solve the radiation is the discrete-transfer method developed by Lockwood and Shah. The two other modes of heat transfer, convection and conduction are also included in the model. The modelling of an individual boiler is fast and easy, due to the fact that the model is modular, i.e. consists of predefined components which can be picked up and connected together through a graphical interface. The fuel, air and flue gas cycle as well as the water-steam cycle are parts of the model. The validation of the model will take place in 1995. The calculation time on a PC (Pentium, 60 MHz) is 1-5 min.

A case of fluidised bed grate modification will be presented. Calculated results of thermal performance, are validated against measurements of plant instrumentation at different loads and operation points.

2. INTRODUCTION

This paper describes a design model for boiler heat transfer which has been developed in Imatran Voima Oy in 1994 -1995. The model is a modular steady-state model, which can be used to model the fuel, air, flue gas and water-steam cycles of the boiler.

In the next chapters will be presented the theory of the model, then the procedure of setting up a model and then a comparison between the model results and measurements. The paper ends with a short conclusion.

3. THEORY

3.1 Radiation

The most important task in developing a boiler model is to choose the heat radiation model. The reason is that radiation is physically very complicated and also the dominant heat transfer mechanism in the furnace of the boiler. The criteria set for the model where:

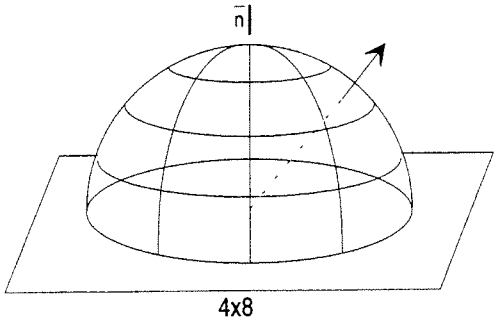
- reasonable calculation time (10 of seconds)
- good accuracy
- possibility to model different kind of geometry, including the radiation heat exchangers in the furnace.
- references in many applications

The model which was chosen was developed in cooperation with the Technical Research Centre of Finland and is based on the Discrete Transfer method.

3.1.1 Discrete Transfer method

The theory of the Discrete Transfer method (DT) is developed by Lockwood and Shah [1]. The method is based on rays sent out in fixed directions from surface cells which then are traced in the radiation source (= flue gas in boilers). The DT-method can be classified as a mixed method between the zone, flux and the Monte Carlo methods.

By tracing the rays between the heat exchange surfaces it is possibly to calculate the net powers to the surfaces. The radiation source between the surfaces changes the intensity of the rays due to the local absorption, emission and scattering. From experience it is known that 32 rays from each surface cell is enough to give a good accuracy [2].

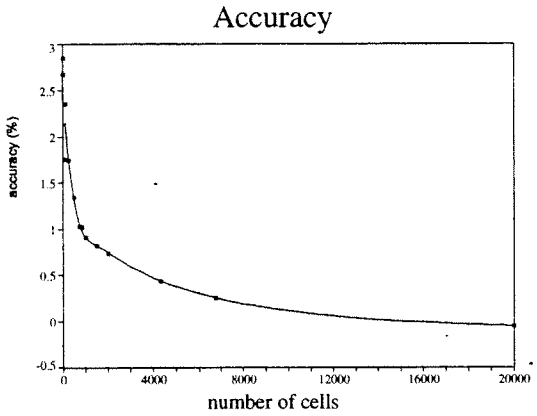


Picture 1: The picture shows the sectors from which centre the rays of the surface cell are being sent out. The normal angle is divided in 4 parts and the azimuthal angle in 8 parts.

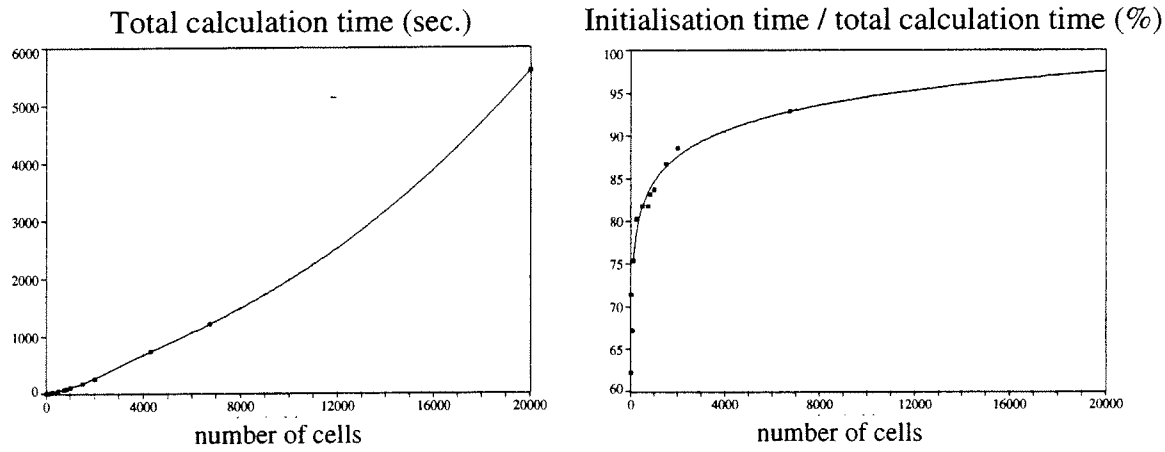
The values of temperature at outlet and total radiative heat transfer from the furnace are usually predicted satisfactorily by using constant absorption and scattering coefficients. However the radiative properties can be changed in the different parts of the boiler by the user. If desired, it is also possibly to let the model calculate the emissivity of the flue gas, from the gas components of H_2O and CO_2 taking into account the soot distribution given by the user.

The calculation time increases and the accuracy improves rapidly with a increase of the cell grid. The time is mainly spent to the initialisation of the model (>80 %), which fortunately only has to be made once.

Tests were made with the DT-model before it was chosen. They showed that a grid consisting of 2000 cells gives a good accuracy in a reasonable calculation time ≈10 s [3].



*Picture 2: The picture shows the dependence between the accuracy and the cell grid in a simple test case. The dimensions where 9m*10m*35m, the temperature of the radiative source 1600 K and the temperature of the walls 600 K.*



*Picture 3: The picture on the left shows the dependence between the total calculation time and the cell grid. The picture on the right shows the percentage of the initialisation calculation related to the total calculation time and the cell grid. Results from a simple test case (dimensions: 9m*10m*35m).*

3.2 Convection

The convection part of heat transfer in the furnace is very small, but increases rapidly after the furnace, to be the dominant heat transfer mode downstream the furnace. The *transferred heat* due to convection is calculated in the model according to equation 1 [4].

$$\Phi_c = hA\Delta T_{lm} \tag{1}$$

The *heat transfer coefficient* changes with load as shown in equation 2.

$$h = h_0 (m / m_0)^{0.6} \tag{2}$$

3.3 Conduction

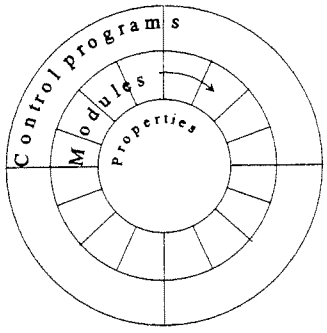
Conduction between the water cooled inner side and the heated outer side of a heat exchange wall, raises the outside temperature of the wall, thus decreasing the radiation heat transfer to it.

A heat exchanger wall is divided in the model into three parts: slag, metal and magnetite. These materials thicknesses and properties are given by the user and used in the calculation of the wall conduction. The *total thermal resistance* caused by conduction is given by equation 3 [4].

$$R_{tot.cond} = \left(\frac{T_{s,1} - T_{s,2}}{q_x} \right) = \frac{\sum L}{\sum k} / A \qquad (3)$$

3.4 Model environment

A computer-based modular simulation environment, named SOLVO®, has been developed at Imatran Voima Oy (IVO) to evaluate the design and off-design performance of power plant processes. Steady-state operation of conventional boilers, integrated gasification combined cycles (IGCC) or steam turbine plants, alone or linked together, can be predicted in various applications, e.g. process design and process computer systems. The model is based on conservation of the mass and heat flows, including solution of additional equations to model characteristic behaviour of a plant component. The approach for solving the set of equations for the whole system is sequential modular.



Picture 4: The SOLVO® simulation environment; The control programs, control all calculations, modules perform the calculations and property subroutines are used to calculate the fluid propoerties.

A versatile macro language is also included to be used for controlling the simulations, print outs etc. It is a FORTRAN-like language including mathematical functions, loops, condition-blocks and sub-macros. The macros are generated via a text editor.

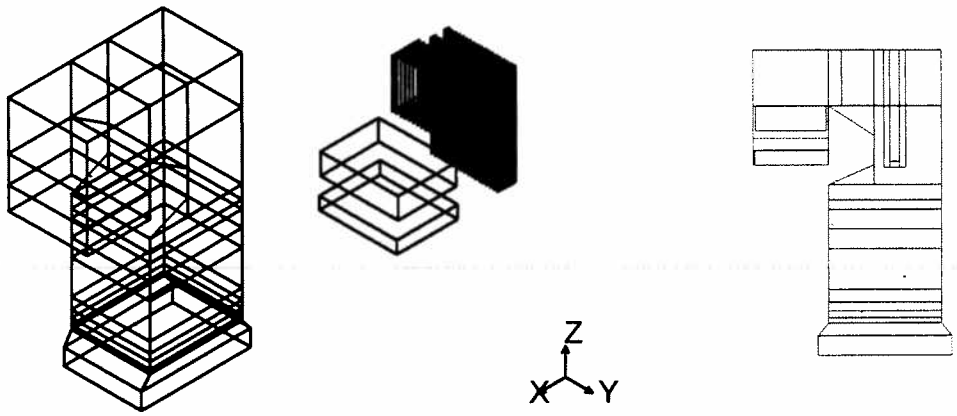
4. MODEL DESCRIPTION

4.1 Configuring the topology and geometry

Defining the topology of the model is the first task when a new application is configured, to do this a graphical user interface is used. By picking components from the SOLVO®-library and connecting these together the topology of the model is made. New components which were added to the library in this project where: *FurnaceWall, FurnaceNose, FurnaceBFB, FurnaceHopper, FurnaceRoof, FurnaceHeater* and *FurnaceBurner*. With these components

and with some of the old *components (FuelTank, Pipe,Valve,etc,)* the topology of the boiler is built.

After that the topology is set up, the geometry dimension of the boiler furnace and its heat exchanger are given. After this it is possible to check the geometry visually. (see picture 5).



Picture 5: The picture shows the model of the BFB-boiler of Rauhalahti. At the left can be seen in 3D the outer walls of the boiler and the heat exchange surfaces inside the boiler. At the right can be seen the same boiler in 2D.

4.2 Parametrising the model

The model has to be parametrised to be able to run it in different loads and operating modes. With the help of these parameters the components and the whole process can be simulated. A parametrised model can be run using only a few main measurements, which tell the operating mode, the load, the control adjustments and the state in plant limits.

The model parametrisation can be based on process values from a design point or on physical properties such as the convective heat transfer coefficients. Typically a existing boiler is parametrised based on process values from the commision test. When changes are planned to a boiler such as adding new heat surfaces the parametrisation should be based on physical properties.

5. CASE RAUHALAHTI

5.1 The Rauhalahti boiler

The peat fired boiler of Rauhalahti power plant was built in 1986 and was converted from a traditional pulverised fired boiler to the worlds largest bubbling fluidised bed (BFB) boiler in 1993. The main fuel is peat, but also many wood-based fuels can be burnt in the bed. Furthermore, it is possible to use the old burners and burn pulverised coal. The fuel power is 295 MW and the steam parameters are:

- live steam pressure 136 bar
- live steam temperature 533 C
- live steam mass flow rate
  - with peat/wood in BFB 100 kg/s
  - with peat/wood in BFB + coal 110 kg/s
  - only pulverised coal 70 kg/s

The model of Rauhalahti boiler was constructed using 46 components, the flue gas channel was divided into 15 zones and from the heat exchange surfaces 62848 rays were sent out. A cell grid of 17*9*28 cells was used in the flue gas zone. The calculation time of the model is 1-5 minutes (Pentium 60 Mhz) after the initialisation stage.

5.2 Comparisons between measurements and predicted values of the models

The results from the calculations can be seen in table 1.

Boiler load	**100 %**		**70 %**		**110 %**		**35 %**	
Heat power (kW)	Meas.	Model	Meas.	Model	Meas.	Model	Meas.	Model
hot economiser	29165	29165	n.a.	19869	n.a.	32384	n.a.	6936
evaporator	97603	97232	74383	80846	105254	102606	n.a.	56146
superheater 1	41529	41520	29224	27504	100394	46236	n.a.	10038
superheater 2 (rad.)	31209	31301	22550	23077		34454	n.a.	12778
superheater 3	18803	18806	14095	12630		21233	n.a.	5035

Table 1. Comparison between the measured values and the values predicted by the model, at different loads. n.a. = not measured

6. CONCLUSIONS

A design boiler heat transfer model has been developed and added to a simulation environment called SOLVO®. The model was developed to meet the needs of a relatively fast, easy to use, but still accurate heat exchange design model. The need arised from modernisation projects where boilers are modified to improve the combustion and to reduce the pollutants.

The paper describes the implemented heat transfer models, especially the used radiative model, the Discrete Transfer method which is the most efficient with respect to calculation time and accuracy.

The configuration of an application was presented step by step and the simulation environment of SOLVO® and the user interface were described.

Main data of Rauhalahti power plant BFB-boiler was shown and the models results were compared to measured ones. The model proved to be accurate and easy to use.

The model will be tested in the near future with other boiler types. Some research work and development is still needed to make it even further easier to use and to decrease the calculation time. The calculation time can be reduced, e.g., by using symmetry planes in the DT-model in symmetric boilers.

REFERENCES

1. Lockwood, F. C., Shah, N. G., A new radiation solution method for incorporation in general combustion prediction procedures. 18th Symposium (int.) on combustion,
2. The Combustion Institute, 1405-1414, 1981
3. Jacobson, T., Säteilylämmönsiirtoyhtälön ratkaiseminen Discrete-Transfer-menetelmällä. Licenciate Thesis, Technical University of Helsinki 1992. 57 p. In Finnish
4. Lampenius, B., Internal memo DLP1-A6-93, Imatran Voima Oy, 1993. 15 p. In Finnish
5. Incropera, F.P., Dewitt, D.P., Fundamentals of heat transfer, John Wiley&Sons 1981. 791 p

Modelling of RPSA Air Separation

Harry Verelst, Li-Qun Zhu and Gino V. Baron

Vrije Universiteit Brussel, Department of Chemical Engineering,
Pleinlaan 2, B1050 Brussels, Belgium

A simplified model of the Rapid Pressure Swing Adsorption air separation process was used to study the effect of system parameters on the performance. The results were compared to experimental results obtained in a laboratory RPSA system for three different commercial adsorbents. These were selected to experimentally verify the trends predicted by simulation in a wide range of conditions. It appeared that the often used equilibrium model cannot be used with confidence and that a more complex model with linear driving force (LDF) as used in this study was not adequate in all conditions either. Results show that the performance of a single column RPSA is strongly limited by mass transfer resistances, especially at the shorter cycle times. For some adsorbents the micropore diffusion is limiting and the LDF model works well. For others both the macropore diffusion and convection appear to be limiting, and hence the model has to be expanded and adapted to the material used.

1. INTRODUCTION

Pressure swing adsorption (PSA) processes are important unit operations for industrial gas separation. With the improvement of adsorbent properties and in process design over the past few years, this operation can now compete with cryogenic separation in ever larger capacities[1]. One way to obtain higher productivity of the equipment is to decrease the cycle times which leads to lower adsorbent inventory and cost. Rapid pressure swing adsorption (RPSA, [2]) is such a process in which a single column is packed with small adsorbent particles and uses short cycle operation (2-20s). Several experimental and theoretical studies about this short cycle adsorption process have appeared [3-8]. An even more efficient process has been proposed by Air products and Chemicals Inc.[9] which can produce 50% pure O_2 with a productivity of 240$Nm^3/h/m^3$ adsorbent.

The separation of air by the RPSA process is based on an equilibrium selectivity difference between nitrogen and oxygen, but in short cycle time RPSA processes, the working capacity and selectivity of the adsorbent will mainly depend on the limiting mass transfer rates. Doong and Yang [6] and Baron [5] show that mass transfer resistance may become important under rapid cycling conditions when they modelled RPSA operation using a LDF model. With the assumption of local equilibrium, Guan and Ye [7] simulated the air separation process by RPSA with a 13X adsorbent, but they mention that agreement with experiments is better when mass transfer effects are included in the model.

2. PRINCIPLE OF RPSA

A Rapid Pressure Swing Adsorption (RPSA) column consists of a 1 to 2 m long cylinder, packed with 5A or 13X molecular sieve particles of .2 to .5 mm diameter operated in the following cyclic manner. On one end of the column feed gas is entered at a pressure of a few bars by opening the feed valve for about 1 second. Then follows a pressure equalizing (delay) period of 2 to 3 seconds and a purge period of 5 to 15 seconds by opening the purge

valve to a lower or atmospheric pressure. At the other end of the column, product is withdrawn continuously through a control valve or pressure regulator. A product surge tank is often placed between the column and this valve.

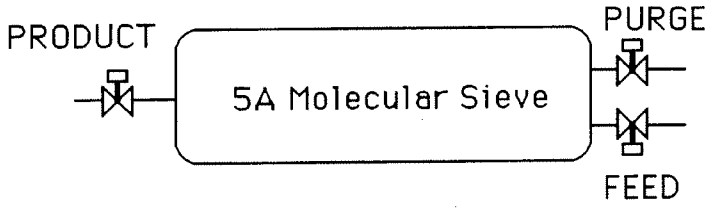


Figure 1. RPSA single column adsorber

In the case of air separation, compressed feed air enters the column, nitrogen is preferentially adsorbed and oxygen passes on. At the feed end of the column, the sieve is fairly rapidly saturated with nitrogen and regeneration becomes necessary. Opening the purge valve causes the pressure to fall, nitrogen to desorb from the sieve and to leave the column in backflow. Some of the oxygen from the product end of the column is then used as a purge gas to further elute the nitrogen (reflux). As this occurs at a lower pressure, only a small amount of the product oxygen is needed for this purging. A nitrogen enriched purge gas is obtained and oxygen enriched air is continuously available at the product end.

3. EXPERIMENTAL

The experimental parameters are summarised in Table 1. For 5A, also data from [11] were used. Table 2 gives the Henry law constants at 25°C of the materials used in examples in this paper. Mass transfer coefficients were determined using a gas chromatographic setup. This worked well for 5A, where the coefficient obtained did not depend on the particle size used, pointing at a mass transfer limitation on the level of the crystals. For 13X, the values did not correspond very well with those predicted by the accepted equations. Values for 13X used in the models were adjusted to reproduce the RPSA experiments, starting from the rough experimental values.

Table 1. Experimental parameters.

Adsorbents:	5A with 20% binder, 13X with 20% binder, 13X binderless
Column Diameter	0.0125 m
Particle Size	250-425 μm
Column Length	1 m and 1.6 m
Permeability	about 50. 10^{-12} m^2
Weigh of adsorbent	about 100 g/m column
Feed Pressure	2 to 4.5 bar
Temperature	298 K
Feed concentration	78.11% N_2, 20.96% O_2 and 0.93 Ar
Timing	feed 0.2-4 s; delay 0.5-7 s; purge 1-15 s

Table 2. Henry law constant (mol/kg.bar) for N_2 and O_2 in the Langmuir isotherm at 25°C.

adsorbent	KN_2	KO_2	regeneration temp. °C
13X without binder	0.58	0.137	350
5A with binder	0.376	0.119	250

4. MODEL EQUATIONS AND SIMULATION

Figure 2 shows the most important mass transfer processes to be considered in complete models. The inclusion of all these resistances yields equations which are too complex to handle in routine modelling of the dynamic behaviour of an adorbent bed. In most adsorption operations one can adequately approximate the mass transfer processes by a single lumped mass transfer coefficient, or even assume equilibrium.

A simplified LDF model was developed, but in which all parameters were measured in independent experiments or setups, and based on the following assumptions:

- ideal gas behaviour
- gas flow follows the Forschheimer equation
- gas plug flow (no radial gradients) with axial dispersion
- isothermal model
- equilibrium follows Langmuir equation
- argon and oxygen are lumped as one component
- external, macropore and micropore transport resistances are lumped in a single mass transfer parameter
- valves are characterized by their valve constants
- dead volumes in piping and column are well mixed cells

The equations corresponding to the described model are a set of four coupled, extremely stiff parabolic partial differential equations (internal and external partial pressures of two components) with a number of boundary conditions in the form of algebraic and differential equations (valves, piping).

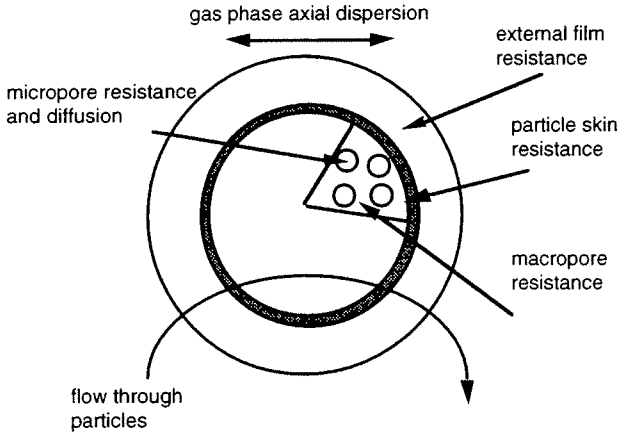


Figure 2. Transport resistances in adsorbent beds

The mass balance for component i in fluid phase is:

$$\frac{\partial(v_f P_i)}{\partial z}+A\frac{\partial P_i}{\partial t}+B\frac{\partial q_i}{\partial t}-D_L\frac{\partial^2 P_i}{\partial z^2}=0$$

$$A=\varepsilon_{ext}+\varepsilon_{macr}$$

$$B=1-\varepsilon_{ext}-\varepsilon_{macr}$$

and the adsorption equilibrium:

$$q_i=\varepsilon_{micr}P_i + (1-\varepsilon_{micr})\frac{K_d P_i}{1+\Sigma L_i P_i}$$

$$K_d=\frac{\rho_{cristal}K_i RT}{(1-\varepsilon_{micr})}$$

with P_i the partial pressure of component i, q_i the amount adsorbed, x the O_2 mole fraction, P total pressure, v_f the gas velocity. The mass transfer rate is given by a linear driving force model:

$$\frac{\partial q_i}{\partial t}=k_i(q_i^*-q_i)$$

with k_i the mass transfer coefficient. The relation between flow rate and column pressure gradient is expressed by Forchheimer's equation:

$$-\frac{\partial P}{\partial z}=\frac{\mu}{A_p}v_f + 1.75\,\frac{(1-\varepsilon)\rho_g}{\varepsilon^3 d_p^2}v_f^2$$

All dead volumes, piping and valves were modelled as well. The inlet and outlet flow are described by the equation for the flow of gas through a nozzle

$$V_n= K_v\sqrt{\frac{p_1(p_1-p_2)}{\gamma T}} \qquad \text{if } p_1/p_2 \le 2$$

$$= K_v\frac{p_1}{2\sqrt{\gamma T}} \qquad \text{if } p_1/p_2 \ge 2$$

Boundary conditions at Z=0, for the feed step:

$$\frac{\partial P_{z=0}}{\partial t}=\frac{P_f V_{n\,feed} - P_{z=0}v_{f\,z=0}A}{V_{z=0}}$$

$$\frac{\partial x_{z=0}}{\partial t}=\frac{V_{n\,feed}P_f(x_f - x_{z=0})}{V_{z=0}P_{z=0}}$$

for the delay step:

$$\frac{\partial P_{z=0}}{\partial t}=-\frac{v_{f\,z=0}P_{z=0}A}{V_{z=0}}$$

$$\frac{\partial x_{z=0}}{\partial t}=0$$

for the purge step:

$$\frac{\partial P_{z=0}}{\partial t}=-\frac{v_{f\,z=0}P_{z=0}A}{V_{z=0}}-\frac{P_{pur}V_{n\,pur}}{V_{z=0}}$$

$$\frac{\partial x_{pur}}{\partial t}=\frac{v_{f\,z=0}A\,(x_{pur}- x_{z=0})}{V_{z=0}}$$

And at Z=L:

$$\frac{\partial P_{z=L}}{\partial t}=\frac{v_{f\,z=L}P_{z=L}A}{V_{z=L}}-\frac{P_{pro}V_{n\,pro}}{V_{z=L}}$$

$$\frac{\partial x_{pro}}{\partial t}=\frac{v_{f\,z=L}A\,(x_{z=L}- x_{pro})}{V_{z=L}} \qquad \text{if } v_{f\,z=L} \ge 0$$

$$\frac{\partial x_{pro}}{\partial t}=0, \quad \frac{\partial x_{z=L}}{\partial t}=0 \qquad \text{if } v_{f\,z=L} \le 0$$

A large number of cases was simulated using this isothermal linear driving force. The partial differential equations in the model were reduced to ordinary differential equations by the method of orthogonal collocation [10] using typically 19 internal grid points. A Gear algorithm [12] was used for the integration of the large system of ordinary differential equations. In this work the numerical calculations lead to worst case component material balance errors of less than 2%.

5. RESULTS AND DISCUSSION

Table 3 shows the important if not dominant role of mass transfer limitations in this operation, for a case studied by Guan and Ye [7]. The table gives the simulation result for their equilibrium data and timing and our experimental mass transfer coefficient, obtained for similar adsorbent particles. If the mass transfer coefficient is increased, the result reaches their equilibrium model result. Considerable improvement of this operation by enhancing mass transfer is thus possible.

Table 3. Effect of nitrogen mass transfer coefficient on purity, recovery and productivity.

k_{N2} (s^{-1})	product purity %	product recovery %	productivity O_2 kg/kg ads./day
12*	88.4	19.3	1.17
24	91.9	28.0	1.70
36	91.4	31.8	1.93
480	90.4	40.2	2.31
equilibrium**	90.0	41.0	2.20

Simulation conditions: timing scheme- .5/7/10s; feed pressure- 4.78bar; column diameter- 0.016m,column length- 1.2m, product surge tank- $5.93 \times 10^{-4} m^3$; equilibrium data from [7]; Oxygen mass transfer coefficient changed in proportion.
* Experimental mass transfer coefficient in this work; ** Result from Guan and Ye [7]

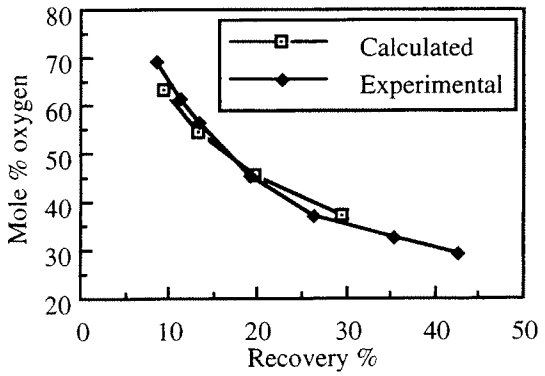


Figure 3. RPSA with 2.5 bar feed in a 1 m column with 5A (250°C), cycle timing .5/1/4 s. Experiment and simulation

For 5A the simple LDF model can predict the performance over the full range as shown in Figure 3, as the resistance is at the crystal, and macropore resistance is negligible. For 13X however, the model cannot fit the full purity range with a single, constant lumped mass transfer resistance. It appears that the mass transfer coefficient needed to fit the experimental data has to vary as the inverse of local oxygen partial pressure. Figure 4 shows the result for two different mass transfer coefficients. It is clear that a more complex mass transfer model is

needed, probably accounting for intraparticle macropore convection due to the rapid pressure fluctuations.

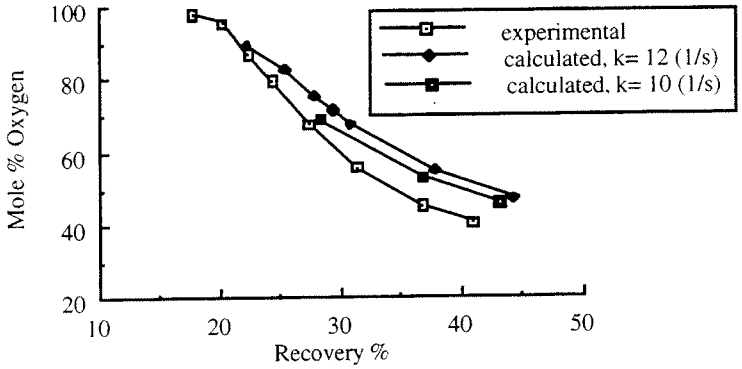


Figure 4. RPSA with 2.5 bar feed in a 1 m column with binderless 13X (350°C), cycle timing .5/1/4 s. Experimental data and model prediction.

6. CONCLUSION

The present experimental and simulation results show that the performance of RPSA can be improved at high purity by faster kinetics and at low purity by higher selectivity and capacity. Adsorbents with faster kinetics allow to decrease the cycle time or the column length to obtain the same product purity, and hence increase the productivity. Currently used models are shown to be inadequate to model the fast adsorbents as these are limited by combined macropore diffusion and convection resistances. The performance cannot be predicted with confidence by LDF models and significant efforts are needed to correctly model the operation under these conditions. By analysis of the equations and parameters in that case one can expect at least an order of magnitude increase in complexity and computer requirements. These improved models may prove to be the ultimate tool in designing high performance RPSA operations.

ACKNOWLEDGMENTS

The authors thank the Brussels Region for the financial support of this project .

REFERENCES

1. Basta, N., Ondrey, G. and Moone, S., Chem. Eng., Nov., 39 (1994).
2. Keller,G.E. and Jones, R.L., ACS Sym. Ser. No135, 275 (1980).
3. Keller, G.E., J. Separ. Proc. Technol., 2(3), 17-23 (1981).
4. Verelst, H & Baron G. V., Proc. World Congress Chem. Eng., Tokyo, Vol.2, 865, (1989).
5. Baron, G. V., Gas Sep. & Purif., 7, 111-117 (1993)
6. Yang, R.T., in D.M.LeVan(Ed.) Adsorption and Ion Exchange: Fundamentals and Applications, 1988, pp. 145-154.
7. Guan, J.Y. and Ye, Z.H., in M.Suzuki (Ed.), Fundamentals of Adsorption, Kodansha, Tokyo, 1993, pp. 243-250.
8. Alpay, E., Kenney, C.N. and Scott, D.M., Chem. Eng. Sci.,49, 3059, (1994)
9. Sircar, S., Abstract at the AIChE Annual Meeting (1994).
10. Villadsen, J.V. and Michelsen,M.L., Solution of Differential Equation Models by Polynomial Approximation, Prentice-Hall, Englewood Cliffs, NJ (1978).
11. Verelst, H. and Baron, G., J. of Chem. & Eng. Data, 30, 66-71, (1985).
12. NAG Library, Routine D02EFJ, Numerical Analysis Group, U.K.

The Model of Radionuclides Spreading on a Vast Territory*

V.L. Katkov, A.V. Bakulin, G.P. Kuznetsov, I.A. Nichiporovich

Institute of Engineering Cybernetics
6 Surganov Street, Minsk 220012, Republic of Byelarus

The objective of a project lies in the creation of a mathematical model for radionuclides spreading on a territory like Byelarus that will cover such factors as natural decay of radionuclides. washing pollutants off by precipitation (rain, melted snow), relief and type of an underlying surface, wind transfer of particles, impact of forests, lakes, and rivers. The availability of such model allows to solve the problem of a short- and a long-term forecast of radionuclides spreading. Another important goal is the evaluation of different variants of nuclear power station location to account the ecological safety of their operation including radionuclides disastrous ejection. In Byelarus a great amount of facts on the Chernobyl disaster results has been gathered. These results are of "a static" nature - only the state of things is recorded without attempts to give a quantitative forecast for the development of a situation in future. The forecast might be very significant for the solution of economic, social, agricultural, nature-protection and other problems related to the development of the situation on contaminated areas.

The same problems were studied in Russia, Ukraine, Sweden, International Agency on Atomic Energy and other countries and organizations [1-7]. We endeavoured to regard those results while designing our integral system.

The simulation carries out during 1-2 years (short-term forecast) and 4-5 years (long-term forecast). The natural decay of radionuclides is computed through known formulae for each elementary cell of the territory. Long-living particles being mostly dangerous are considered, they are Cesium-137, Strontium-90, Plutonium-238, ..., -241.

The wind transfer of particles is simulated via two ways: by the solution of the transfer and diffusion equation, and by the simulation of a random process of the fallout. For this purpose we used the climatic observations' data as well as the actual information obtained during everyday meteoobservations for the analyzed time interval.

To simulate washing radionuclides off, some schemes are proposed and tested that take into account the intensity of atmospheric precipitation, snow melting speed, and earth surface type. Since the mechanism of such phenomena has been studied insufficient, it has constructed a series of empirical formulae based on the data of meteoobservations, and has performed a random process of particles' transfer and diffusion.

Main difficulties while constructing the model were related to the development of random process describing wind transfer and particles washing off. The model has the block of preliminary processing of meteoobservations to get the necessary statistic characteristics, and,

* The research described in this publication was made possible in party by Grant No. MW9000 from the International Science Foundation.

first of all, distribution functions for appropriate random quantities. The program has written on C and FORTRAN and operates under MS-DOS for on IBM compatible computers.

The preliminary experiments show the model satisfactory describes physical process analyzed with using Chernobyl disaster data as test.

REFERENCES

1. Yu.A. Izrael, V.N. Petrov, D.A. Severov, Modelling the Radioactive Fallout Near the Chernobyl Nuclear Power Station, Meteorology & Hydrology, 2 (1987) (in Russian).
2. Ya.A. Izrael, V.N. Petrov, S.I. Avdyushin et al., Radioactive Contamination of the Environment Around the Chernobyl Nuclear Power Station, Meteorology & Hydrology, 2 (1987) (in Russian).
3. Yu.S. Sedunov, V.A. Borzilov, N.V. Klepikova et al., Physicomathematical Modelling of the Regional Transport of Radioactive Pollutants in the Atmosphere in Consequence of the Chernobyl Accident, Meteorology & Hydrology, 9 (1989) (in Russian).
4. M.V. Buikov, E.K. Garger, and N.N. Talerko, Research into the Formation of Spotted Pattern of Radioactive Fallout with the Lagrangian-Eulerian Model, Meteorology & Hydrology, 12 (1992) (in Russian).
5. Ch. Person et al., The Chernobyl Accident. A Meteorological Analysis of How Radionuclides Reached Sweden, SWHI Rep. Meteor. and Climatol., 1986.
6. A. Albergel, D. Martin, B. Strauss, and J.-M. Gross, The Chernobyl Accident: Modelling of Dispersion over Europe of the Radioactive Plume and Comparison with Air Activity Measurements, Atm. Env., 1988, vol. 22.
7. International Agency on Atomic Energy, General Conference, Vienna, August 30 - September 5, Report GS (SPL. 1)/3 24.09.86.

COMPARISON OF THEORETICAL AND IDENTIFIED MODELS FOR CONTROL OF A PILOT PLANT DISTILLATION COLUMN

R. Lamanna, F. Aponte and M.I. Campos.

Dpto. de Procesos y Sistemas. Universidad Simón Bolivar.
Apdo. 89000. Caracas 1081-A. Venezuela.e-mail: lamanna@usb.ve

ABSTRACT

The modelling and simulation of a pilot plant binary distillation column is presented. A theoretical, dynamic model of the pilot plant is formulated, based on non-stationnary mass balances, tray hydraulic equations and equilibrium relationships. Input-output models of the plant are also obtained using classical identification techniques. A comparative analysis of the simulation results of the theoretical and the identified models is presented, which discusses not only the precision aspects but the implementation limitations of the two approaches.

1. INTRODUCTION

Mathematical models as dynamic simulators of industrial processes are essential in automation and control design.

Distillation theoretical models, based on comprehensive sets of dynamic heat and mass balance relationships, are often quite complex due to the complex nature of the operation itself. However, if they are carefully constructed, simulators based on this kind of models are very useful tools for they represent precisely the dynamic operation of the plant [2,6].

The final objective of this research is the implantation of some model based control atrategy on a real distillation column plant. The controller uses a simulator (model) to predict future outputs of the plant needed to compute the present control signal. In fact this computation is based on the minimization of a cost function depending on future control errors [1,3].

The important step of modeling the plant is treated in this paper, where we present the results of a theoretical model and those of a linear input-output model obtained by means of a parametric identification algorithm [4,5,7].

2. THE PILOT PLANT

The process under consideration is an eleven stage pilot plant column, installed at the Simón Bolívar University, Caracas (see figure 1). It separates a 40 wt. % methanol-water mixture which is introduced into the column at a rate 30 liter/hour on the fifth tray.

The feed stream is fed as a saturated liquid (at its bubble point), at a rate F (lt/hr), with composition X_f (mole fraction more-volatile component).The overhead vapor is totally condensed, and retained into a reflux tank, with liquid hold-up M_d (lt). Reflux is pumped back to the top tray at a rate L_o (lt/hr). Overhead distillate product is removed at a rate D (lt/hr), with composition X_d. At the base of the column, the bottom product is pumped out at a rate B (lt/hr), with composition X_b. Vapor boilup is generated in a reboiler at a rate V. The composition of the boilup is Y_b and it is supposed to be in equilibrium with the liquid at X_b.

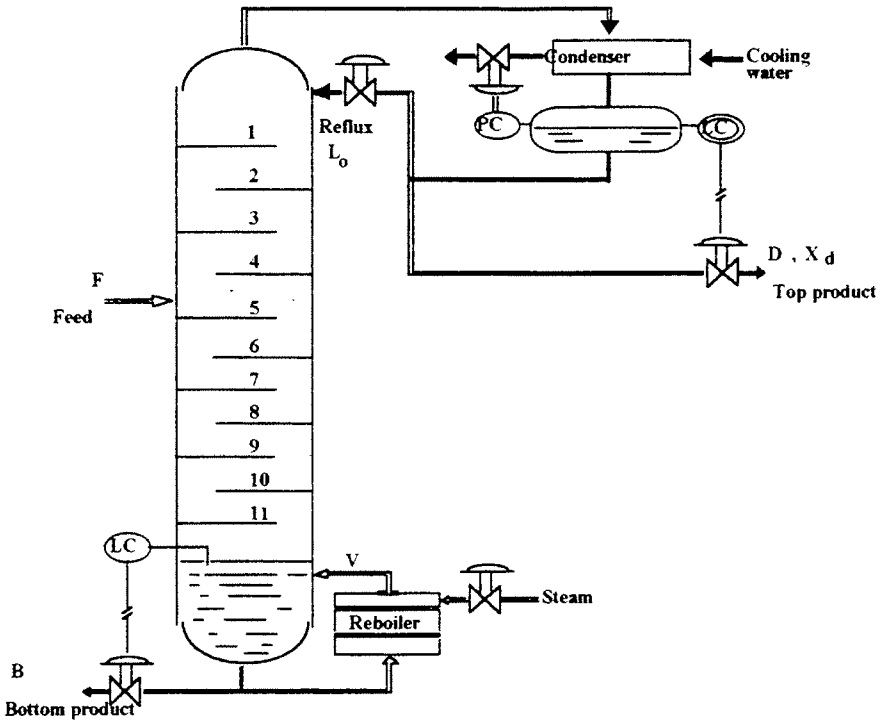


Figure 1.Schematic of pilot plant distillation column.

3. THE THEORETICAL MODEL

A simulation program based on a phenomenological model of the distillation plant is developed. The program consists of two modules.

The first module is the initial calculus of the number of theoretical trays, global efficiency, minimum reflux ratio and steady-state operating conditions of the plant, given the feed-stream properties and products specifications.

The computation is based on the separation degree desired. The program uses the Fenske-Underwood-Gilliland method to compute the number of theoretical trays needed; and the Kirkbride correlation to determine the optimum feed tray [2].

The second module contains the recursive integration of the differential equations that describe the dynamic behaviour of the system, taking account of the following simplifications:

- the vapor leaving each tray is in equilibrium with the liquid on the tray,
- the tower is at constant pressure,
- the liquid hold-up on each tray is perfectly mixed and incompressible,
- heat losses and temperature changes up the column are negligible,
- the vapor hold-up on the trays is negligible,
- the dynamics of the condenser and the reboiler are much faster than that of the column itself, and are therefore negligible.

With these assumptions, the dynamic model can be written as:

<u>for the n-th tray:</u>

$$\frac{dM_n}{dt} = L_{n-1} - L_n = u\frac{dL_n}{dt} \qquad (1)$$

$$\frac{dM_n X_n}{dt} = VY_{n+1} + L_{n-1}X_{n-1} - VY_n - L_n X_n \quad (2)$$

for the feed-tray:

$$u\frac{dLn}{dt} = F + L_{n-1} - L_n \quad (3)$$

$$\frac{dM_n X_n}{dt} = VY_{n+1} + L_{n-1}X_{n-1} - VY_n - L_n X_n + FX_f \quad (4)$$

for the reflux drum:

$$\frac{dMd}{dt} = V - L_0 - D \quad (5)$$

$$\frac{dM_d X_d}{dt} = VY_0 - L_0 X_d - DX_d \quad (6)$$

for the reboiler:

$$\frac{dMb}{dt} = L_n - B - V \quad (7)$$

$$\frac{dM_b X_b}{dt} = L_n X_n - L_b X_b - VY_b \quad (8)$$

where:
Mn : total liquid hold-up on the n-th tray
X_n : light-component composition in the liquid phase
V : Vapor flow-rate
Y_n: light-component composition in the vapor phase
L_n : output liquid flow-rate from the n-th tray
u : hydraulic coefficient depending on the tray hydraulics [6].

The compositions in each equilibrium phase are related by:

$$Y_n = K_n X_n \quad (9)$$

where the equilibrium coefficient K_n is computed as:

$$K_n = \frac{\gamma_n \phi_n^{sat} P^{sat}}{\phi_n P} \quad (10)$$

and:
γ_n : activity coefficient in the liquid phase
ϕ_n^{sat} : fugacity coefficient of the light component at saturation conditions
P^{sat} : saturation pressure
ϕ_n : fugacity coefficient of the solution
P : tray total pressure.

The heat input to the reboiler, Q, determines the vapor boil-up in the column, V, as $V=Q/\lambda$ where λ is the latent heat of vaporization, and is obtained in terms of the enthalpy of the gaseous and liquid methanol-water solutions [2].

To complete the dynamic model we use two equations representing the proportional level controllers on the column base and reflux drum (see figure 1), and we fix the reflux ratio L_O.

The mathematical model is solved via numerical integration to simulate the time responses of the plant, under perturbations in F (the feed-stream fow-rate to the column) and/or in Q (the heat input to the reboiler).

Validation of the theoretical model is carried out by comparison with the behaviour of the real system in some perturbation situations. During the experiments, the pilot plant was operated under the initial steady state conditions listed in table 1.

Table 1. Operating conditions.

	Top Product	Bottom Product	Feed
Composition (w %)	0.94	0.30	0.41
Temperature (K)	322	375	313
Flow-rate (l/h)	14.0	16.0	30.0

Figure 2 shows the behaviour of the theoretical model developed (dotted lines) and that of the real plant (continuos lines) for the top product composition and flow-rate under pseudo-random perturbations on the input variable Q. Figure 3 corresponds to the responses under a step change in the feed flow-rate F to the plant.

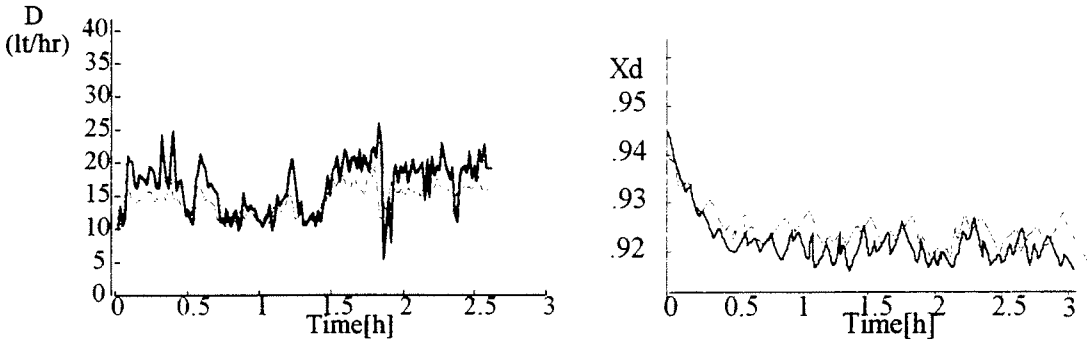


Figure 2. Plant and model outputs D and X_d under pseudo-random input signals on Q.

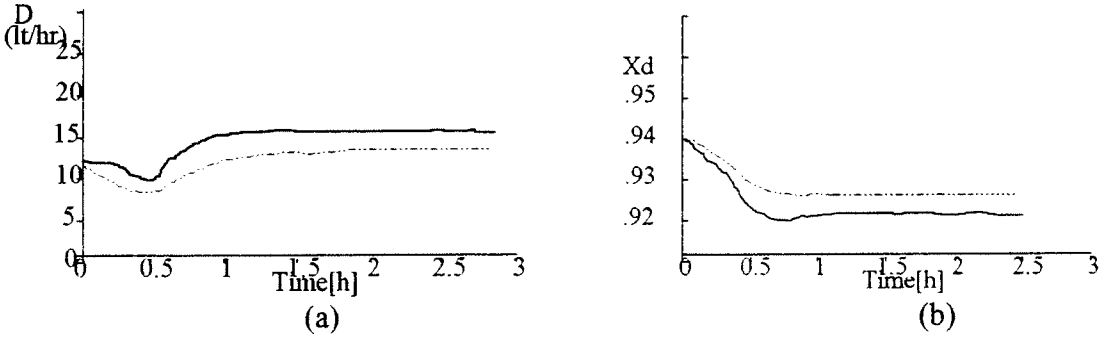


Figure 3. Plant and model outputs D and X_d under a step change in the feed flow-rate F.

4. RESULTS OF CLASSICAL RECURSIVE IDENTIFICATION

An alternative for the destillation plant modelling is the system identification, which relays on the effective processing of noisy and imprecise input-output operating data of the plant. A classical identification approach consists in estimating the coefficients of a linear parametric input-output model using recursive algorithms based on the leasts squares or similar methods [4,5,7]. This is a well known solution, that produces an analytical mathematical model which can be explicitly or implicitly used in the process controller design, even in on-line automation strategies.

The operating input-output data of the plant needed for the identification is obtained monitoring the column top product (flow-rate D and composition X_d) under perturbations in the feed flow-rate F, and the heat flow-rate to the reboiler Q.

Pseudo-random ternary sequences are used in the input variables, with amplitudes of 2%, 5% and 10% around the steady state operating point of the plant (see Table 1). Careful treatment is needed in the design of the input sequences, selecting amplitudes that will not drive the column away from its operationg condition, and producing at the same time a sufficiently rich information. It is needed as well the filtering and normalizing of the output operating records, to make them useful for the recursive identification or for the neural network training.

The two inputs are varied simultaneusly to collet the output variables, which are recorded every 60 seconds, over total periods of 2.5 hours.

A linear, auto-regressive input-output model (ARX model) is used to represent the process:

$$\hat{\underline{y}}(k) = \sum_{i=1}^{nb} \hat{B}_i \, q^{-i} \, \underline{u}(k) - \sum_{i=1}^{na} \hat{A}_i \, q^{i} \, \underline{y}(k) \qquad (11)$$

where:

$\underline{y}^T = [\, D \,,\, X_d \,]$,

$\underline{u}^T = [\, F \,,\, Q \,]$,

$\hat{\underline{y}}(k)$: vector of estimates of the outputs at time k, of dimension [2 x 1].

The System Identification Toolbox of Matlab 4.2 is used to carry out the iterative identification procedures, by means of the well known recursive least squares identification algorithm. Considerable effort has to be dedicated to the testing of different model structures, to determine the better set of structural indices (we finnaly selected n_a=2 and n_b=1).

Figure 4 shows the behavior of the ARX multivariable model (dotted lines) and that of the real plant (continuos lines) for the top product temperature and flow-rate under pseudo-random perturbations on the input variable Q. Figure 5 presents the responses under a step change in the feed flow-rate F to the plant.

5. CONCLUSIONS

A phenomenological model, and a linear, auto-regressive input-output model (ARX model), are used to represent the dynamics of a pilot plant distillation column.

The linear ARX model obtained via recursive identification emulates the plant more precisely than the theoretical model, in the neighborhood of the operating point considered. It works quite well even considering the real plant noisy environment.

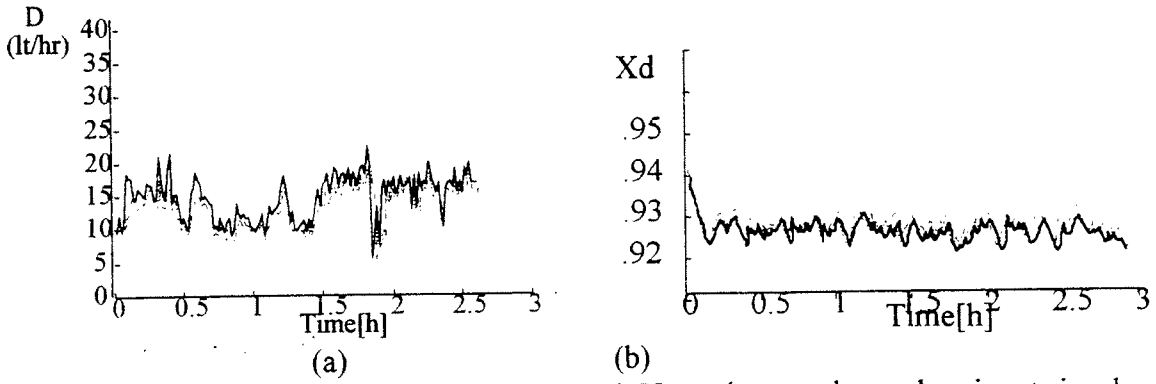


Figure 4. Plant and ARX model outputs D and X_d under pseudo-random input signals on Q.

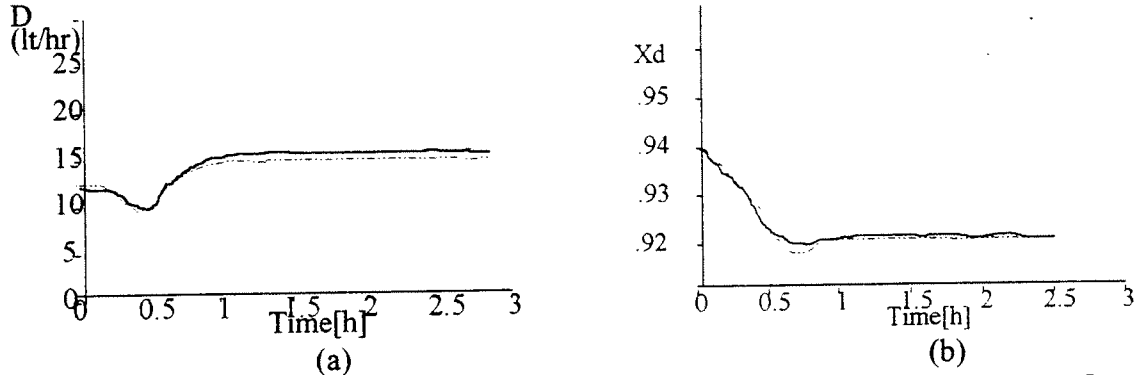


Figure 5. Plant and ARX model outputs D and X_d under a step change in the feed flow-rate F.

The theoretical model should be more general because it predicts the operation of the plant even in new operating conditions. However, it shows steady-state deviations in several cases. These modelling errors are due to the imprecise measurements of some physical constants, and also because the mathematical equations do not take account of real existing perturbations. On the other hand, the alternative of using an identified input-output model is very interesting because, even considering the needed filtering of noise and the adquisition and normalizing of the data, it can be more easily obtained than a good theoretical model.

Nevertheless, we have to anticipate the lack of precision of the ARX model for simulation of the column in other operating points, due to the non-linearities of the plant. A good overall automation strategy should take account of this situation and use perharps non-linear identified models (of the neural networks type, for instance).

REFERENCES

1. Ästrom, K. y Wittenmark B., *Computer Controlled Systems: Theory and Design*, Prentice-Hall. 1990.
2. Franks R., *Modeling and Simulation in Chemical Engineering*, New York: Wiley-Interscience. 1972.
3. Goodwin G. y Sin K., *Adaptive Filtering, Prediction and Control*, Prentice Hall, 1984.
4. Ljung, L.: *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs. 1987.
5. Ljung, L. and T. Söderström. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, MA.,1983.
6. Luyben, W.: *Practical Distillation Control*, Van Nostrand Reinholds, New York, 1992.
7. Söderström, T. and P. Stoica. *System Identification*. Prentice Hall, Englewood Cliffs, NJ.,1989.

Analysis of dynamical and structural properties of a microbial landfill ecosystem by means of computer simulation

R. Böttner and B. Schleusener

Zentrum für Umweltwissenschaften, Universität Potsdam
Rudower Chaussee 5, (WITEGA, Geb. 4.1), D-12489 Berlin, Germany

After a careful system analysis of the community structure of the involved multiple, interacting microbial populations and of the abiotic physico-chemical conditions and influences on the microbial landfill ecosystem, a mathematical model is developed. The model consists of a set of nonlinear ordinary differential equations, derived on the basis of mass balances, microbial growth models and stochiometric equations. System parameters are extracted from the literature, the equations are solved numerically with MATLAB and SIMULINK and the model was qualitatively validated and calibrated using data from experimental measurements.
By means of simulation experiments dynamical properties (stationary states, stability, parameter sensitivity) of the microbial ecosystem and basic principles refering to the importance of microbial communities in biodegradation have been investigated.

1. INTRODUCTION

Microbial ecosystems are essential parts of global natural element cycles and mass and energy flows and are responsible for the dynamic equilibrium of synthesis and degradation in the interplay of living organisms and abiotic environment on earth. On the other hand, due to their low extension and local restrictedness, partly in very small ecological niches and microenvironments, microbial ecosystems are well suited to study fundamental principles of structure, function and interaction between organisms and their abiotic surroundings.

Microrganisms have direct significance for waste and contaminated sites management, because microorganisms are the active part of many processes of waste transformation and decomposition of organic and anorganic substrat mixtures. In the first phase of a landfill development, the „intensive reactorphase", subtle interactions of different microorganisms with one another and with physical, chemical (waste composition) and meteorological factors constitute the driving force, substantially determining, which ecological risk and environmental impact originate from hazardous substance delivery as leachate or gaseous emission in the short or long time horizon.

The presented model of microbial-ecological processes in landfills is the first step in the ongoing work of the authors towards the construction of a comprehensive landfill model, including also the processes of leaching and transport of hazardous chemical substances in the subsurface and the surrounding air.

2. STRUCTURE OF THE LANDFILL ECOSYSTEM

Soon after the initial placement of waste, the main part of the landfill becomes anaerobic [STEGMANN, EHRIG 1980]. An anaerobic association of bacteria, the so called methanogenic syntropy, starts the degradation of solid organic carbon to carbon dioxide (CO_2) and methane (CH_4). Figure 1 shows the most important interactions between the participating bacterial groups, the substrates and intermediate products.

The solids (c_1) consisting of carbohydrates, lipids, and nitrogenous material are hydrolyzed by extracellular enzymes (e_0), produced by a group of bacteria (b_1^f), to build lower molecular dissolved organic solids [MCINERNEY, BRYANT 1983]. These products are the substrate for the same bacteria which produce volatile organic acids and alcohols (c_3), CO_2 (c_4), H_2 (c_5) and acetic acid (c_6).

Acetogenic bacteria (b_2^a) consume the alcohols and carboxylic acids (C>2) of the hydrolyzers and acid formers (b_1^f) and transform them to acetic acid (c_6), CO_2 (c_4) and H_2 (c_6). The only way to perform this transformation is to have hydrogen-uptaking bacteria close by. The hydrogenic methanogens (b_3) are doing this work and produce methane (c_7) from CO_2 and H_2. Hydrogen uptake also increases the energy yield of fermentative bacteria, which leads to more fermentative products, more exoenzymes and consequently to more hydrolysis of biopolymers. The acetophilic methanogens (b_4^{ma}) convert acetic acid (c_6) to methane (c_7) and carbon dioxide (c_4).

It is conceivable that such an association of bacteria, linked by mutualistic and commensal relationships is a very tight and stable one. Methanogenic bacteria are strict anaerobes, grow slowly in comparison to the fermentative and acetogenic bacteria and can be inhibited by large concentrations of the volatile acids, their substrate.

3. MATHEMATICAL MODEL OF THE LANDFILL ECOSYSTEM

The growth, substrate utilization and product formation of microorganisms can be modelled by mathematical equations. Starting point are the following quantities [BERGTER, 1983]:
x - microbial biomass (concentration) [mg COD/L], s - limiting substrat [mg COD/L]

d_X - death rate of the bacterium x [1/DAY], $\frac{dx}{dt}$ - growth rate [mg COD/L*DAY]

$\frac{dx}{dt}\frac{1}{x} = v$ - specific growth rate [1/DAY].

For the description of the dynamics of the bacterium x one gets :

$\frac{dx_i}{dt} = v(s)\,x_i - d_{X_i}\,x_i$. The dependence of the specific grow rate v(s) from the limiting substrate concentration is of Monod type : $v(s) = v_{max}\frac{s}{K_s + s}$.

v_{max}-maximum specific growth rate [1/DAY], K_s- half saturation constant [mg COD/L].

It is assumed, that the bacterial growth rate is proportional to the substrate utilization rate :

$\frac{dx}{dt} = Y_{x/s}\frac{ds}{dt}$. The factor $Y_{x/s} = \frac{dx/dt}{ds/dt} = \frac{dx}{ds}$ is the stochiometric yield coefficient.

The substrate utilization rate amounts to :

$$\frac{ds}{dt} = -\frac{1}{Y_{x/s}}\frac{dx}{dt} = -\frac{1}{Y_{x/s}}v_{max}\frac{s}{K_s + s}x. \qquad (1)$$

For the description of the bacterial product formation p (mg COD/L) we use the equation :

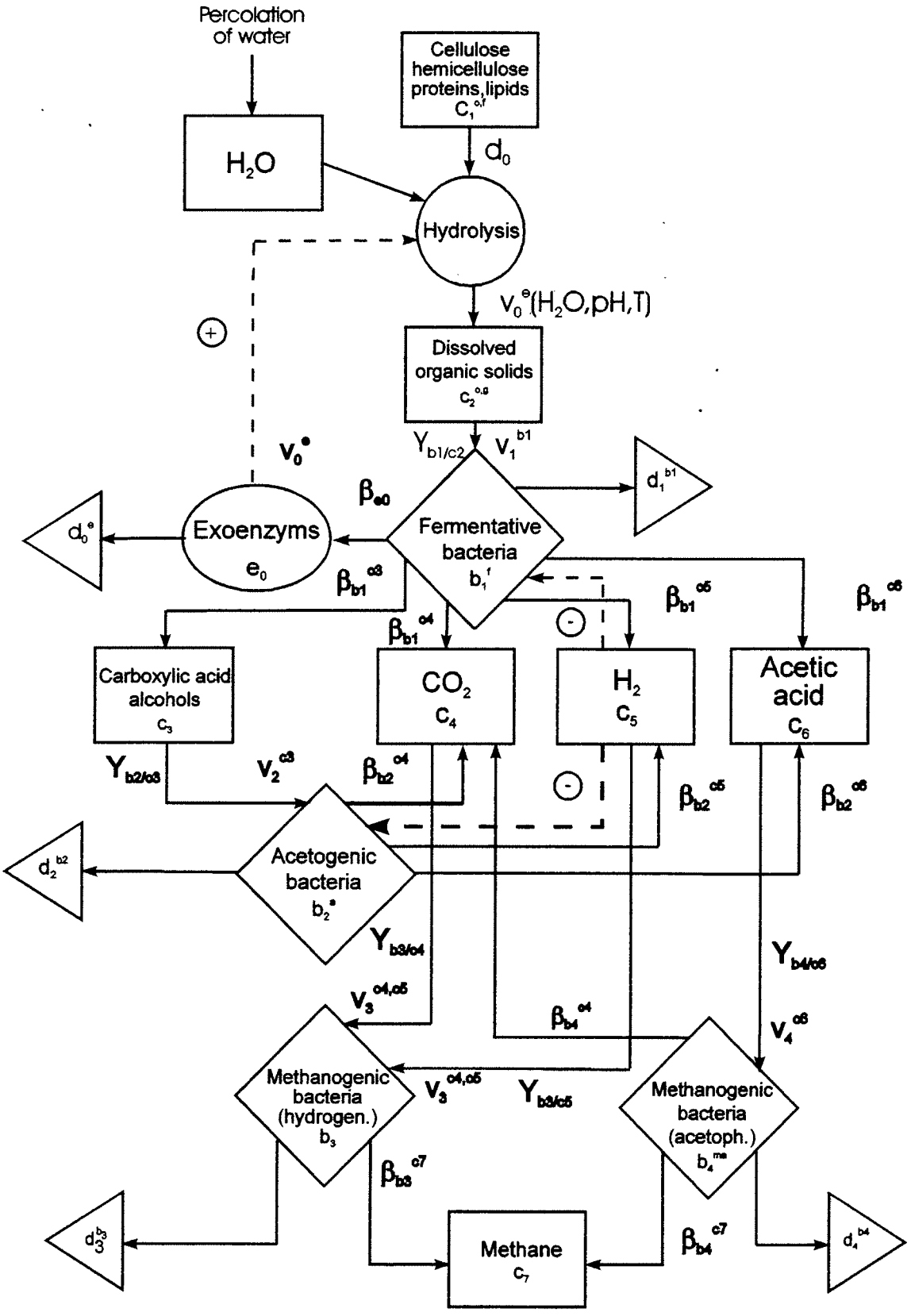


FIG. 1 FOODWEB OF THE LANDFILL ECOSYSTEM

$$\frac{dp}{dt} = \alpha \frac{dx}{dt} + \beta x - d_p p \qquad (2)$$

d_p- decay constant of p [1/DAY], [BU'LOCK, KRISTIANSEN 1989].

The bacterial growth rate can be controlled not only by limiting substrates in the Monod type form, but also by other inhibitory or stimulating substances and conditions with different functional shapes. The following modification of the Monod type control characteristic allows a flexible modelling of various effects of the factor s :

$$v(s) = v_{max} \frac{s^n}{K_s + s^m} \qquad (3)$$

n=0, m>0 yields an inhibitory control characteristic, whereas n≥1, m>n results in a substrate inhibition type control characteristic (Fig. 2). Using the described building blocks, the complex foodweb of the considered microbial community (Fig.1) is represented by a set of 12 nonlinear ordinary differential equations, two of them are given as examples (denotation as in Fig. 1).

$$\frac{dc_2}{dt} = d_0 c_1 + e_0 v_0^e \frac{c_1}{K_{c_1}^e + c_1} - \frac{1}{Y_{b_1/c_2}} v_1^{c_2} \frac{c_2}{K_{c_2}^{b_1} + c_2} b_1^f \frac{KI_{c_5}^{b_1}}{KI_{c_5}^{b_1} + c_5} - d_{c_2} c_2 \qquad (4)$$

$$\frac{db_1^f}{dt} = v_1^{c_2} \frac{c_2}{K_{c_2}^{b_1} + c_2} b_1^f \frac{KI_{c_5}^{b_1}}{KI_{c_5}^{b_1} + c_5} - d_1^{b_1} b_1^f \qquad (5)$$

4. SIMULATION RESULTS

The dynamical properties of the derived system of differential equations are analyzed by simulation experiments with MATLAB and SIMULINK. Parameters are taken from the literature [EASTMAN, FERGUSON 1981; LAWRENCE, McCARTY 1969] and the results are validated with experimental data [EHRIG, SCHEELHAASE 1993].

A typical time course of sytem variables is shown in Fig. 3 and 4 for normal function of the landfill ecosystem. Characteristic oscillations in the early phase (1-3 years) of landfill development are also present in the experimental studies (EHRIG, SCHEELHAASE 1993). Fitting closely to this transition phase, the system uses the biodegradable organic substances until the supply is exhausted (after about 25 years).

Quite different proceeds the ecosystem development if the population of hydrogenophilic methanogens is not present in the deposited waste or does not found favourable conditions for an unrestrained growth. Figs. 5 and 6 reflect this situation of disturbed biodegradation. Organic acid concentration deteriorates in about fifty days and the hydrogen concentration increases up to the level, which leads to the inhibition of acetogenic and fermentative bacteria.

Several parameter values, which are used in the simulation experiments, are only estimates. Therefore the model predictions are expected semiquantitative.The data insufficiency results from troublesome experimentation due to extreme sensitivity of methanogenic bacteria to oxygen. Considering the situation in an other way, there is good reason for modelling and simulation studies in order to get knowledge about parameter sensitivities.

FIG. 2 SUBSTRATE-INHIBITION CONTROL

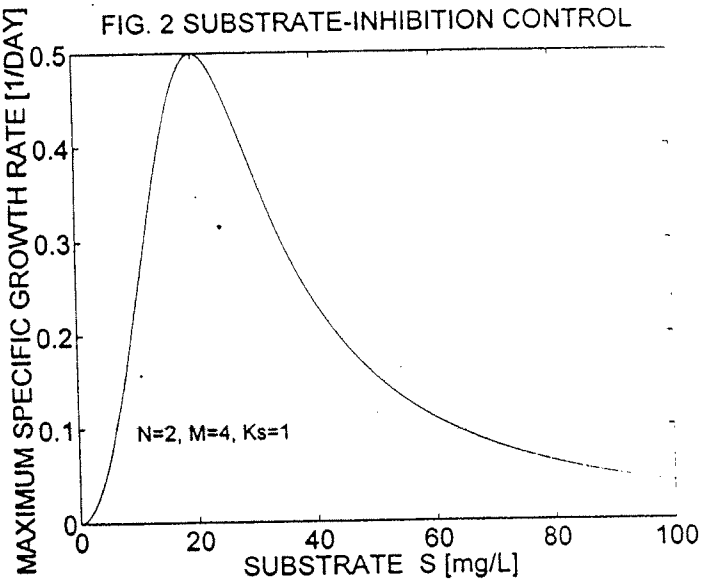


FIG. 3 NORMAL FUNCTION OF THE ECOSYSTEM

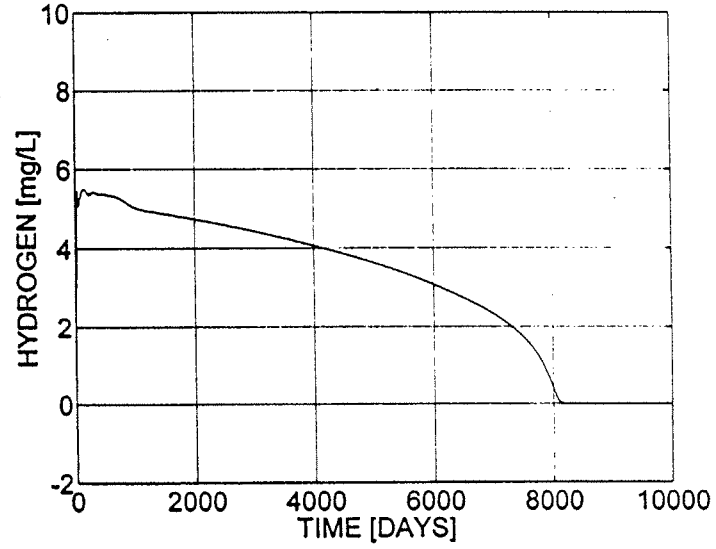


FIG. 4 NORMAL FUNCTION OF THE ECOSYSTEM

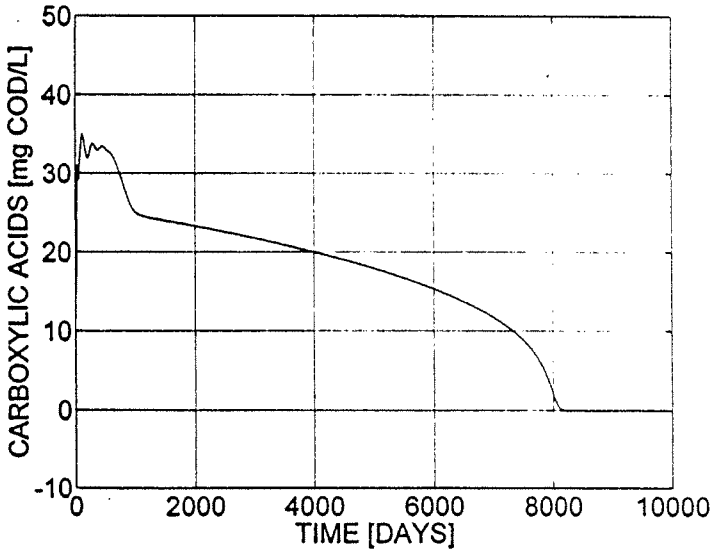


FIG. 5 WEB WITHOUT HYDROGEN CONSUMER

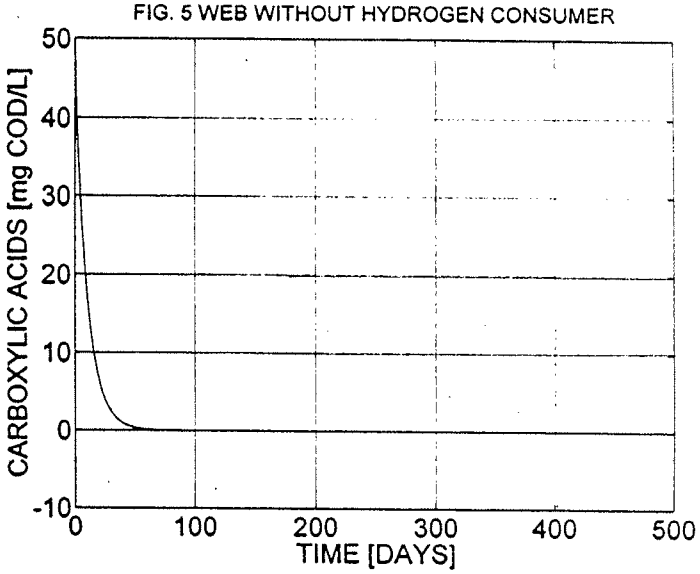
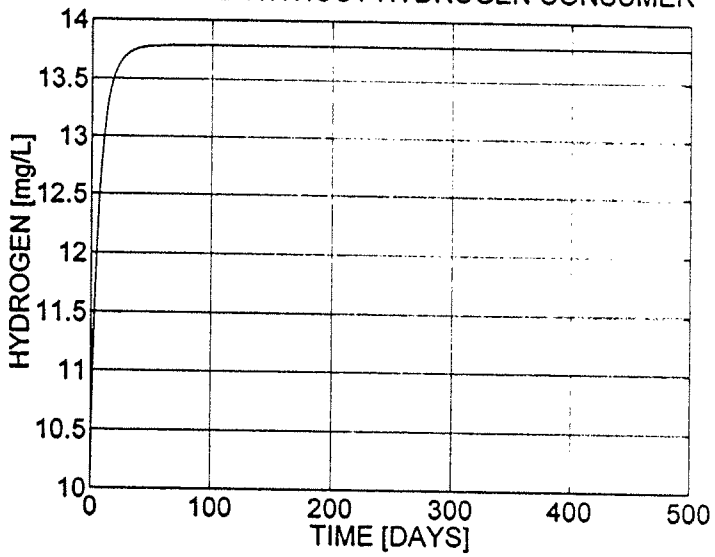


FIG. 6 WEB WITHOUT HYDROGEN CONSUMER



REFERENCES

BERGTER, F. (1983) : Wachstum von Mikroorganismen. Experimente und Modelle. Fischer Verlag Jena.

BU'LOCK, J.; KRISTIANSEN, B. (1987) : BASIC Biotechnology. Academic Press London New York 1987.

EASTMAN, J.A.; FERGUSON, J.F. (1981) : J. Water Pollut. Control Fed., **53**, 352.

EHRIG, H.J.;SCHEELHAASE, T. (1993) : Pollution Potential and Long Term Behaviour of Sanitary Landfills. In : Proceedings Sardinia 93, Fourth International Landfill Symposium, CISA, Cagliary, Italy, pp 1203-1225.

LAWRENCE, A.W.;McCARTY, P.L. (1969) : J. Water Pollut. Control Fed., **41**, R1.

MCINERNEY, M.J.; BRYANT, M.P. (1983) : Review of methane fermentation fundamentals. In : Wise, D.L.(ed.) CRC Press, Boca Raton, Florida.

STEGMANN, R.; EHRIG, H.J. (1980) : Entstehung von Gas und Sickerwasser aus geordneten Deponien-Möglichkeiten der Beeinflussung. Müll und Abfall, Heft 2.

An application of the bifurcation analysis for determining the conditions for the bursting emergence in the stomatogastric ganglion neuron model

O.E. Dick[a] and Y.A. Bedrov[b] and G.N. Akoev[c] and K.-D. Kniffki[d]

[a]Physiology of Reception Department, Pavlov Institute of Physiology, nab. Makarova 6, 199034, St. Petersburg, Russia

[b]Applied Mathematics Department, Pavlov Institute of Physiology,

[c]Physiology of Reception Department, Pavlov Institute of Physiology,

[d]Physiology Institute University of Würzburg, 97070 Würzburg, Germany

We examine the problem of determining the conditions for the emergence of bursting oscillations in the stomatogastric ganglion neuron model (Guckenheimer et al., 1993). Taking into account that the presence of slow oscillations of intracellular calcium concentration is a necessary condition for the bursting existence in the neuronal membranes, we use the conditions for the slow oscillations emergence instead of the conditions for bursting. The conditions for the slow oscillations in the full system are determined by bifurcation values corresponding to the Hopf bifurcation in the slow subsystem. We give the explicit parametric forms of the Hopf bifurcation curves on the two parameter planes. The presence of these forms enables us to examine the question of the number of the bifurcation curves on each of the chosen planes. The problem is reduced to finding the number of discontinuities of the forms obtained. We show that there is only one bifurcation curve on each of the planes. This is an argument in favor of an assumption that the bursting region is simply connected. The curves constructed determine the parameter values at which in the full system either bursting oscillations or slow oscillations exist. Using numerical integration we obtain the approximate boundary of the bursting region.

1. INTRODUCTION

The stomatogastric neuron is a component of the neural circuit generating the pyloric rhythm in the lobster stomach [Selverston *et al.*, 1976]. This cell is a conditional burster. It means that in the absence of neuromodulators and blockers the neuronal membrane is in the rest state or generates periodic spike activity. Bursts (complex periodic oscillations separated by periods of quiescence) can be triggered by different neuromodulators that change maximal ionic conductances [Harris-Warrick & Johnson, 1987].

The purpose of the present work is to examine the problem of determining the conditions for the emergence of bursting oscillations in the stomatogastric neuron model. We use the fact that for the neuronal membranes the necessary condition for the bursting mode is the presence of slow oscillations of concentration of intracellular calcium. It enables

us to use the conditions for the emergence of slow oscillations instead of the conditions for bursting. That is why instead of the full system (given in Section 2) we analyze the subsystem of the second order, dynamics of which corresponds to the slow component of bursting. The Hopf bifurcation values in the slow subsystem determine the conditions for the emergence of slow oscillations in the full system. In Section 3 we demonstrate the possibility to obtain the explicit parametric forms of the Hopf bifurcation curves on the $(I, \bar{g}_{KCa})$, $(I, \bar{g}_{Ca})$ planes, where I is a stimulus value, $\bar{g}_{KCa}$, $\bar{g}_{Ca}$ are the calcium-dependent potassium and calcium potassium conductances. Analyzing these forms we see that the bifurcation curves can have discontinuities of the first kind. The number of the discontinuities determines the number of the bifurcation curves on each of the chosen parameter planes. In Section 4 we apply the constructed bifurcation curves for determining the region of the bursting existence.

2. THE MODEL AND THE SLOW SUBSYSTEM

The full system describing the activity of the stomatogastric ganglion neuron consists of six differential equations [Guckenheimer *et al.*, 1993]:

$$
\begin{aligned}
&\dot{V} = F_1(V, h, n, h_a, z, Ca \mid \bar{g}_{KCa}, \bar{g}_{Ca}, I) \\
&= I - \bar{g}_{Na} m_\infty^3(V) h (V - V_{Na}) - \bar{g}_K n^4 (V - V_K) - \bar{g}_L (V - V_L) \\
&- \bar{g}_a m_{a\infty}^3(V) h_a (V - V_K) - \bar{g}_{Ca}\, z\, \tfrac{V - V_{Ca}}{0.5 + Ca} - \bar{g}_{KCa}\, Ca\, \tfrac{V - V_K}{0.5 + Ca}, \\
&\dot{h} = F_2(V, h) = 0.8\, \tfrac{h_\infty(V) - h}{\tau_h(V)}, \\
&\dot{n} = F_3(V, n) = 0.8\, \tfrac{n_\infty(V) - n}{\tau_n(V)}, \\
&\dot{h}_a = F_4(V, h_a) = h_{a\infty}(V) - h_a, \\
&\dot{z} = F_5(V, z) = \tfrac{z_\infty(V) - z}{23.5}, \\
&\dot{Ca} = F_6(V, z, Ca) = -\tfrac{1}{2} p\, k_{Ca}\, z\, \tfrac{V - V_{Ca}}{0.5 + Ca} - p\, Ca,
\end{aligned}
\tag{1}
$$

where V is the membrane potential, h and h_a are the variables related to the probabilities of inactivation of sodium and fast potassium channels, n and z are the variables related to the probabilities of activation of delayed rectifier potassium and calcium channels, respectively; $m_\infty(V)$, $m_{a\infty}(V)$, $h_{a\infty}(V)$, $z_\infty(V)$, $\tau_h(V)$ and $\tau_n(V)$ are given functions of V; Ca is the intracellular calcium concentration. The constants $\bar{g}_{Na} = 15[\mu S]$, $\bar{g}_K = 8[\mu S]$, $\bar{g}_a = 77.95[\mu S]$, $\bar{g}_L = 0.0854[\mu S]$ are the maximal sodium, delayed and transient potassium and leakage conductances; $V_{Na} = 30[mV]$, $V_K = -75[mV]$, $V_{Ca} = 140[mV]$, $V_L = -40[mV]$ are the reversal potentials for corresponding ions; $p = 0.003[ms^{-1}]$, $k_{Ca} = 0.0078[mV^{-1}]$ is the rate constant for the removal of intracellular calcium.

The slow subsystem is described by two differential equations and one algebraic equation:

$$
\begin{aligned}
&\dot{z} = F_5(V, z), \\
&\dot{Ca} = F_6(V, z, Ca), \\
&0 = F_1(V, z, Ca \mid \bar{g}_{KCa}, \bar{g}_{Ca}, I)
\end{aligned}
\tag{2}
$$

Integrating systems (1) and (2) we found a good agreement in the frequency and amplitude of the slow oscillations of the full system and oscillations of the reduced system.

3. THE HOPF AND SADDLE-NODE BIFURCATION CURVES FOR THE SLOW SUBSYSTEM

Let $J(z, Ca \mid \bar{g}_{KCa}, \bar{g}_{Ca})$ be the Jacobian matrix corresponding to the two first equations of system (2) and let $\lambda^2 + a_1\lambda + a_2$ be its characteristic polynomial with $a_1 = -trace\, J$ and $a_2 = det\, J$. A necessary condition for a Hopf bifurcation is that the roots of the characteristic polynomial at a stationary point have the form

$$\lambda_1 = i\omega \quad \text{and} \quad \lambda_2 = -i\omega \quad (\omega > 0).$$

Hence, the coefficients satisfy

$$a_1 = 0, \quad a_2 = \omega^2 > 0. \tag{3}$$

Then, bifurcation parameter values for a Hopf bifurcation in the slow subsystem are determined by

$$\begin{aligned} &F_5(V, z) = 0, \\ &F_6(V, z, Ca) = 0, \\ &F_1(V, z, Ca \mid \bar{g}_{KCa}, \bar{g}_{Ca}, I) = 0, \\ &trace\ J(z, Ca \mid \bar{g}_{KCa}, \bar{g}_{Ca}) = 0, \\ &det\ J(z, Ca \mid \bar{g}_{KCa}, \bar{g}_{Ca}) > 0. \end{aligned} \tag{4}$$

Let $z_\infty(V)$ be a solution of the first equation of system (4) with respect to z and $Ca_\infty(V) > 0$ be a solution of the second equation of (4) with respect to Ca for a fixed value V.

Whence we obtain that the bifurcation curve on each of the planes $(I, \bar{g}_{KCa})$, $(I, \bar{g}_{Ca})$ can be given by the explicit parametric form

$$\begin{aligned} &\bar{g}_i = \psi(V) = -\tfrac{B_{11}(V)+B_{22}(V)}{A_{11}(V)+A_{22}(V)}, \\ &I = \varphi(V), \\ &\psi(V)^2 C_1 + \psi(V) C_2 + C_3 > 0, \end{aligned} \tag{5}$$

$i = 1, 2$, $\bar{g}_1 = \bar{g}_{KCa})$, $\bar{g}_2 = \bar{g}_{Ca})$.

Expressions for $\varphi(V)$, C_1, C_2, C_3, $A_{11}(V), ..., B_{22}(V)$ are given in the Appendix.

From (5) it follows that the number of the bifurcation curves is determined by the number of the possible discontinuities of the first kind. The function $\varphi(V)$ has no discontinuities. The condition for the function $\psi(V)$ to have discontinuities is

$$A_{11}(V) + A_{22}(V) = 0. \tag{6}$$

For the all considered cases corresponding to the two combinations of parameters equation (6) has no solution. It means that the Hopf bifurcation curve is unique on each of the considered parameter plane and it is an argument in favor of an assumption of simple connectedness of the bursting region.

4. APPLICATIONS TO THE BURSTING REGION

Fig.1 a,b illustrates the Hopf bifurcation curves on the $(I, \bar{g}_{KCa})$ and $(I, \bar{g}_{Ca})$ planes, respectively.

The constructed curves determine the parameter values at which in the full system the bursting mode presumably exists, namely, the bursting region must be a part of the region bounded by the Hopf bifurcation curve for the slow subsystem.

Using the Runge-Kutta method with the step of 0.05 ms we have shown that the bursting region occupies almost all the region of oscillations of the slow subsystem (see Fig.1 a,b). Therefore, the present study demonstrates the good possibility to use for the model considered the conditions for oscillations in the reduced system instead of the conditions for the bursting emergence in the full system.

APPENDIX

$$
\begin{aligned}
&T(V) = \bar{g}_{Na} m_\infty^3(V)(V - V_{Na}) h_\infty(V) + \bar{g}_K n_\infty^4(V)(V - V_K) + g_L(V - V_L),\\
&T_1(V) = \bar{g}_a m_{a\infty}^3(V)(V - V_K) h_{a\infty}(V),\\
&T_2(V) = \tfrac{z_\infty(V)(V - V_{Ca})}{Ca_\infty(V)+0.5},\\
&T_3(V) = \tfrac{Ca_\infty(V)(V - V_K)}{Ca_\infty(V)+0.5}, \varphi(V) = T(V) + \bar{g}_a T_1(V) + \bar{g}_{Ca} T_2(V) + \bar{g}_{kCa} T_3(V),\\
&c = 1/23.5, \quad k = -0.5 p k_{Ca}\\
&C_1 = A_{11}A_{22} - A_{12}A_{21},\\
&C_2 = B_{11}A_{22} + A_{11}B_{22} - A_{12}B_{21} - B_{12}A_{21},\\
&C_3 = B_{11}B_{22} - B_{12}B_{21}.
\end{aligned}
\tag{7}
$$

Variant 1 (for constructing the Hopf bifurcation curve on the $(I, \bar{g}_{KCa})$ plane).

$$
\begin{aligned}
&A_{11}(V) = -cCa_\infty(V),\\
&A_{12}(V) = 0.5 c z'_\infty(V) \tfrac{V_K - V}{Ca_\infty(V)+0.5},\\
&A_{21}(V) = k(V - V_{Ca}) \tfrac{Ca_\infty(V)}{Ca_\infty(V)+0.5},\\
&A_{22}(V) = \tfrac{k z_\infty(V)[0.5(V_K - V) + Ca_\infty(V)(V_{Ca} - V)]}{(Ca_\infty(V)+0.5)^2} - pCa_\infty(V),\\
&B_{11}(V) = c[\bar{g}_{Ca}((V_{Ca} - V) z'_\infty(V) - z_\infty(V)) - (0.5 + Ca_\infty(V))(T'(V) + \bar{g}_a T'_1(V))],\\
&B_{12}(V) = c \bar{g}_{Ca} z'_\infty(V) T_2(V)\\
&B_{21}(V) = k(V - V_{Ca})(T'(V) + \bar{g}_a T'_1(V)),\\
&B_{22}(V) = \tfrac{k(V_{Ca} - V) z_\infty(V)}{Ca_\infty(V)+0.5}(T'(V) + \bar{g}_a T'_1(V)) - p[\bar{g}_{Ca} z_\infty(V) + (0.5 + Ca_\infty(V))(T'(V) + \bar{g}_a T'_1(V)
\end{aligned}
$$

Variant 2 (for constructing the Hopf bifurcation curve on the $(I, \bar{g}_{Ca})$ plane).

$$
\begin{aligned}
A_{11}(V) &= -c((V_{Ca} - V)z'_{\infty}(V) - z_{\infty}(V)), \\
A_{12}(V) &= cz'_{\infty}(V)T_2(V) \\
A_{21}(V) &= 0 \\
A_{22}(V) &= -kp \\
B_{11}(V) &= -c[(0.5 + Ca_{\infty}(V))(T'(V) + \bar{g}_a T'_1(V)) + \bar{g}_{kCa} Ca_{\infty}(V)], \\
B_{12}(V) &= 0.5c\bar{g}_{kCa} z'_{\infty}(V) \tfrac{V_K - V}{0.5 + Ca_{\infty}(V)} \\
B_{21}(V) &= \tfrac{k(V_{Ca} - V)}{c(0.5 + Ca_{\infty}(V))} B_{11}(V) \\
B_{22}(V) &= \tfrac{1}{c} B_{11}(V)(\tfrac{kT_2(V)}{0.5 + Ca_{\infty}(V)} - p) + \tfrac{0.5k\bar{g}_{kCa} z_{\infty}(V)(V_K - V)}{(0.5 + Ca_{\infty}(V))^2}
\end{aligned}
\tag{9}
$$

4.1. Line drawings

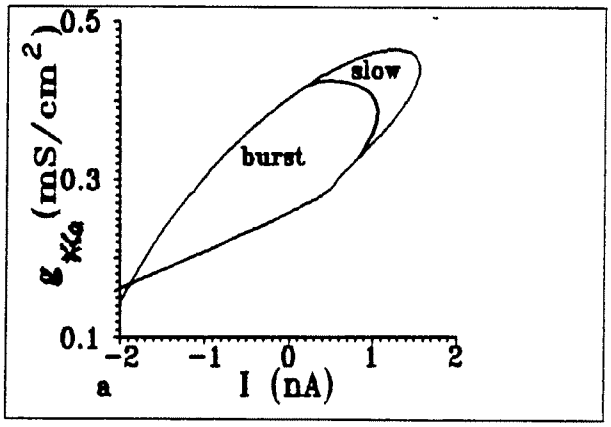


Figure 1. The Hopf bifurcation curve and bursting region on the $(I, \bar{g}_{KCa})$ plane.

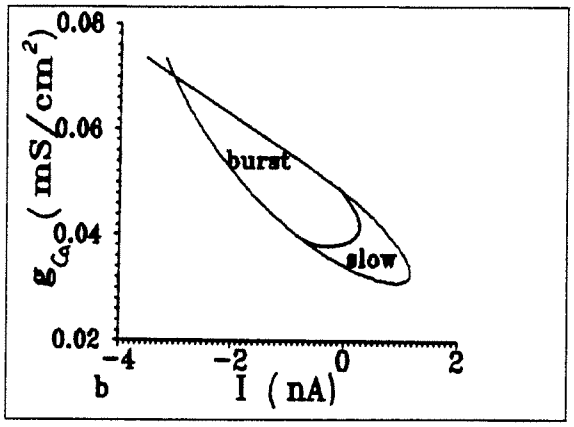


Figure 2. The Hopf bifurcation curve and bursting region on the $(I, \bar{g}_{Ca})$ plane.

REFERENCES

1. R.M. Harris-Warrick and B.R. Johnson, Brain Res. 416 (1987) 381.
2. J. Guckenheimer, S. Gueron and R.M. Harris-Warrick, Phil. Trans. R.Soc. London B 341 (1993) 345.
3. A.I. Selverston, D.F. Russell and J.P. Miller, Prog.Neurobiol. 7 (1976) 215.

NEURAL NETS: THE BRAIN AND THE COMPUTER

Frank Rattay and Helmut Pfützner

Technical University Vienna

SUMMARY

This paper discusses the relevance of modern computer techniques in the field of neural nets and their components. After a brief description of the function of a single physiological neuron (PN), it is demonstrated that:

(i) A better understanding of PN function is possible by studying electrophysiologcal models based on RC networks. Analysis and computer simulations proves to be effective to understand the dynamic functions of PN membranes including the initiation and propagation of action potentials (APs).

(ii) The PN can be used as a model for the design of artificial neurons (ANs) as the basic elements of artificial neural networks (ANNs) which at present time tend to be implemented on computers. Here we present analogies between PNs and ANs very shortly while giving a detailed discussion elsewhere (Pfützner et al. 1995).

The second part of the paper describes the architecture of physiological neural networks (PNNs) with respect to the design of computer implemented ANNs. The structures of cortex and cerebellum demonstrate that the human brain makes use of a variety of types of specialized neurons which are interconnected in a most complex way. Both cortex and cerebellum exhibit a layered structure which is also a characteristicum of ANNs. However, the analogy is restricted to basic rules of interconnections as we have discussed in detail elsewhere (Pfützner et al. 1995). Brain's architecture makes clear a high level of spatial and temporal complexity which so far has been taken over by ANN designers in a very elementary way only.

MODELING AND ANALYSIS OF NEURONS

In spite of the differences in shape, in most of the neurons we find the following functional parts: The cell body (soma) and the dendritic processes are the input region of the neuron, the axon with its branches is used for transport and distribution of neural signals (conductile region). In the input region excitatory synaptic activity causes small current impulses passing the neuronal membrane in the synaptic area. A part of these currents reaches the cell body and causes an increase of the membrane voltage. The effect is low for distant synapses because of the exponential decay of signal strength along the fiber's length coordinate. Accumulation of such graded inputs will lead to an action potential of the active membrane at the cell body when threshold is reached. Dendrites and the cell body are covered with excitatory synapses, whereas inhibitory synapses are found mostly at the cell body. Inhibitory synapses are small in amount, but they are more effective because they generate stronger postsynaptic currents and they do not lose their signal amplitudes by leakage currents.

The electrical behavior of a nerve cell may be modeled by a network of RC elements, each representing a small region of the cell membrane. The RC-circuits are connected to a network

by intracellular resistances between different regions (Perkel et al. 1987; Bressloff and Taylor 1994; Rattay 1990, 1994).

The dendrites are often assumed to have a constant membrane conductance, whereas the active membrane is more complicated to model because of the gating mechanism of the ionic channels. These are responsible for generation of action potentials and their propagation. Models for the membrane voltage are available for passive dendrites (Rall 1962) as well as for axons [(Rattay 1990, 1993; Rattay and Aberham 1993), Fig. 1]. The neuronal cell bodies show great variations in their membrane compositions: even for the standard neuron which is assumed to collect the synaptic inputs until a certain threshold is reached, the voltage across the membrane essentially depends on the number and types of ionic channels involved (Llinàs 1988). Because of the individual behavior only few models are available for the membranes of neural cell bodies (e.g. DeSchutter and Bower 1994; Belluzzi and Sacchi 1991).

The approach of modeling and analysis of real neurons has just begun. Including the time courses of the ionic currents show that the neural membrane have quite different properties at different locations which e.g. (1) enables a single neuron to work as pacemaker cell (Schild et al. 1993; McCormick and Huguenard 1992); (2) causes surprising effects compared to the predictions from the theory of Rall because of inhomogenities of cell membranes in form of hot spots in dendrites, (3) enables the neural impulse to propagate from the unmyelinated axon into the branching nonmyelinated terminal region (Fig. 1, Rattay (to appear)).

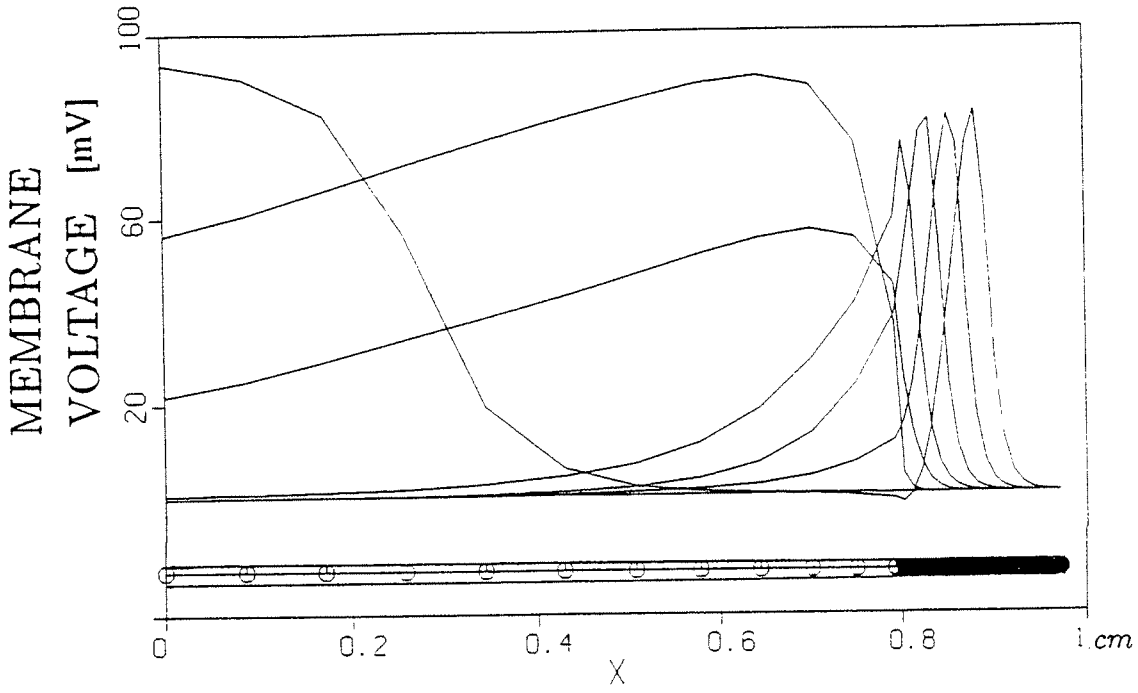


Fig. 1. Propagating action potential along a nerve fiber of 1 cm length which changes membrane composition. The curves show voltage distributions in intervals of 0.1ms. The left part of the axon is myelinated, i.e. it consists of 12 nodes of Ranvier (marked by circles in the lower part) with active membrane and the nodes are separated by insulating segments which allow high propagation velocity at low energy consumption. The distal part of the fiber is unmyelinated and electrical behavior changes drastically. Computer simulation shows that signal transport from myelinated into the unmyelinated part is possible, if the last internodes become gradually shorter in a way as it is seen in microscope.

Analysis of the dynamics of compartmental models shows that neural reactions essentially depend on temporal parameters. The law of „all or nothing" for neural signals is not valid in all situations: Changes in geometry cause differences in spike shape and refractory behavior (Fig. 1), and thereby from a high frequency chain of spikes several impulses may be lost in the branching part. It is not always predictable, which branch will lose a special neural signal. The duration of an action potential is of the order of 1ms, but accumulation of signals in the input region is still lower. In contrast to this temporal behavior it is fascinating that the brain is able to discern firing patterns containing a time difference of 10 μs as it is known, e.g., from directional hearing experiments.

THE ARCHITECTURE OF NEURAL NETS

ANNs have been introduced as a computational concept in order to attain human-like performance in speech and image recognition. The basic elements of neural net models are the net topology, the node characteristics and the learning rules. It is believed that some of the most prominent features of the biological brain also have been used in ANNs. However, it is not possible to obtain the performance of the biological brain.

For specific applications different ANNs are used. They are well defined in their net topology, node characteristis and learning rules (Lippmann 1987). The following features of PNNs are, however, not found in ANNs commonly used: (1) The temporal component in signal propagation, which may be of importance for durations down to 1% of a neural impulse, (2) chemical influences which change the firing behavior, (3) cooperation of different specialized neurons within the neural net. For details see Pfützner et al. 1995.

We receive up to 10^9 bits/second in the form of neural impulses (spikes) with our sensory system, however, less than 100 bits/s become conscious verbal information. As an example, within three layers of neurons in the retina the optical system processes the information and reduces it to 10^7-10^8 bits/s, which arrive via 10^6 fibers of the optical nerve at the lateral geniculate nucleus and finally at the optical cortex. The human neocortex consists of about 10^{10} interneurons, each of them makes about 10^4 synaptic contacts with other neurons and a great number of those 10^{14} electric switches operate several hundred times within the few seconds which are necessary to recognize a situation and to say, "Good morning, Mr. Miller, I have seen your girl friend last night."

THE CORTEX: SPECIALIZED NEURONS MAKE CONTACTS IN ALL DIRECTIONS OF THE CLOSE NEIGHBORHOOD, PYRAMIDAL CELLS MAKE THE LONG-DISTANCE CALLS

The human cerebral cortex (gray matter) is about 2-3mm thick and neuronal cell bodies are concentrated in 6 layers (I-VI; Fig. 2). 70-75% of these neurons are pyramidal cells (Braitenberg 1977, Hendry et al 1987). Pyramidal cells of layer V have about 15000 spines at their dendritic input region, and each of them receives information in form of a synaptic signal on the head of the spine. The spines may change their chemical properties (calcium or other second messenger concentrations) in response of neural activities for information storage in the brain (Koch et al. 1992).

Fig. 2 shows schematically different types of cortical neurons arranged in a cylinder with a diameter of 300μ which is directly influenced from the branching region of a single nerve fiber (axon) carrying the information from another cortical region in the form of neural impulses which are called action potentials (AP). The non-pyramidal cells are local circuit neurons and have no axons which descend into the subcortical white matter. Since the non-spiny or

inhibitory neurons are in this group, it follows that all inhibitory activity in the cortex is generated locally (Creutzfeldt 1977, Fairén et al. 1984).

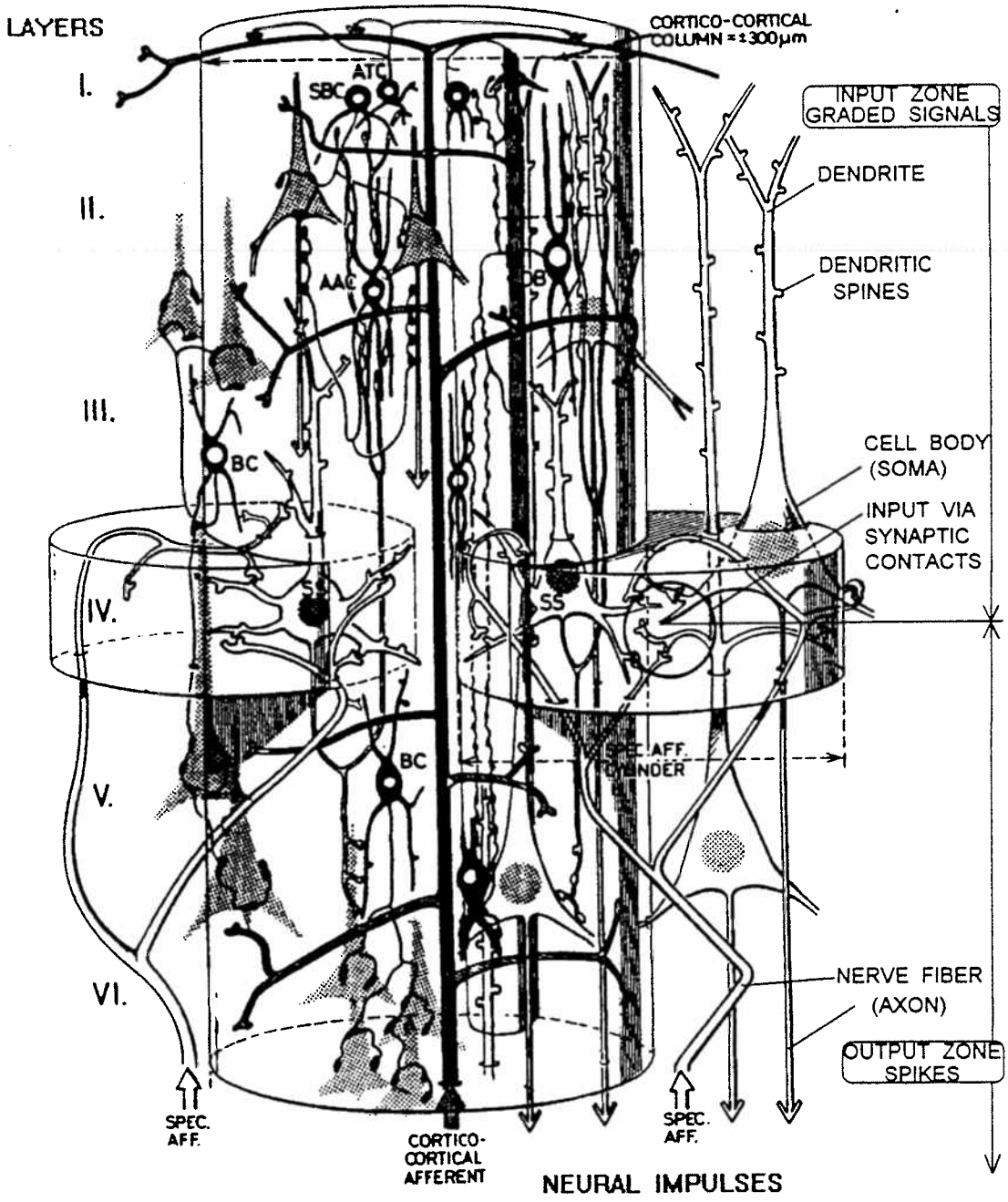


Fig. 2. Neural network of the human cortex. The terminal section of an afferent cortico-cortical axon supplies pyramidal cells and non-pyramidal cells of the six cortical layers (ATC axonal tuft cell, SBC small bascet cell, AAC axoaxonic chandeller cell, BC bascet cell, CDB double bouquet cell, SS spiny stellate cell). Cells in black are assumed to be inhibitory.
On the right edge the functional elements of a neuron are marked. Within the circle a specific afferent axon makes synaptic contact with a spine of the dendrite of a pyramidal cell.
Modified after Szentágothai 1983

THE CEREBELLUM: A SIMPLE ARCHITECTURE CONTAINING A DATA BUS SYSTEM A WITH TEMPORAL COMPONENT

The cerebellar cortex has three layers (Fig. 3). The inner layer III is packed with 10^{10}-10^{11} granule cells. They obtain their inputs via mossy fibers from deep cerebellar nuclei and send axons into layer I to form a system of parallel fibers, each several mm long. Layer II is occupied by Purkinje cells. Their dendrites which span layer I and II are more or less within a plane, and each of these neurons receives inputs from about 200000 parallel fibers. The axons of Purkinje cells form the sole output from the cerebellar cortex. As they are inhibitory, they can only modulate ongoing activity. The significance of such an arrangement is that a group of Purkinje cells, which are supplied with the same information by the parallel fiber system, spans several joints in a somatotopic region, e.g. shoulder, elbow and wrist joints of the arm and thereby providing a possible mechanism for coordinating multi-joint movement (Nichols et al. 1992). It is believed that the delay caused by the slow propagation of neural impulses along the axons of the granulle cells (parallel fibers) is of importance for the timing of the motor system (Braitenberg 1967).

By comparison of Fig. 2 and Fig. 3 it is seen that - in contrast to the neocortex - the cortex of the cerebellum has an orthogonal structure and a data bus with a temporal component.

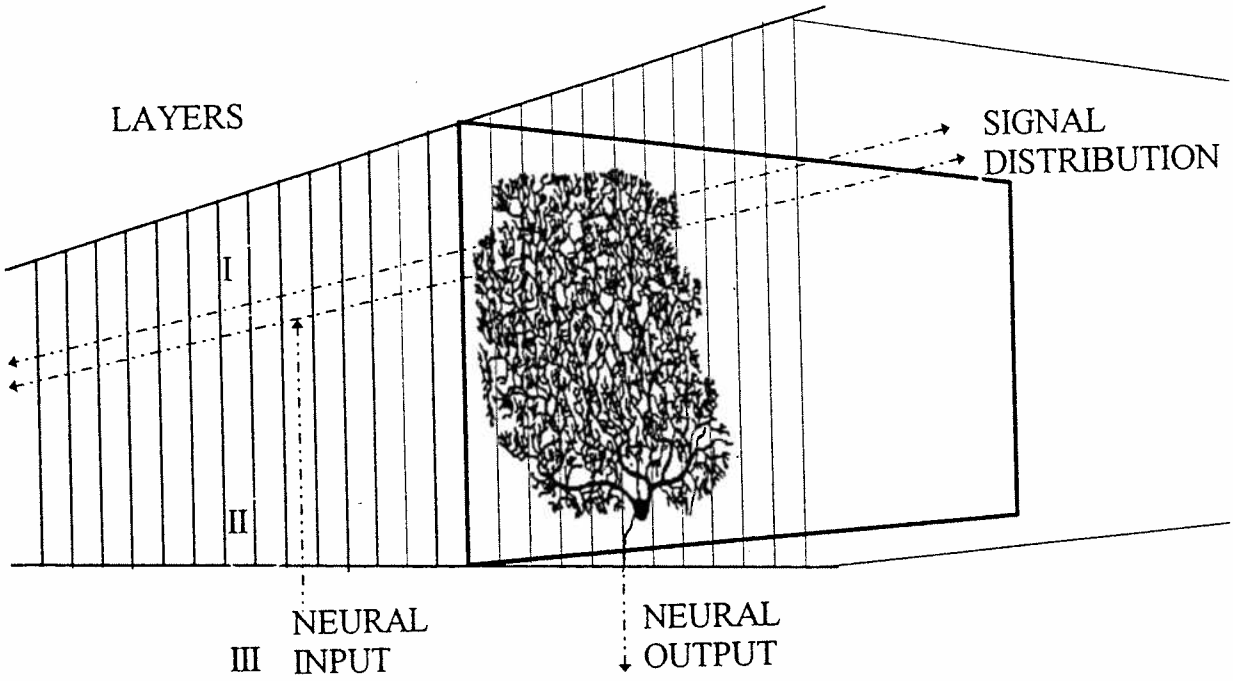


Fig. 3 Organization of the cerebellar cortex. A system of parallel axons from granulle cells crosses at right angles the planes containing the dendritic branches of two-dimensional Purkinje cells. With some delay this data bus system supplies a series of Purkinje cells with the same information.

CONCLUSION

Well organized specialized neural elements manage the high performance of the biological brain. Further loans from the PNN features will be a good challenge to improve the efficienty of ANNs.

REFERENCES

Belluzzi O. and Sacchi O. 1991. A five-conductance model of the action potential in the rat sympathetic neurone. Prog. Biophys. molec. Biol. 55, 1-30

Braitenberg V. 1967. Is the cerebellar cortex a biological clock in the millisecond range ? In: Progress in brain research. The Cerebellum. Elsevier Publ. Amsterdam, 25, 334-346

Braitenberg V. 1977. *On the texture of brain.* Springer

Bressloff P.C. and Taylor J.G. 1994. Dynamics of compartmental model neurons. Neural Networks 7, 1153-1165

Creutzfeldt O.D. 1977. Generality of the functional structure of the neocortex. Naturwissenschaften 64, 507-517

De Schutter E. and Bower J.M. 1994. An active model of the cerebellar Purkinje cell: I. simulation of current clamps in slice. J neurophysiol. 71, 375-400

Fairén A., DeFelipe J. and Regidor J. 1984. Nonpyramidal neurons. In: A. Peters and E.G. Jones, eds. Cerebral Cortex, vol 1, Plenum

Hendry S.H.C., Schwark H.D., Jones E.G. and Yan J. 1987. Numbers and proportions of GABA-immunoreactive neurons in different areas of monkey cerebral cortex. J. Neurosci. 7, 1503-1519

Koch C. Zador A and Brown T.H. 1992. Dendritic spines: Convergence of theory and experiment. Science 256, 973-974

Lippmann R.P. An Introduction to computing with neural nets. IEEE ASSP Magazine April 1987, 4-22

Llinàs R. 1988. The intrinsic electrophysiological properties of mammalian neurons: insights into central nervous system functions. Science 242, 1654-1664

Nichols J.G., Martin A.R. and Wallace B.G. 1992. *From neuron to brain: A cellular and molecular approach to the function of the nervous system.* Sinnauer Associates, Sunderland MA, 3rd. edition, p 540

McCormick D.A. and Huguenard J.R. 1992. A model of the electrophysiological properties of Thalamocorical relay neurons. J. neurophysiol. 68, 1384-1400

Perkel D.H., Mulloney B. and Budelli R.W. 1981. Quantitative methods for predicting neuronal behavior. Neuroscience 6, 823-837

Pfützner H., Nussbaum C., Booth T. and Rattay F. 1995. Physiological analoga of artificial networks. Studies in Applied Electrodynamics ISEM, Cardiff, GB

Rall W. 1962. Theory of physiological properties of dendrites. Ann. N.Y. Acad. Sci. 96, 1071-1092

Rattay F. 1990. *Electrical nerve stimulation.* Springer Wien.

Rattay F. 1993. Simulation of artificial neural reactions produced with electric fields. Simulation Practice & Theory. 1, 137-152

Rattay F. 1994. Modelling the excitation and the propagation of nerve impulses by natural and artificial stimulations. Proc. IMACS Symp. on mathematical modelling, Vienna, Vol 2, 323-327

Rattay F. (in press). Propagation and distribution of neural signals: a modeling study of axonal transport. Physics of the Alive

Rattay F. and Aberham M. 1993. Modeling axon membranes for functional electrical stimulation. IEEE-Trans. Biomed. Eng. BME 40, 1201-1209

Schild J.H., Khushalani S., Clark J.W., Andersen M.C., Kunze D.I. and Yang M. 1993. An ionic model for neurons in the rat medical nucleus tractus solitarii receeiving sensory afferent input. J Physiol. 469, 341-363

Szentágothai J. 1983. The modular architectonic principle of neural centers. Rev. Physiol. Biochem. Pharmacol. 98, 35-61

The knowledge modelling for human skeleton animation : towards a simulation prototype

H . Fourar*, A . Koukam+ , P. Deschizeaux*

*Université de Haute Alsace, 2, rue des frères lumière, 68093 Mulhouse. France

+Institut Polytechnique de Sévenans, 90010 Belfort-Cédex. France

1. INTRODUCTION

Computer-aided animation systems are widely used for studying many human aspects such as movement, performance and human factors. Through the progress made to date in computer animation, the need for the systems to provide assistance and more help to the animator has been well recognized [6], [2]. The assistance can range from control to guidance, including automatic animation. The main problem in using these systems is the degree of assistance that they provide to the user. From the user view point, we can distinguish two main approaches. In the first approach, the user must describe explicitly each human movement and the trajectory of each object. Thus the user is overly involved in the strategies and tools of producing animation. In the second approach, the user describes the animation task in abstract terms and the system undertakes to generate the sequence of required movements and actions. Achieving this assistance requires a system that embeds several kinds of knowledge about the entities to be animated [9]. The main subject of this paper is attacking the problems posed by the design of systems based on the second approach. To this end, we propose a modelling approach drawing from research into object oriented modelling and artificial intelligence. From object oriented modelling [10], we discovered suitable concepts such as class, object and relationship to elaborate the real world and operational models. The real world model consists on the human skeleton object and the entities composing it's environment, while the operational one focus on the tasks which can be executed by the human skeleton. From the artificial intelligence research, we use the planning paradigm to define the animation process. Such paradigm constitutes a basis for the mapping from a given task to a sequence of atomic actions. The object oriented modelling of the animation universe is presented in section 2. Each class identified in the animation universe is described by using the entity-relationship model as a graphical representation [3]. The animation process and an example are enlarged upon in section 3.

2. KNOWLEDGE MODELLING OF THE ANIMATION UNIVERSE

The animation universe is composed by the human skeleton, its environment and the operational model. For each of these components, we give an object oriented model that we represent graphically.

2.1. The human skeleton model

We have used the wire frame model for geometrical representation of the human skeleton. The skeleton (mannequin) is constituted of segment groups linked by the articulations. A segment represents the frame of the member. The articulation defines the associated points around which the member turns. The modelling of the skeleton is based on the detail concept. A detail is a segment group connected by articulations. Thus a detail can be included in other details to authorize the transformation of a segment group, as shown in figure 1.

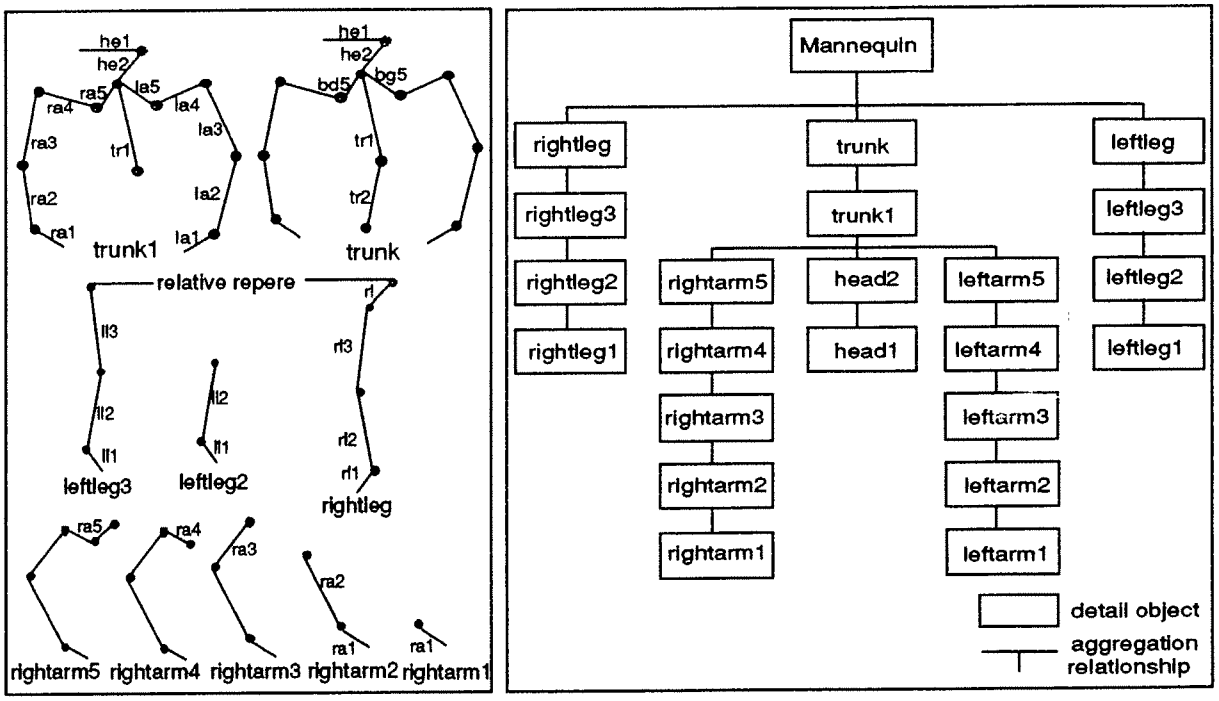


Figure 1. Detail based description of human skeleton.

The human skeleton is described by a class (Mannequin) that possesses a static view defining the properties of the skeleton, the relationships that connect it to other classes and a dynamic view that describes its behavior. The figure 2 presents the object model of human skeleton. Relationships are shown by circles and lines drawn between entities. In this model, each relationship is characterized by the cardinalities which define the minimal and maximal participation number of a class instance involved in this relationship. Class entities may have attributes, which are the data values held by the objects.

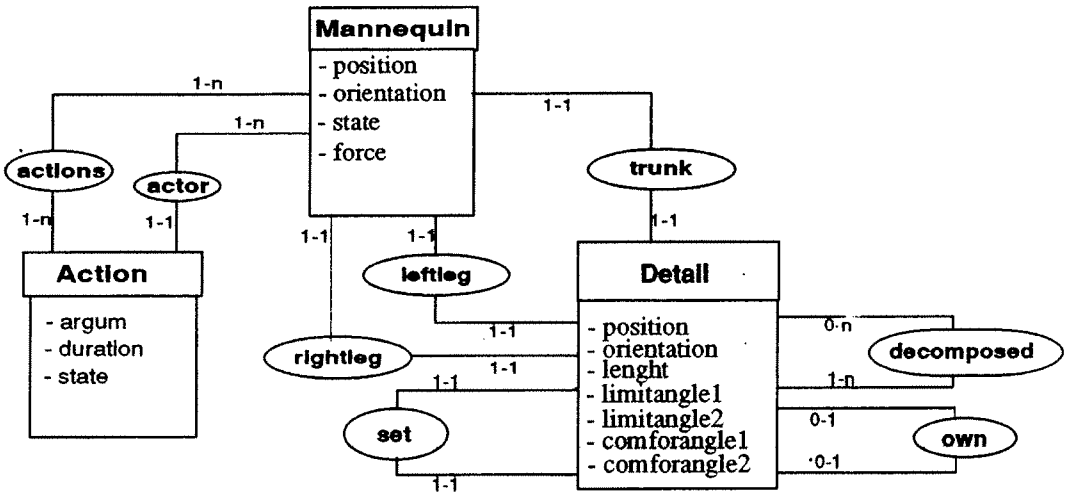


Figure 2. Human skeleton model.

Some attribute values of the object details are constrained by the anatomical limits of the human skeleton. Thus, each movement of the human skeleton must be performed according to these constraints.

2.2. The environment model

We have used the constructive solid geometry (CSG) model for the objects of the environment. The modelling by the CSG tree provides a hierarchical and description of objects that compose the environment as shown in figure 3. Thus, an object can be defined in terms of the primitive objects (cube, sphere, cylinder,...) by using the union and the difference relationships [7]. The figure 4 gives an example of the CSG representation.

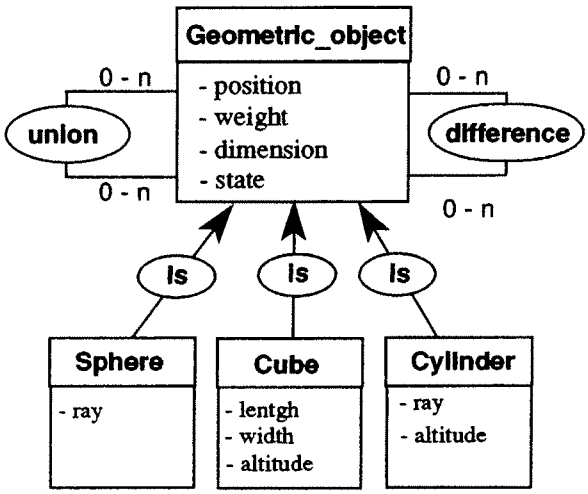


Figure 3. Environment model.

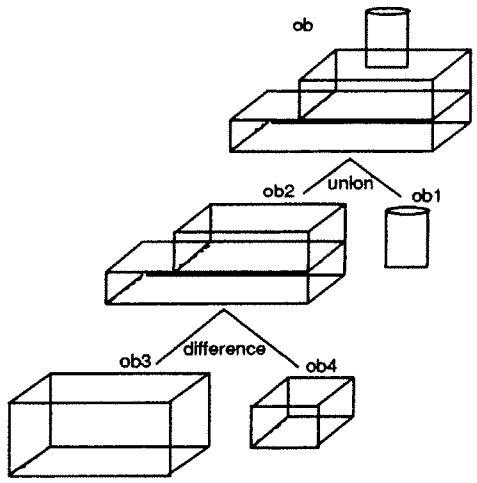


Figure 4. Example of a CSG representation.

The union relationship expresses the aggregation, while the difference relationship, allows to extract an object of another. The **is** relationship expresses that the classes cube, sphere and cylinder inherit the properties and the methods of the class Geometric_object.

2.3. The operational model

The operational model describes the animation tasks. It is organized hierarchically around four classes : task, action, movement and primitive (figure 5). A task object represents a goal to achieve in the animation process and is defined as a sequence of action objects. The low level class of the operational model is the primitive class which describes the rotation and translation of the human skeleton members. Each action splits hierarchically in movements then in primitives forming respectively the movement and the primitive plans. We have introduced the time in the representation of plan to express the temporal constraints such as simultaneity and precedence.

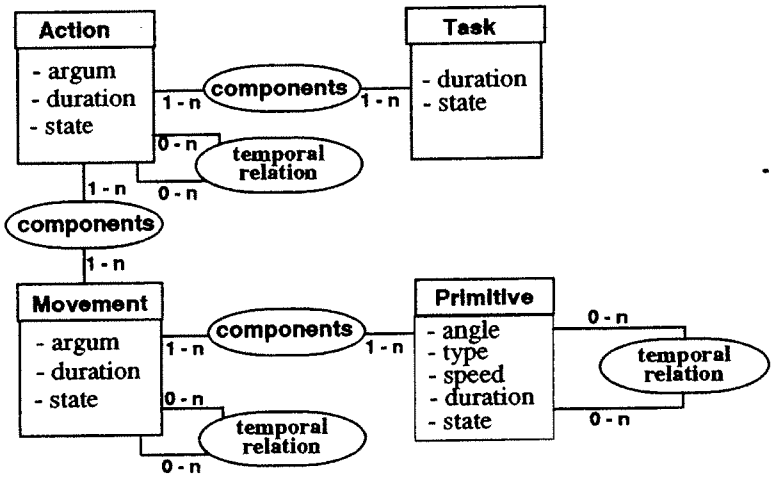


Figure 5. Operational model.

The temporal representation of the plan is based on the formalism of Allen that we found to be better qualified to express the time in planning [1]. This formalism allows us to express the causality relationships between the actions by using six temporal relationships: equals, starts, during, finishes, before, meets and overlaps.

3. ANIMATION PROCESS

3.1. The animation planning process

The major difficulty in performing a high level animation task lies in the conversion of abstract descriptions into low level primitive actions [4]. The animation process is based on the planning paradigm, an artificial intelligence approach to a theory of actions. However, using this paradigm presupposes the knowledge modelling of a domain which is generally expressed in operators (parameterized templates defining the possible actions of the domain) together with a state schema (a set of predicates that describe the state of the word for that domain). In our study, the operators and the state schema are defined by the operational and the real world models respectively.

Given a task, the plan generation is invoked to produce a sequence of geometrical primitives which can be executed by a CAD system. This sequence is

constructed hierarchically according to the abstraction levels identified in the operational model. The planning process generates first an abstract plan constituted of instances of the action class (action level). This plan is then affined, leading to a sequence of movements (movement level) which is finally transformed into a concrete plan: a sequence of geometrical primitives [8]. The following section presents an example to illustrate the planning process.

3.2. Example

We give a partial example of a hierarchical plan that can be generated to achieve the take task. The arrows between levels connect a high level operator with those instantiated to perform it. For example, an instantiation of the lift action involves the instantiation of four movements : incline_bust$_1$, lower_leg, lift_bust and incline_bust$_2$. Arrows within levels show how the performing of certain operators is partially ordered with respect to causality relationships. For example, the movement Incline_bust$_1$ must be performed before lower_leg and incline_bust$_2$. However these later can be carried out concurrently.

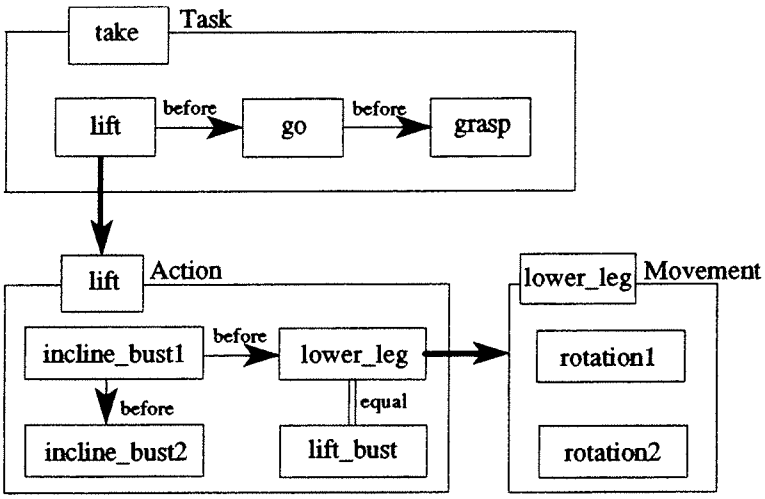


Figure 6. Hierarchical plan of take task.

To illustrate the instances generated at each level, we give below the example of the movement lower_leg.

(Movement : lower_leg
-- argum : leftleg3
-- components : [rotation$_1$, rotation$_2$]
-- before : [incline_bust$_1$]
-- equals : [lift_bust])

The plan generated to achieve this movement is constituted by the geometrical primitives rotation$_1$ and rotation$_2$, and the sequential execution of this plan by a CAD system is illustrated by the figure 7.a. As there is no temporal relationship between the two rotation primitives, they can be performed concurrently as shown in figure 7.b.

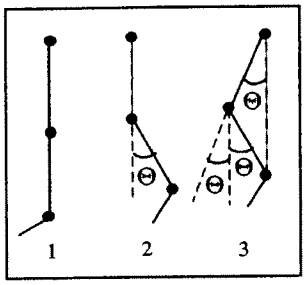


Figure 7.a. Sequential execution of plan.

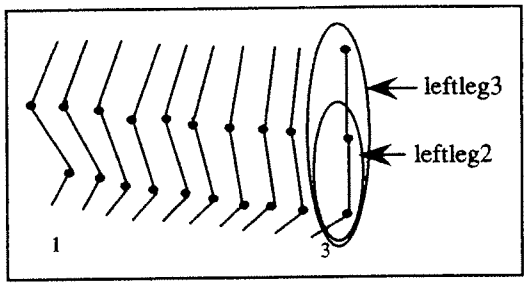


Figure 7.b. Parallel execution of plan.

4. CONCLUSION

This paper has described an approach for human skeleton animation. The approach involves the use of the object oriented paradigm for modelling the animation universe and the planning paradigm for modelling the animation process. Based on this approach, we have built a prototype for experimentation with the KOOL (Knowledge Object Oriented Language) expert system generator [5]. Given a task, the prototype generates a sequence of low level geometrical primitives which can be directly executed by the CAD software CATIA.

REFERENCES

1. J.F Allen, Towards a General Theory of Action and Time. Artificial Intelligence, 23:123-154, 1984.
2. N.I Badler, B.A Barsky and D. Zeltzer, Making Them Move : Mechanics, Control and Animation of Articulated Figures. Morgan Kaufmann Publishers, INC California, 1992.
3. P. Chen, The Entity-Relationship Model Toward a Unified View of Data, ACM TODS 1(1), 1976.
4. N. Cherif, GEMSA: Computer-Aided Movement Generation for Scene Animation. Computer Animation'90, Ed Magnenat-Thalmann & D. Thalmann, 119-128, 1990.
5. KOOL 4x4 Manuel de référence. Bull S.A, Cedoc-Dilog, décembre 1990.
6. N. Magnenat-Thalmann and D. Thalmann, Computer Graphics and Animation. Elsevier Science Publishers B.V, Computer Physics Reports 11, Amsterdam 1989.
7. M. Mahieddine, Modélisation, visualisation et animation d'objets graphiques 3D: approche orientée objets. Thèse de docteur en informatique, université de Nice sophia-antipolis, octobre 1991.
8. H. Fourar, Un système à base de connaissances pour la simulation et l'animation d'un mannequin 3D. Rapport interne. Université de Haute Alsace, Mulhouse, France, Juin 1993.
9. C.W Reynolds, Flocks, herds, and schools : A distributed behavioral model, Proc SIGGRAPH '87, Computer Graphics 21(4):25-34, 1987.
10. S. Shlaer and S.J Mellor, Object-Oriented Systems Analysis: Modeling the World in Data. Prentice Hall, Englewood Cliffs, 1988.

Author Index

Proceedings

Session "Software Tools and Products"

F. Breitenecker
I. Husinsky
editors

ARGESIM Report No. 2

ISBN 3-901608-01-X

ISBN ebook 978-3-901608-01-8

DOI 10.11128/arep.1-2-3.ar2

ISBN 3-901608-01-X

ISBN ebook 978-3-901608-01-8

DOI 10.11128/arep.1-2-3.ar2

ARGESIM Report No. 2

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

Printed by *CA Druckerei*, Vienna, Austria

Foreword

Developments over the last years show that beside the classical tools theory and experiment simulation becomes more and more the third major tool for problem solving in application and research. Nowadays simulation is found in nearly every application area, research activities result in new methodologies and tools for simulation, and more and more simulation software, simulators, and simulation systems are offered on the market.
The *EUROSIM Congress*, the European Simulation Congress, an international event normally held every three years, aims to be a common forum for presenting European and international recent results and applications in simulation, and to stimulate the exchange of ideas and experiences among scientists and engineers active in simulation.
EUROSIM is the Federation of the European Simulation Societies, acting as a European forum for Simulation Societies and promoting the advancement of system simulation in industry, research, and education.

All these intentions are reflected in the 5th European Simulation Congress *EUROSIM 95*, the 2nd Congress after the formal foundation of EUROSIM. The scientific programme consists of invited and contributed papers to regular sessions and to "Special Interest Sessions", of contributions to the session "Software Tools and Products", and of posters.

The invited and contributed papers to regular sessions and to "Special Interest Sessions" are published in the Congress Proceedings printed by Elsevier Science B.V. and in a Late Paper Volume (ARGESIM Report no. 1, ISBN 3-901608-01-X). The Proceedings contain eight invited papers and 212 contributed papers, the Late Paper Volume contains 20 contributed papers. The papers were selected by the International Programme Committee from 459 abstracts received.

The session "Software Tools and Products" presents papers dealing with State-of-the-Art and new features of simulation languages, simulators, and simulation environments. These contributions also passed the review process and are published as ARGESIM Report no. 2 (Proceedings EUROSIM'95 - Session "Software Tools and Products", ISBN 3-901608-01-X).

The reviewed Poster Session completes the scientific programme. The abstracts of the 111 posters are published as ARGESIM Report no. 3 (EUROSIM'95 - Poster Book, ISBN 3-901608-01-X).

It is interesting to compare the titles of papers and posters presented at previous European Simulation Congresses with those at the present congress. Even a brief glance through the four volumes of Proceedings and Late Paper Volumes shows that in this twelve year period considerable, remarkable, and sometimes astonishing advances have been made in a number of different areas. For example, developments in parallelism and distributed processing are now not only being seen in simulation applications but are also frequently used.
Object-oriented methods are being implemented now, and artificial intelligence and knowledge-based tools appear to be an established part of system modelling and simulation methodology. The availability of improved graphic algorithms and tools is also leading to some very interesting and innovative research and application in terms of man-machine interface and of animation and visualisation, both for discrete-event and continuous-system simulation.

New developments in terms of mathematical modelling and simulation techniques as well as in terms of general methodology are of little significance unless they are stimulated by the requirements of the real world in terms of industry, business, agriculture and the sciences. We are very pleased, therefore, that application papers are so well represented. This also applies to papers on parallel and distributed simulation, where beside graphics the fastest development can be observed.

We are also pleased that the idea of "Special Interest Sessions" could be realized. These sessions deal with recent developments in areas where methodology and application are considered together. The results of the closing discussion at the end of these sessions are summarized in manuscripts which will be edited and published in abbreviated form in *EUROSIM - Simulation News Europe* (SNE), the newsletter of the EUROSIM member societies. Some of these papers will be prepared for publication in EUROSIM's scientific journal *SIMULATION PRACTICE AND THEORY*. A separate role is played by the Industry Session on "Model Exchange and Software Independent Modeling" where people mainly from industry report on this topic without necessarily having to publish a paper in the Proceedings. Furthermore, we are pleased, that the contributions to the session "Software Tools and Products" show a very broad spectrum of simulation software, and that the Poster Session presents new ideas under development.

The European Simulation Congress *EUROSIM 95*, held in Vienna (Austria) at the Technical University of Vienna from September 11 through September 15, 1995, is organized by *ASIM* (Arbeitsgemeinschaft Simulation), the German speaking Simulation Society, in co-operation with the other member societies of EUROSIM: *AES* (Asociación Española de Simulación), *CSSS* (Czech & Slovak Simulation Society), *DBSS* (Dutch Benelux Simulation Society), *FRANCOSIM* (Société Francophone de Simulation), *HSTAG* (Hungarian Simulation Tools and Application Group), *ISCS* (Italian Society for Computer Simulation), *SIMS* (Simulation Society of Scandinavia), *SLOSIM* (Slovenian Society for Simulation and Modelling), *UKSS* (United Kingdom Simulation Society).

The moral co-sponsorship of *CASS* (Chinese Association for System Simulation), CROSSIM (Croatian Society for Simulation Modelling), *IFAC* Advisory Board Austria, *IMACS* (International Association for Mathematics and Computers in Simulation), *JSST* (Japanese Society for Simulation Technology), *LSS* (Latvian Simulation Society), *OCG* (Austrian Computer Society), *PSCS* (Polish Society for Computer Simulation), *ROMSIM* (Romanian Society for Modelling and Simulation), *SCS* (Society for Computer Simulation), *SiE* Esprit Working Group "Simulation in Europe" supports this congress.

A successful conference is always due to the efforts of the many people involved. To this purpose, particular acknowledgement goes to the members of the Scientific Committee for their contributions in the paper selection process, to the members of the Local Organizing Committee, and more especially to the head of this committee, to Mr. Manfred Salzmann. We would like to thank *Unseld + Partner* and *CA (Creditanstalt)* for sponsoring the printing of this report.

Felix Breitenecker Irmgard Husinsky

Technical University of Vienna

Scientific Committee

F. Breitenecker (Austria), Chairman

H. Adelsberger (D)
M. Alexik (SQ)
W. Ameling (D)
S. Balsamo (I)
I. Bausch-Gall (D)
S.W. Brok (NL)
F.E. Cellier (USA)
V. Ceric (HR)
L. Dekker (NL)
J. Dongarra (USA)
V. De Nitto (I)
H. Ecker (A)
G. Feichtinger (A)
P. Fishwick (USA)
J.M. Giron-Sierra (E)
H.J. Halin (CH)
N. Houbak (DK)
R. Huntsinger (USA)
I. Husinsky (A)
T. Iversen (N)
A. Jávor (H)
K. Juslin (FIN)
G. Kampe (D)
E. Kerckhoffs (NL)
W. Kleinert (A)
P. Kopacek (A)
W. Kreutzer (NZ)
M. Lebrun (F)
F. Lorenz (B)
F. Maceri (I)
D. Maclay (GB)
H. Mang (A)
Y. Merkuryev (LV)
Z. Minglian (VRC)
I. Molnar (H)
D.P.F. Möller (D)
D. Murray-Smith (GB)
F.J. Pasveer (NL)
R. Pooley (U.K.)
W. Purgathofer (A)
P. Schäfer (D)
T. Schriber (USA)
A. Seila (USA)
W. Smit (NL)
F. Stanciulescu (RO)
A. Sydow (D)
H. Szczerbicka (D)
S. Takaba (J)
I. Troch (A)
G.C. Vansteenkiste (B)
W. Weisz (A)
J. Wilkinson (GB)
R. Zobel (GB)
B. Zupancic (SLO)

Organization Committee

Felix Breitenecker, Dept. Simulation Techniques,
Institute for Technical Mathematics, Technical University of Vienna

Irmgard Husinsky, Dept. for High Performance Computing, Computing Services,
Technical University of Vienna

Local Organizers are the Dept. Simulation Techniques of the Technical University of Vienna and the Dept. for High Performance Computing of the Computing Services of the Technical University of Vienna, and the "ARGE Simulation News".

Local Organizing Committee

M. Salzmann;

K. Breitenecker, B. Gabler, M. Holzinger, C. Kiss, M. Klug, N. Kraus, M. Lingl,
I. Mannsberger, J. Schuch, G. Schuster, H. Strauß, S. Wassertheurer, W. Zeller

About ARGESIM

ARGE Simulation News (ARGESIM) is a non-profit working group providing the infra structure for the administration of **EUROSIM** activities and other activities in the area of modelling and simulation.

ARGESIM organizes and provides the infra structure for

- the production of the journal EUROSIM Simulation News Europe
- the comparison of simulation software (EUROSIM Comparisons)
- the organisation of seminars and courses on modelling and simulation
- COMETT Courses on Simulation
- "Seminare über Modellbildung und Simulation"
- development of simulation software, for instance: mosis - continuous parallel simulation, D_SIM - discrete simulation with Petri Nets, GOMA - optimization in ACSL
- running a WWW - server on EUROSIM activities and on activities of member societies of EUROSIM
- running a FTP-Server with software demos, for instance
  - * demos of continuous simulation software
  - * demos of discrete simulation software
  - * demos of engineering software tools
  - * full versions of tools developed within ARGESIM

At present ARGESIM consists mainly of staff members of the Dept. Simulation Technique and of the Computing Services of the Technical University Vienna.

In 1995 ARGESIM became also a publisher and started the series **ARGESIM Reports**. These reports will publish short monographs on new developments in modelling and simulation, course material for COMETT courses and other simulation courses, Proceedings for simulation conferences, summaries of the EUROSIM comparisons, etc.

Up to now the following reports have been published:

No.	Title	Authors / Editors	ISBN
# 1	Congress EUROSIM'95 - Late Paper Volume	F. Breitenecker, I. Husinsky	3-901608-01-X
# 2	Congress EUROSIM'95 - Session Software Products and Tools	F. Breitenecker, I. Husinsky	3-901608-01-X
# 3	EUROSIM'95 - Poster Book	F. Breitenecker, I. Husinsky	3-901608-01-X
# 4	Seminar Modellbildung und Simulation - Simulation in der Didaktik	F. Breitenecker, I. Husinsky, M. Salzmann	3-901608-04-4
# 5	Seminar Modellbildung und Simulation - COMETT - Course "Fuzzy Logic"	D. Murray-Smith, D.P.F. Möller, F. Breitenecker	3-901608-04-4
# 6	Seminar Modellbildung und Simulation -COMETT - Course "Object-Oriented Discrete Simulation"	N. Kraus, F. Breitenecker	3-901608-04-4
# 7	EUROSIM Comparison 1 - Solutions and Results	F. Breitenecker, I. Husinsky	3-901608-07-9
# 8	EUROSIM Comparison 2 - Solutions and Results	F. Breitenecker, I. Husinsky	3-901608-07-9

For information contact: ARGESIM, c/o Dept. Simulation Techniques,
attn. F. Breitenecker, Technical University Vienna
Wiedner Hauptstraße 8-10, A - 1040 Vienna
Tel. +43-1-58801-5374, -5386, -5484, Fax: +43-1-5874211
Email: argesim@simserv.tuwien.ac.at

Table of Contents

"HYPAS" - Software Simulation Package for Electro-Hydraulic Drive Installations.

Authors:
Fl. Ionescu, Professor at Fachhochschule Konstanz, Department of Mechanical Engineering, 55 Braunegger Str., 78462 Konstanz, Germany

Fl. Hasler, Student at Fachhochschule Konstanz, Department of Mechanical Engineering, 55 Braunegger Str., 78462 Konstanz, Germany

A. Ciomaga, Student at Fachhochschule Konstanz, Department of Mechanical Engineering, 55 Braunegger Str., 78462 Konstanz, Germany

1. Introduction

The program package is based on an original method of structured methodology and on a high experience in the field of hydro-pneumatical design and mathematical modelling. HYPAS frees the user from the difficult task of describing the system analytical behaviour and offers the possibility to design and to simulate the installation's behaviour using its own constructive elements. The scheme's set-up is made through a standardised representation. The basics are the non-linear and linearised mathematical models, which have been developed and implemented taking into consideration the specific physical behaviour of the studied components.

2. How to construct a simulation scheme

The construction of the simulation schemes is based on the association of some graphical symbols to the compound elements of the electro-hydraulic drive installations. The de-facto design of these symbols and the set-up for the connection points with other constructive elements is done with the help of a graphical symbols' editor. These symbols hide in their background mathematical models which are used in the simulation stage as a descriptor for the operation mode. Both the graphical symbol and the associated mathematical model can be adjusted during the program running.

The fundamental objects are classified from the functional point of view through object-folder definition, which group the elements with the same functionality but are different from the constructive point of view. The compound elements of these object-folders are the background used to construct the electro-hydraulic drive installations which are to be simulated.

The connections among the different elements of the simulation scheme are made by means of the connection lines. These connection lines include a set of segments which link different objects and which intersect each other.

The macro-definitions are the general form for a composite element representation. These represent a collection of elementary objects or of other macro-definitions connected in a complex way. The definition for these macros was introduced to ease the design and the comprehension of the scheme, by dividing it in different aggregates from the functional and/or constructive point of view, as well as to separate the constructive elements of the electro-hydraulic drive installations. Obviously, the macro's associated mathematical model will be made by the reunion of all associated models of the included elements. The internal structure of a macro-definition can be observed by means of a window which allows both the view and the change of its internal structure. With the help of this window as well as of the options offered by the program one can modify the macro-definition's structure and the structure for the included macros, at any level, deleting or adding objects or modifying the connections among them. These changes will modify the mathematical model for the selected macro-definition.

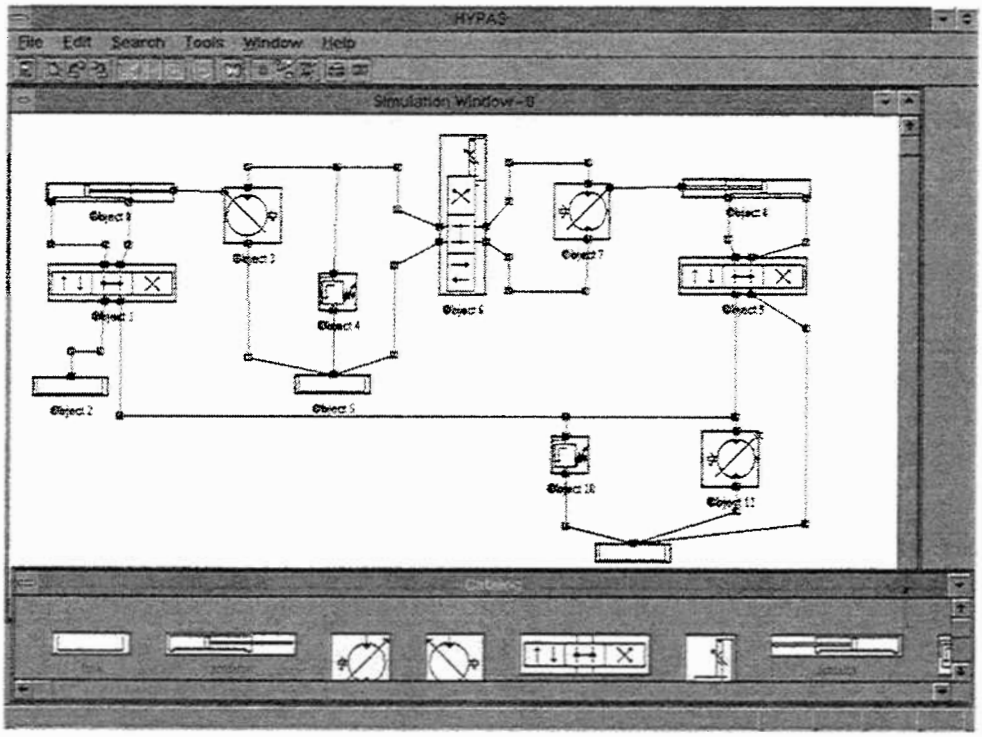
To start the design of a particular scheme one must open a simulation window, which will keep the scheme, and the desired catalogues which contain the objects included in the electro-hydraulic drive installation. To construct a scheme one must select the needed symbols from the catalogues and add them to the simulation window through a drag-and-drop like technique and then must connect them

according to the construction mode of the installation. The definition for the macros is made both through the selection of a group of objects and through the explicit construction.
The possible changes in the structure of an already designed scheme include:

- objects rearranging,
- object resizing,
- rearranging of the connection points within the associated graphical symbol of an object,
- object renaming,
- connection removing,
- object deleting.

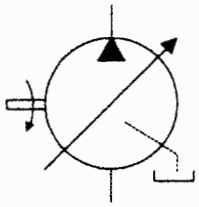
The program allows to change the graphical symbols associated to different objects and to define new symbols by means of the graphical symbols' editor as well as to create new object folders and to add new objects in existent catalogues.

The following picture shows an example of a simulation scheme drawn by means of the simulation program.

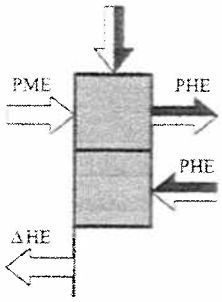


3. Types of Diagrams

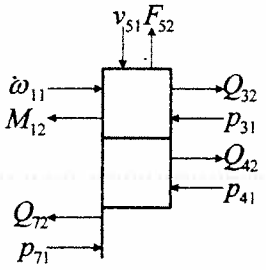
A researcher must have the adequate physico-mathematical methods for phenomena understanding in the elements' and installations' conception and auto-development stage. According to our point of view, these are realised in a graphical way to free the user from the very often difficult mathematical modelling operation. The design of these symbols is possible by means of the diagrams. We have designed four types of diagrams to help the user during the simulation process: ***Functional-Standard-Diagram, Energetical-Block-Diagram, Informational-Block-Diagram and Analogic-Block-Diagram***. We shall briefly describe each of them in the following.



i. The ***Functional-Standard-Diagram*** describes the functional behaviour of the installation (containing: engine, pomp, actuator, valves, lines, and so on) by means of some standardised symbols (at the left you can see a pomp viewed by means of the functional-standard-diagram).



ii. The ***Energetical-Block-and program flow Diagram*** shows the mechanical energy conversion into the hydraulic energy and the reverse process. One installation consists in different modules each of them containing maximum seven gates (at the left you can see a pomp viewed by means of the energetical-block-and program flow diagram).

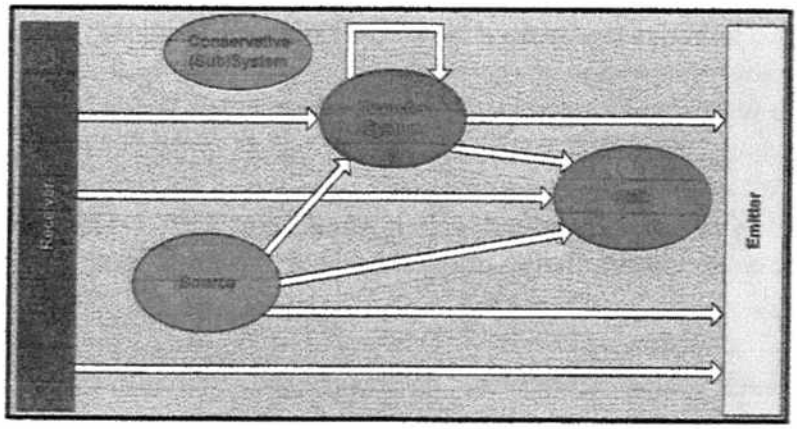


iii. The ***Informational-Block-Diagram*** is obtained from the energetical one by means of the energy decomposition into its components: flow and force variables. The variables are represented as orientated vectors, but their orientation during the operating time is not a fixed one, taking into consideration the harmonic variables involved in the process. The installation consists in associated modules, each of them containing maximum seven gates and each gate operate with maximum 2 variables (at the left you can see a pomp viewed by means of the informational-block diagram).

iv. The ***Analogic-Block-Diagram*** contains the exact linearised or simplified mathematical model of the installation and offers the mathematical flow of the information. The mathematical models are decomposed up to the elementary information layer. The informational hierarchy contains the following layers: group proprieties, semimodule, module, chain and installation. This diagram also contains all the mathematical operations with their variables and non-liniarities which describe the behaviour of the semi-module or module.

4. Mathematical Modelling

The Systemic Model Description (SMD) is referring to the design mode for a generalised physical system in the simulation program. The SMD concept is based both on the objective reality and on its scientifically perception as one can observe in the following picture.



Systemic Model Description

The Systemic Model Description assumes the Model Systemic dichotomy:
- According to the *Energy exchange with the environment:*
  Transfer - Exchange with the environment
  Conservative - No energy transfer with the exterior

- Terminal energy exchange blocks:
  Source - Specialised energy spring
  Sink - Energy flow terminal

- Surface energy exchange units (SEEU):
  Receiver - Input, Sensor - translation unit from environment to the inner world of the system
  Emitter - Output - energy translation unit to the environmental world

This description hides in its background the schemes' creation mechanism as compound systems. Except the surface subsystems, all subsystems can be on their turn aggregated. By means of this method, one can form a hierarchy with different depth description levels.

Observations:

- The number of Receivers or Emitters defines the number of system Input or Output gates, respectively.
- A *Source* has no inputs and one output at least. A *Receiver* has at least one input and no outputs.
- A *Conservative* system has no I/O gates. In order to exist an energy flow there must exist at least an energy *Source*.
- The highest hierarchical system is a *Conservative* system.

This „natural" way for the system description offers the possibility to implement the Object Oriented technique. Other consequences of this method consist in its independence from the field of study, and in the possibility to use the same hierarchy to solve another problem, taking into consideration problem's particularities though.

5. Numerical Simulation

The Integration Methods used by the program are:
- Euler
- Runge-Kutta 3rd Order
- Runge-Kutta 4th Order
- Runge-Kutta - Gill
- Gill

Each integration method can be chosen individually for each object or globally for the entire scheme. The selection of the optimal method highly depends both on each model's structure and parameters and on the integration step so it is not possible to recommend a specific method. During the new models' creation and optimisation the chosen solution will be the one which is the closest to the real behaviour of the given element for as many parameters as possible.

6. Conclusion

The presented program is based on an accurate mathematical theory. This theory allows the use of the same program to solve the simulation problems which appear in other fields of research.

7. Literature

Ionescu, Fl. Computer Aided Design of Hydraulic and Electrohydraulic Drive Installations. Proceedings of the 9th Trienal World IFAC Congress, Budapest, Hungary, Pergamon Press, 1st Volume, pp. 569-574, 1985.

Ionescu, Fl. Numerical Simulation of Large Hydrostatic Drive Systems. Proceedings of the 3rd Cairo Univ. Conf. on Current Adv. in Mech. Design & Production, Cairo, Egypt, December 28-30, 1985, pp. 135-142.

Ionescu, Fl., Stoffel, B. Contribution to the Automatic Generation of Mathematical Models for the Computer Assisted Analysis and Synthesis of Hydraulic Drive Systems. The 2nd International Conf. on Fluid Power, 19-21 March 1991,Tampere, Finland.

COMPUTATION OF NOISE RADIATED BY FLUIDS FLOW AROUND ROTATING BLADES IN THE SUBSONIC CASE WITH THE DECIVENT SOFTWARE

PEYRAUT FRANÇOIS

SCIENCES INDUSTRIES CONSEILS
14, Avenue de Sceaux
78 000 VERSAILLES
FRANCE
Tel. : (1) 39 49 05 27
Fax : (1) 39 02 77 23

1 - INTRODUCTION

In this paper, we present the DECIVENT software. This is an acoustic simulation's software devoted to ventilator blade's fast design. It allows engineers to compute the noise radiated by fluids flow around rotating blades in the subsonic case.

The acoustic calculation is based on the Lighthill's aeroacoustic analogy ([3]). The computation is made in free and far field. Considering the subsonic case, we only compute thickness and load noises.

Conciliating the necessities of rapidity and accuracy, a simplified approach based on radial equilibrium has been choosen for the steady force's computation. Numerical and experimental correlations have valided this approach.

DECIVENT has been developed with fortran 77.

2 - NOTATION

Bold quantities represents vectors. The other quantities are real or complex numbers following the context.

3 - THEORY

Assuming periodic the different noise's sources, we consider the tone noise which maximum intensity appears at blade's passing frequency.

Taking the Lighthill's aerodynamic theory, the wave equation for the acoustic pressure p can be written :

$$\phi P - 1/c^2\ \delta^2 P/\delta t^2 = \delta^2 T_{ij}/\delta x_i.\delta x_j - \text{div } \mathbf{f} + \delta Q/\delta t \qquad (1)$$

with c the sound speed and Q the flow rate. **f** represents the blade's force applied to fluid, T_{ij} is the Lighthill's stress tensor, ϕ symbolize the Laplacian operator and div the divergence operator.

The left hand member of equation (1) concern the acoustic wave's propagation and the right hand member is related to the various mechanichs of noise's generation.

The term $\delta Q/\delta t$, of monopolar type, represents the thickness noise due to fluid's volume moved by the blade's rotation.

The term div **f**, of dipolar type, is called load noise. It represents the forces applied to fluid by the blades.

Finally, the quadrupolar noise, associated to the term $\delta^2 T_{ij}/\delta x_i.\delta x_j$, is connected to the stress tensor which acts in the fluid near the blades. As it's a second order term at the considered Mach number, we can neglect it.

By using the Green's function G of the problem :

$$G(\mathbf{X},t,\mathbf{Y},\tau)=\delta(t-\delta-r/c)/(4\pi r) \tag{2}$$

with **X** the reception point, **Y** the emission point, t the reception time, τ the emission time, δ the Dirac's function and r the distance beetween the reception and emission points :

$$r=||\mathbf{X}-\mathbf{Y}|| \tag{3}$$

we express the acoustic pressure in integral form :

$$P_e(\mathbf{X},t) = \int_T \int_S m_v V_n \, dG(\mathbf{X},t,\mathbf{Y},\tau)/d\tau \, dS(\mathbf{Y}) \, d\tau \tag{4}$$

for the thickness noise and :

$$P_c(\mathbf{X},t) = \int_T \int_S f_i \, dG(\mathbf{X},t,\mathbf{Y},\tau)/dy_i \, dS(\mathbf{Y}) \, d\tau \tag{5}$$

for the load noise.

These integrals are computed over time (T) and spacial ranges (S=blade's surface). m_v represents the fluid's density and V_n the blade's normal speed.

In the thickness noise, we have only quantities connected to blade's geometry and machine's working. No aerodynamic computation is needed.

On the contrary, for the load noise, it's necessary to compute the aerodynamic forces applied to fluid by the blades.

DECIVENT is intended for beginning project's design. So time computation for fast iterative calculation must be short. These requirements exclude the using of tridimensionnal software of fluid's mechanic with the object of representing the aerodynamic forces with much accuracy.

That is the reason why we have choosen a radial equilibrium approach mixed with a Lowson type empiric estimation of unsteady forces ([1] and [4]).

This approach gives us entirely satisfaction because, even if discrepancies with experimental results reach occasionally 10 %, the general tendencies are respected.

Taking into account time's periodicity of the acoustic emission and assuming that the section is thin and that the acoustic sources are compact, the integration of (4) and (5) is analytic over the azimuth and the chord and numeric over the blade's wingspan.

4 - VALIDATION

DECIVENT has been valided for thickness noise by comparisons with numerical computations made by the ONERA on a two-blades helicopter's rotor in stationary flight ([2]). The comparison's results, on figure 1, are extremely conclusive.

Moreover, DECIVENT has been experimentally corraleted with ventilators of the CDV company. The discrepancies, oscillating beetween 1 % and 10 % (see figures 2 and 3), are reasonably good for a tool devoted to fast design at a beginning project. We must remark that this last correlation was made without knowing exactly test procedures.

5 - CONCLUSIONS AND FUTURE PROSPECTS

DECIVENT is a blade's acoustic design tool conceived for fast using by no-specialists. It allows to seek the best noise-flow compromise for rotating machines as ventilators, propellers, pumps, compressors ...

The future prospects can be classified on three levels.

DECIVENT has been soon valided by existing measures. But it could be interesting to refine the discrepancies beetween theory and experiment by making additional tests in which SCIENCES INDUSTRIES CONSEILS would be involved.

The second prospect's level concern the coupling of DECIVENT with the sound propagation in pipes.

The third prospect's level is the taking into consideration of aerodynamic phenomena's finest computation (wide band noise, vortex, drag noise ...)

REFERENCES

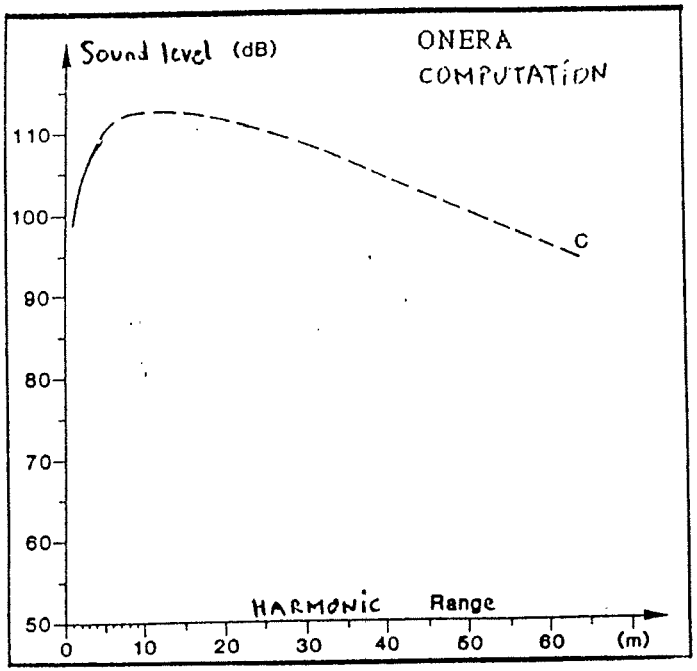
[1] "Etude Aéroacoustique des Spectres de Raies Générés par les Ventilateurs Axiaux en Régime Subsonique", J.P. Bridelance, thèse de Docteur Ingénieur, ENSAM Paris, 1982

[2] "Calcul du Bruit de Raies Emis par un Rotor d'Hélicoptère en Champ Lointain", Michel Caplot, thèse de l'Université de Technologie de Compiègne, 1985

[3] "On Sound Generated Aerodynamically. I General Theory", M. J. Lighthill, Proceedings of the Royal Society, Series A, Vol. 211, pp. 564-587, 1952

[4] "Theoretical Analysis of Compressor Noise", M. V. Lowson, J. Acous. Soc. Am., Vol. 47, n°1, pp. 371-385, USA, 1970

SPECTRUM ENVELOPE



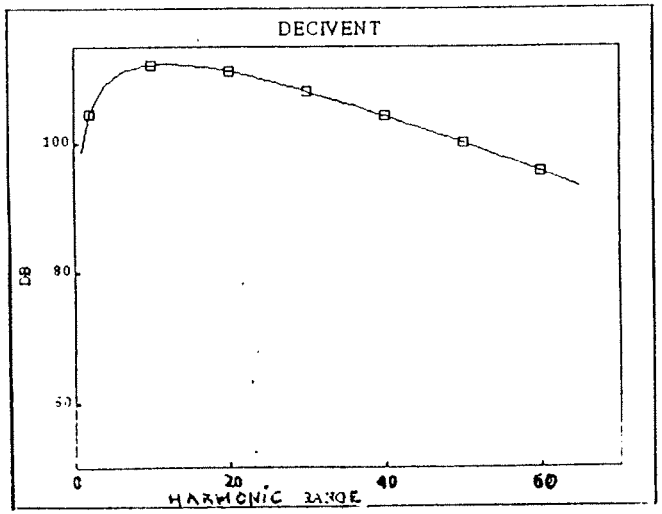


Figure 1

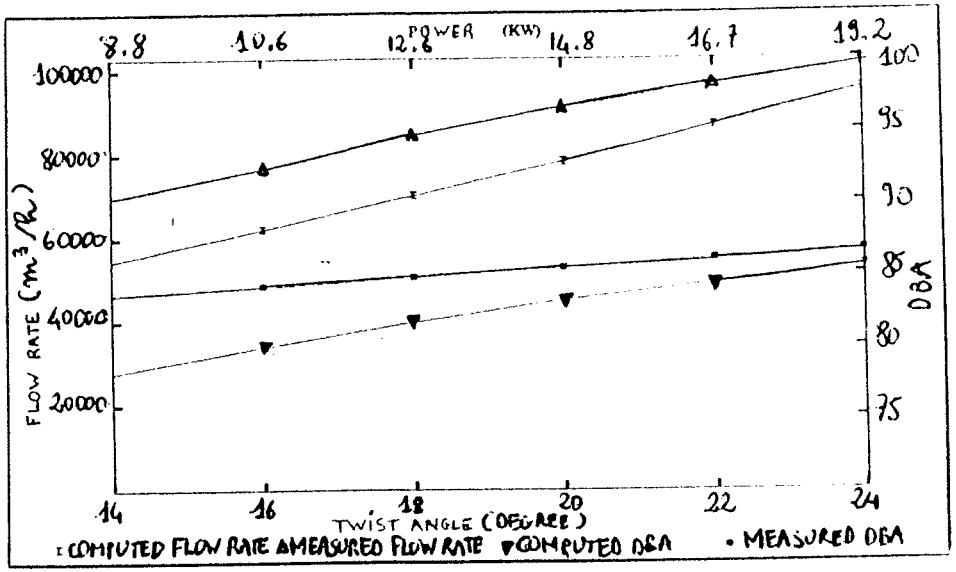


Figure 2 - 9-blades ventilator with a radius of 0.7 meters

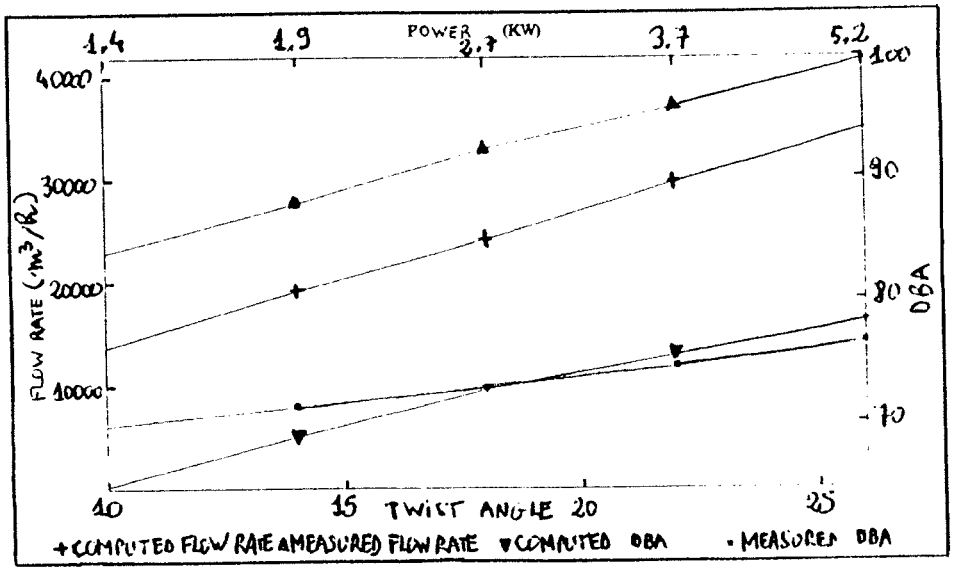


Figure 3 - 9-blades ventilator with a radius of 0.5 meters

Simulation Assistance for Analog Simulation*

Ralf Jendges
Institute for System Design Technology
German National Research Center for Computer Science
GMD-SET, 53754 St. Augustin, Germany
email: jendges@gmd.de

Abstract

Successful analog simulation is a necessary part of the design process of integrated circuits. On the other hand, the simulation can fail due to limitations of the circuit simulator or incorrect specifications. Often the error messages are only vague and the user is left without support to find the cause of the problem. This paper describes an approach to improve this situation by assistance for simulation. We present a toolbox named SAINT (Simulation AssIstaNT). SAINT supports the user with several ways to analyze the simulation problem, eliminate the problem and get the simulation running. The system can be used to locate simulation problems and to overcome convergence problems. The methods to localize simulation problems are based on partitioning and simulation of the parts. These methods and some examples are discussed in detail.

1. Introduction

Analog simulation is an integral part of the design process of integrated circuits. Circuit simulators are widely used to verify the electrical performance of circuits. In addition these simulators can be used to analyze microsystems on a behavioral level. The mechanical, optical or chemical parts of the system are taken into account by modelling the corresponding equations [Paa93].

At the same time, circuit simulation is one of the algorithmically most complex and CPU consuming parts of the design process. In spite of considerable progress in the development of simulators there are still limitations and the simulation can fail. The main reasons for a failing simulation are errors in the specification (input file) or some weaknesses of the numerical method. Typical examples for the latter are convergence problems related to the Newton-Raphson method and numerical instabilities of the integration method. Often the error messages give only vague information about the cause of the problem. Sometimes the error message is even missing. The user is left without support but with the task to make a diagnosis, eliminate the problem and get the simulation running. The probability for problems during the simulation increases with the size and complexity of the simulated system. At the same time, it becomes more difficult to analyze and overcome the problem. Thus, there is a strong demand to assist the user in overcoming problems which occur during the simulation. We will call this „simulation assistance".

Experienced users are familiar with special techniques to deal with simulation problems [Kie94]. These techniques are limited because they are performed manually and usually only modify the controlling parameters and options of the simulator. Knowledge-based systems were introduced to bypass simulation problems [Kel89] [Zan90]. These systems speed up the diagnosis but the modifications are still restricted to controlling parameters and parasitic elements. The advantages of topological changes like partitioning are not used. Recent work applies homotopy methods for the special problem of robust computation of dc operating points [Mel93]. The corresponding simulation package is proprietary.

This paper describes a novel approach to simulation assistance. First, we present a toolbox named SAINT (Simulation AssIstaNT). SAINT supports the user with different ways to analyze the simulation problem and improve the simulation. Secondly, we study in detail our methods to localize simulation problems. At last, we state some results obtained with the implemented localization techniques.

* This research is supported in part by the German Ministry of Education, Science, Research and Technology (BMBF) under grant 13MV0183

2. The SAINT system

The scheme of the proposed diagnosis procedure (fig. 1) is based on the steps:

- analyzing the simulation results
- extracting certain data
- selecting an appropriate strategy
- modifying the simulator input according to this selection
- repeating the simulation and analyzing the new results

The user controls this iterative procedure which can be repeated successively to get the desired diagnosis result.

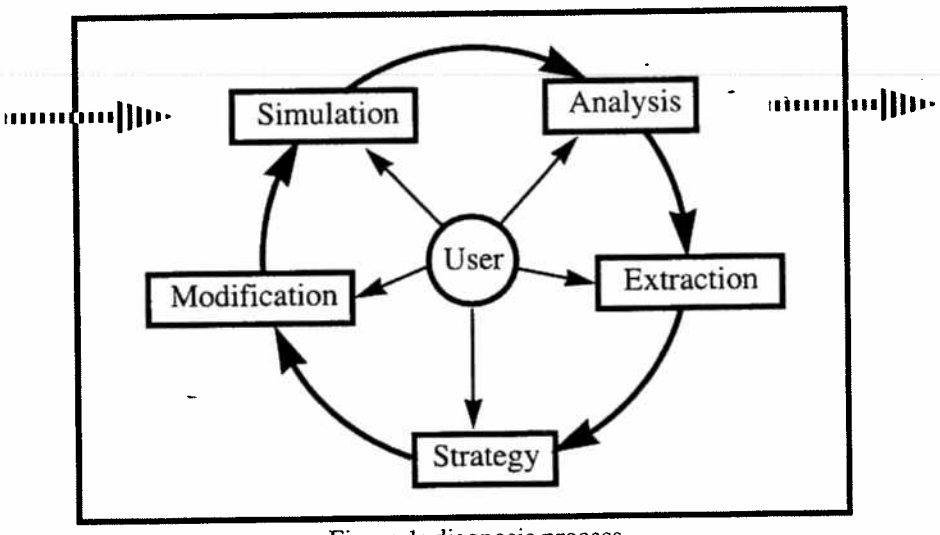


Figure 1: diagnosis process

The purpose of the simulation assistant SAINT is to offer a set of possibilities to obtain additional information to overcome simulation problems. The available techniques are combined in a toolbox to make the diagnosis more efficient. Currently SAINT helps to locate simulation problems and to improve dc convergence. The techniques to find the dc operating point are based on continuation methods [Deh94]. The methods to localize problems will be discussed in the next section.

To improve the simulation one can modify the input or the source code of the simulator which is used. We decided to change only the input file but not the simulator itself. Information about the improvement due to this changes is extracted from the simulator output. In this sense SAINT is based on a black box approach to control the simulator. This approach has essential advantages. It is easier to interface the simulation assistant to different simulators and to integrate it in an analog system design environment [Deg89]. This is a useful property concerning analog design automation.

SAINT has a graphical user interface. The user can select the input file, different kinds of modification and controlling parameters. SAINT performs corresponding actions like parsing the input file, partitioning of the system, generating modified input files, running the simulator, analyzing the simulation output and displaying results.

The diagnosis is partially-automated to minimize the necessary effort of the user. This also decreases the probability of new errors created by manual operations. The time spent for the design process can be reduced this way.

SAINT is implemented as an experimental system for the simulator SPICE3 [Joh92]. Therefore it is easy to adapt SAINT for the numerous simulators which accept the SPICE input format. The package is coded in C/C++ and the graphical user interface is written with Tcl/Tk [Ous94].

3. Localization of simulation problems

To localize problems is one important way to assist in understanding and solving simulation problems. SAINT has the capability to partition the system and repeat the simulation for the parts automatically. Successful or failing simulation of the parts indicates the local or nonlocal character of the problem.

The localization techniques were developed for the transient analysis. But dc convergence problems can be investigated too if they are transformed manually to transient problems. This is easily done by ramping all sources.

Partitioning is a well-known method for the design of integrated circuits [San77][Joh91]. Our algorithm is based on a modified node-tearing method [San77]. The costfunction is defined as the number of adjacent elements. Elements are combined to parts in such a way that the costfunction becomes minimal. The partitioning can be controlled with parameters for minimal and maximal part size. It is possible to specify „global nodes". The adjacency relations of this nodes will be ignored by the partitioning algorithm. This is useful to handle e.g. the power supply.

SAINT offers a choice between two types of coupling for the parts. In the first case we use data from the failing simulation of the original system to apply signals to the connections which are cut [Jen94]. The extracted data are the voltage waveforms up to the time when the simulation aborts. Linear approximation is used to predict the signals for the subsequent time interval. The parts are simulated separately without exchange of data between them. So, strictly speaking, this kind of coupling is a decoupling.

The second type of coupling is a waveform relaxation method [Rue87]. Controlled voltage and current sources exchange the signals between the parts. The simulation of the parts is performed iteratively. Figure 2 shows the corresponding algorithm where the subscript i denotes the part index and the superscript k denotes the iteration count. While the first type of coupling produces an error due to the linear approximation, the second coupling is free from this disadvantage. On the other hand, the first technique leads right from the beginning to an exact solution in the time interval up to the failure of the original simulation.

initialize all waveforms y_i^0
$k := 1$
while (*iteration limit not reached*)
{
  $i := 1$
  while ($i <=$ *number of parts*)
  {
    compute waveforms y_i^k *for part* i *for* $t \in [0, t_{max}]$ *while waveforms for other parts* j *are fixed:*
    $y_j = y_j^k$ *for* $j < i$
    $y_j = y_j^{k-1}$ *for* $j > i$
    $i := i + 1$
  }
  $k := k + 1$
}

Figure 2: basic waveform relaxation algorithm

The simulation of the parts can be repeated for different partitionings to test the system. If we detect a part with simulation problems, we can split this part to locate the problem more precisely. This procedure can be repeated successively. With every step the actual part becomes less complex. Therefore, even a single successful step will be useful because analyzing the failure is simplified.

4. Examples

The localization methods were already applied successfully for various examples including resistor networks, a SRAM and a BIC-monitor.

We present here the application for a simple resistor network to illustrate the basic concept. The resistor network consists of two voltage sources V_1 and V_2, eleven equal resistors and one resistor R_t which is time dependent (fig. 3). R_t is modeled by a subcircuit such that $R_t = a \cdot t + b$ with $a < 0$ and $b > 0$. Thus, there is a zero at the time $t_0 = -b/a$ and the simulation of the entire circuit fails.

The choice of appropriate controlling parameters forces the algorithm to decompose the network into two parts. The dashed lines in figure 3 indicate the partitioning. These parts **A** and **B** are simulated automatically. We observe that the simulation aborts for part **B** which contains the resistor R_t. Hence, the attempt to determine the

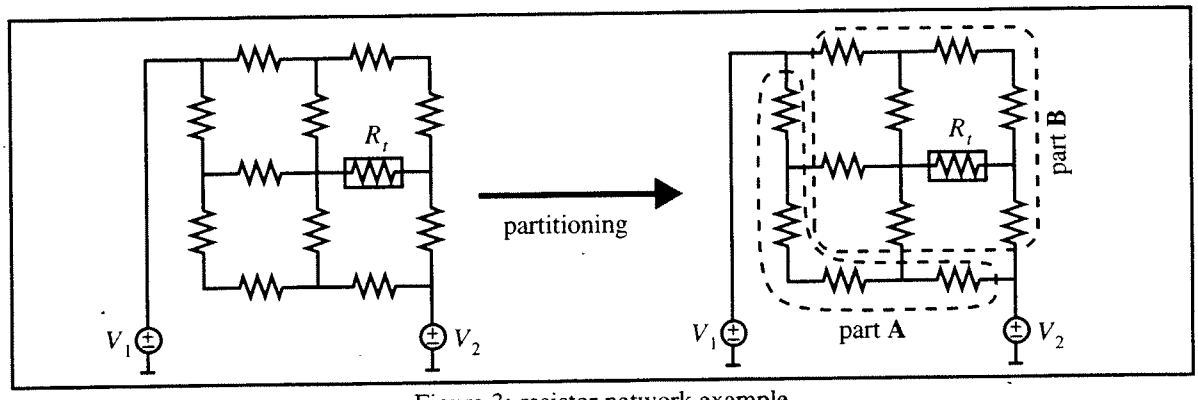


Figure 3: resistor network example

part which causes the problem was successful. The same technique can be performed successively if the system is more complex.

5. Conclusion

A novel approach to analyze and overcome problems which occur during simulation has been presented. The simulation assistant system SAINT provides several useful capabilities including localization of problems and improvement of dc convergence. Costly diagnosis processes like partitioning, modification of input files and iterations are automated to reduce the diagnosis time.

The methods to locate simulation problems are based on partitioning and simulation of the parts. A waveform relaxation technique allows to perform the simulation of the parts iteratively. This procedure can be repeated for different partitionings to detect faulty parts and to localize more precisely.

Current work is dedicated to a multilevel generalization of the waveform relaxation technique. This will improve the convergence of the iteration process. Further investigations are focusing on the approximation of solutions which can not be computed for the entire circuit.

References

[Deg89] M.G.R. Degrauwe et al., „Towards an Analog System Design Environment", IEEE Journal of Solid-State Circuits, Vol. 24, No.3, June 1989

[Deh94] M. Dehlwisch, R. Jendges, „Fehleranalyse für die Analogsimulation", Tagungsband 6. ITG-Fachtagung Mikroelektronik für die Informationstechnik, Berlin, 1994

[Jen94] R. Jendges, M. Dehlwisch, „Partitionierung und Resimulation zur Analyse simulationskritischer Systeme", Tagungsband 9. ASIM Symposium Simulationstechnik, Stuttgart, 1994

[Joh91] W. John, W. Rissiek, K.L. Paap, „Circuit Partitioning for Waveform Relaxation", Proceedings European Design Automation Conference EDAC, 1991

[Joh92] B. Johnson et. al., „SPICE3 Version 3f User's Manual", University of California, Berkeley, 1992

[Kel89] T.M. Kelessoglou, D.O. Pederson, „NECTAR : A Knowledge-Based Environment to Enhance SPICE", IEEE Journal of Solid-State Circuits, Vol.. 24, No.2, April 1989

[Kie94] R.M. Kielkowski, „Inside SPICE, overcoming the obstacles of circuit simulation", McGraw-Hill, 1994

[Mel93] R.C. Melville, L. Trajkovic, S.-C. Fang, L.T. Watson, „Artificial Parameter Homotopy Methods for the DC OperatingPoint Problem", IEEE Trans. on CAD, CAD-12, No. 6, June 1993

[Ous94] J. K. Ousterhout, „Tcl and the Tk toolkit", Addison-Wesley, 1994

[Paa93] K. L. Paap, M. Dehlwisch, R. Jendges, B. Klaassen, „Modeling Mixed Systems with SPICE3", IEEE Circuits & Devices, Vol. 9, No. 5, September 1993

[Rue87] A. E. Ruehli (Ed.), „Circuit analysis, simulation and design", Part 1 and 2, North-Holland, 1987

[San77] A. Sangiovanni-Vincentelli, L. Chen, L.O. Chua, „An Efficient Heuristic Cluster Algorithm for Tearing Large-Scale Networks", IEEE Trans. on Circuits and Systems, CAS-24, No. 12, Dezember 1977

[Zan90] M. Zanella, P. Gubian, „A Learning Scheme for SPICE Simulations Diagnostics", Proceedings IEEE International Symposium on Circuits and Systems, New Orleans, 1990

Marketing in Simulation

H. G. Unseld
Unseld + Partner
Business and Marketing Consulting • Simulation!
Lerchenfelderstraße 44/9, A-1080 Vienna, Austria
E-mail: hgunseld@unseld.co.at

1. Introduction

In simulation we frequently find ourselves debating the „obvious" market demand and the „obvious" right product we can offer to satisfy this demand. Unexpected low sales, however, show that „obvious" facts do not necessarily end up with happy simulationists and satisfied customers. Therefore, the „obvious" question is: What are the elements of successful marketing for simulation in industrial and other profit-oriented organisations?

Marketing is applied in virtually all areas, where a demand meets a product. This process always requires human interaction and involves all steps of the communication cycle. Simulation can be considered as an art of communication, too. Marketing in simulation is a bundle of elements describing the process, where a person or a company is ready to pay money for getting a better insight into various fields, which usually gain more individual importance to the customers, if analysis and intuition do not work any more. This paper describes a new approach based on experience and reflections among marketing experts.

2. Needs and Conditions of Profit-Oriented Organisations

The evolution and changes of modern industrial management structures, i.e. Business Reengineering, leads to organisations with a strong focus on core business and core functions. Support functions are reduced to their minimum. Managerial functions imply extensive information technology structures.
For people working in these organisations it is vital to know how to organise themselves in order to support the core functions. To perform their jobs well the required knowledge of the people concerned has to be within their reach and span of competence. Communication trainings and workshops are meant to support the transition phase to the „new" organisation. Their objectives are in tune with the company's objectives. Their direct goals are clear, concise and easy to understand; their contributions are measurable. Their relation and access to information technology (IT) determines their possible degree of integration into the overall company strategy.
The IT system represents the company's backbone and provides state-of-the-art services and performance. This scenario describes an „ideal" situation. The management will undertake all actions to achieve these goals. And so far, all is perfect. This is the world, where the simulationist brings in his ideas to challenge industrial ambitions.

The reality, however, deeply contrasts depending on the history of the company, the type of industry and the competitive environment.

In all projects with profit-oriented organisations a simulation provider faces more or less the following facts

- increasing time pressure on professionals
- management afraid of chaos and information security: subjective high complexity
- inconsistent IT structures and data, lack of documentation and experts
- insufficient IT knowledge and user trainings, sometimes combined with a lack of professional capacity

Each simulation application project, however, requires precise data, an in-depth understanding of the behaviour of the modelled reality (=company) and a minimum knowledge of the process itself. The obvious gap between idealistic and realistic view must be filled in by confidence. The level of confidence needed for providing the information leads to an unmatched heavy burden on even small simulation projects. To handle simulation implementations successfully, a complete simulation package consisting of a bundle of software and service elements is needed, with more emphasis on service - from the confidence point-of-view.

As an interesting fact, IT systems providers find themselves trapped with the gap of real and perceived core complexity in „crafts-men-type" and „professional" customers (see Fig 1). A given degree of complexity is under-estimated by „professional customers" and overestimated by „crafts-men-type customers". The higher the degree of complexity will grow, the bigger the span of subjective complexity will become. This explains, why simulation as a „complexity-uncovering-product" has to fully rely on at least *one* person at the customer knowing the core complexity.

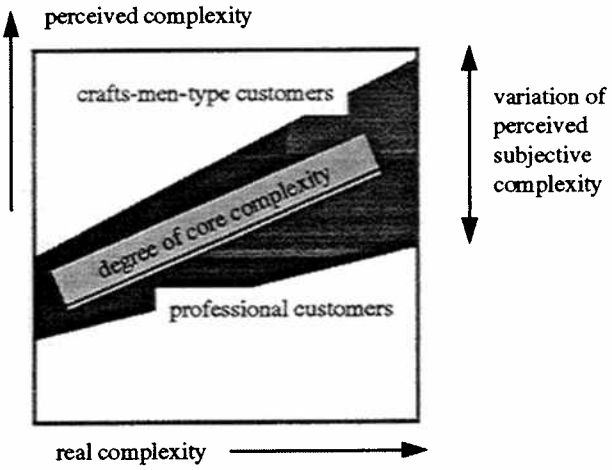


Fig. 1: Real and perceived core complexity

Is investment in simulation worthwhile? Starting with simulation technology can generally be compared with the early stage of the life cycle in Computer Aided Design in terms of know how and expertise as well as soft- and hardware investments. CAD improves the design process for new products and stimulates indirect revenues. Simulation improves the process to better and faster utilisation of the permanent cost-raising resources in a changing company environment. It primarily contributes to cost reduction and new service requirements. Its Return-on-Investment as an „design and optimisation tool" is determined by the inherent potential of the investments done, or to be done.

3. Success Factors for Simulation

Success factors are not a guaranty for success. They are rather ideas for a specific mind set.

Efficient tools. The power and flexibility of modern state-of-the-art object oriented tools in the hand of IT experts, communication experts and simulation experts guaranties a customer the best-of-its-art solution. A customer embarking on a simulation project never accepts less.

Confidence. Getting known - often detailed - inside and confidential information and delicate subjects, a customer needs to trust the simulationist right to begin with. No confidence - no data - no simulation! Since substantial investments are involved confidence must be granted for a long time.

Advocate. In every organisation simulation involves many people in many departments. The majority at the end has to vote PRO SIMULATION. In this context key account marketing and an internal advocate will help.

A new marketing approach must support all success factors. The challenge remains in learning individually which factor is prime and which is second.

4. Marketing Matrix for Simulation

The four classical elements of a marketing matrix are:

- Product mix: describing the product and its features
- Market communication mix: describing the communication between product and market
- Distribution/supply mix: describing the product's way into the market
- Price mix: describing the price conditions for supply and sales

A matrix for simulation will add the timing aspect, the product life cycle aspect, the service aspect and the technology transfer.

Timing. The interaction between customer and simulation provider differs during the phases of a project. Gaining confidence with the appropriate individual positioning of simulation is mandatory during the pre-sales phase. The marketing activities during the project phase will focus on supporting the confidence received on an individual basis. Both activities support the lifelong relationship with a customer.

Product life cycle. Simulation by its modelling nature as well as the relevant soft and hard ware enjoy short and long life cycles, both being determined by the user and the user's community and their relationship; marketing needs define the elements how to describe it. The product/market-matrix for simulation software products outlines possible conflicts.

Technology transfer. The cross functionality of simulation requires efficient communication on most critical and serious technical and organisational topics. A transfer of know how and technology is inevitable and offers synergy especially for demanding customers. This process must be structured and governed. Marketing has to value it.

Service and confidence. Simulation providers always deal with sensitive data. Sometimes even on items, where the internal IT know how is not sufficient. This provides another opportunity to serve the customers needs. The ultimate service goal is to kick any hurdle away for starting a simulation project to the benefit of both parties.

The major factor is the timing synchronisation, yet the timing of the relation between simulation provider and customer on simulation topics.

5. Simulation Performance Package SPP

Simulation providers dealing with the above listed facts and with the established demand for a simulation package usually apply classical marketing methods for advanced software marketing or expert and consultants services marketing. In this paper a new approach is shown, integrating all elements to a simulation performance package SPP.

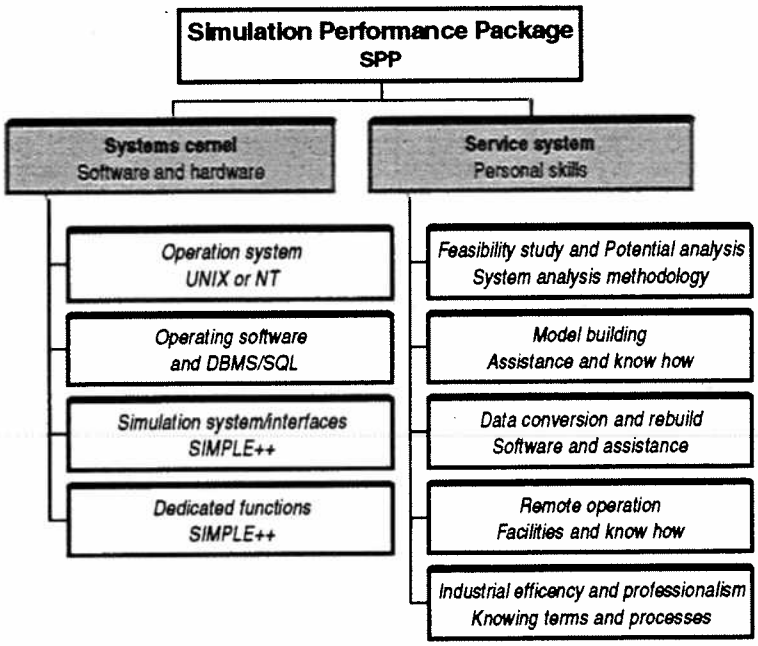


Fig. 2: Simulation Performance Package for a Manufacturing Site

The package consists of two equally important parts: the systems core functions, and the associated service system. Under the term *„systems core functions"* we understand all system soft and hard ware products dealing with simulation: ranging from dedicated operating systems to very special routines i.e. for optimising the throughput in a cutting machine. The relevant simulation software is an integral part of this system. The *„service system"* consists of all available services: ranging from presales software performance comparisons to provision of experts' consultation for very specific technologies. Fig. 2 shows a set of elements of a classical simulation implementation in a manufacturing site based on SIMPLE++. The goal of marketing is to verify the customer's match, and to evaluate the differences during the whole product-in-customer life cycle. Depending on the maturity of the customer the package will be abopted accordingly.

6. Conclusion

Modern simulation technology and a concerted approach according to a proposed new marketing mix matrix will help the simulationists getting closer to customers from profit-oriented organisations. The challenge will remain with the persons dealing with the systems and how they are able to contribute to their customers wealth. This will allow them to generate more revenues for their own business and contribute to a wider acceptance of the simulation idea.

References:

Belz, C. et al (1991): Erfolgreiche Leistungssysteme - Anleitung und Beispiele, Stuttgart

Zerr, K. (1994): Systemmarketing - Die Gestaltung integrierter informationstechnologischer Leistungssysteme als Aufgabe des Marketing, Wiesbaden

Simulation and Animation Software From Wolverine GPSS/H, PROOF, and SLX

Robert C. Crain
Wolverine Software Corporation
7617 Little River Turnpike, Suite 900, Annandale VA 22003, USA
(703) 750-3910 • FAX (703) 642-9634

Wolverine Software has been providing extremely efficient, high-quality simulation software since 1977. This paper presents an overview of Wolverine's current simulation and animation products, GPSS/H and PROOF Animation, and of its forthcoming next-generation simulation product, SLX.

The widespread success of GPSS/H stems both from the superiority of its original design and from years of improvements and enhancements. Although it requires some programming-style effort, GPSS/H provides a natural modeling framework that can be readily used *without* extensive programming experience. It is equally well-suited for modeling simple systems and for modeling large, complex systems. GPSS/H is presently applied worldwide in modeling manufacturing, distribution, transportation, hospitals, computers, telecommunications, and many other types of queueing systems.

Animation is often considered a requirement for a simulation study because it is a meaningful, proven way to show results to an audience with varied backgrounds. Proof Animation is a powerful general purpose animation tool. It is not tied to a specific area, application, or simulation language, nor is it limited in the size of the systems it can animate.

SLX is Wolverine's next-generation successor to GPSS/H. While SLX retains some of the tried-and-proven fundamental concepts of GPSS/H, such as its Transaction-flow world-view, Facilities, Queues, and Storages, SLX is far more than a new implementation of GPSS/H. SLX is a layered modeling system with powerful extensibility mechanisms which facilitate the development of higher-level, graphically oriented, application-specific modeling tools.

GPSS/H

GPSS/H is a discrete-event simulation language. Models are conveniently developed in a text-based environment, and subsequently compiled *directly into memory* and executed. Rapid prototyping and iterative model development are encouraged by *exceptionally* fast compilation and execution.

GPSS/H follows the intuitive and natural *process-interaction* approach to modeling. The modeler specifies the sequence of events, separated by lapses in time, which describes the manner in which "objects" flow through a system. A GPSS/H model thus resembles the structure of a flowchart of the system being modeled. This intuitive modeling approach contributes greatly to the ease and speed with which simulation models can be built. After the model has been built, the process representation is executed by GPSS/H, with the activities of "objects" *automatically* controlled and monitored.

An "object" in a GPSS/H model might be a patient, a telephone call, or any other type of discrete entity. The representations of these entities in GPSS/H are called *transactions*. As the model executes, many transactions may be flowing through the model simultaneously—just as many "objects" would be moving through the real-world system. In addition, multiple transactions can execute GPSS/H model statements at the same instant in time *without any special action required of the modeler*.

The focus of many simulation projects is the use of system resources such as people, machines, conveyors, computers, physical space, and so on. In a GPSS/H simulation model, transactions ("objects") *compete* for the use of these system resources: as transactions flow through the process representation, they *automatically* queue up when unable to gain control of a necessary resource. The modeler does not need to specify the transaction's waiting time or its queueing behavior. Hence, the passage of time in a GPSS/H model can be represented *implicitly*, as in the case of a part waiting for a machine to be free, as well as *explicitly*, as in the case of a part being processed by a machine.

As is the case in most real-world systems, a GPSS/H model may consist of multiple processes operating simultaneously. Furthermore, each process may in some way affect the other processes in the system. GPSS/H provides the capability for multiple parallel processes to interact with each other *automatically*. Transactions ("objects") may be sent between processes; they may control or share common resources; or they may influence the (global) operation of all processes.

Important Features Of GPSS/H

Several unique characteristics make Wolverine's GPSS/H an ideal choice for a general simulation environment. A key feature of GPSS/H is the ***conceptual flexibility*** to model a wide range of *different types* of systems: any system that can be described as a process *flow*, with objects and resources acting upon each other, can be modeled. This may include people on a mass transit system, tasks in an office environment, or data flow within a computer network.

Definition flexibility is also provided within the language: complex math formulas, expressions, and constants can be used virtually anywhere in the model. To promote *model readability*, elements and entities may be specified by names instead of numbers. *Basic simulation output data,* such as queueing and service statistics, are *automatically* provided without any programming.

GPSS/H also allows ***flexibility in the selection of hardware platforms:*** GPSS/H runs on PCs, SUN SPARC workstations, VAX/VMS computers, and mainframes. On the PC, GPSS/H Professional runs as a true 32-bit application under DOS, Windows, OS/2, or Windows NT, providing tremendous speed as well as model size that is limited only by the computer's available memory.

The ***file and screen I/O*** *built into* GPSS/H provide a variety of ways to get data into a model and to produce custom output. A very intuitive "picture" type of format specification, which follows the "what you see is what you get" convention, is used to specify custom output.

A complete ***scripting language*** is available to construct experiments and control model execution. The experimental specifications and parameters, like any other model data, can be read in from a data file or from the keyboard if desired.

The GPSS/H ***Interactive Debugger*** conveniently provides for rapid model development and verification. The debugger provides a "windowing" mode that displays source code, model status, and interactive user input as the model runs. GPSS/H also provides *"just in time"* debugging. Even when the debugger was not invoked at the beginning of a run, if an error occurs during model execution, the debugger automatically "pops up" to allow the modeler to explore the cause of the error.

Features Of GPSS/H Release 3

GPSS/H is continuously improving and evolving. A few of the more significant recent additions to the widely-used GPSS/H Professional version are:

- The BLET Block and the LET Statement can now be used to assign a value to ***any*** GPSS/H data item. Unless you need the rarely used range-type assignments, *there is no longer any reason to use* the ASSIGN, SAVEVALUE, and MSAVEVALUE Blocks.
- GPSS/H now supports built-in random-variate generators for 23 additional statistical distributions (26 in all), and GPSS/H Professional now comes bundled with Unifit II, the highly-regarded distribution-fitting software from Averill M. Law and Associates.
- GPSS/H Professional supports user-written external routines in both C and FORTRAN.
- CHECKPOINT and RESTORE statements allow a model to save its state at a given point during execution, then make repeated runs using that state as the starting point.
- The SYSCALL statement and the BSYSCALL Block, which take an operating system command line as an operand, allow a running GPSS/H model to *shell out* to the operating system to run other software.
- The INSERT compiler directive allows model code to be read from multiple files during compilation.
- The operations that can be performed on Transactions in a User Chain have been extended. New SCANUCH and ALTERUCH Blocks allow examining and changing the Parameters of such Transactions without having to UNLINK and reLINK them.

Run-Time Versions Allow Economical Model Distribution

Sometimes a model is intended to be used by many people, each of whom must have a copy of the simulation software in order to run the model. Because of the number of users, the cost of the simulation software itself may render the project too expensive. Wolverine's Run-time GPSS/H offers a solution.

Run-time GPSS/H is identical to Wolverine's 32-bit GPSS/H Professional for personal computers, except that it costs less and can only run models which have been previously compiled with the regular Professional version.

Security is another important feature provided by the run-time version. Since only *pre-compiled* models can be run, the end user cannot view or edit the model "source" code. Hence, confidential models can be safely distributed.

PROOF ANIMATION

Proof Is A *General Purpose* Animator

Proof Animation can be used to animate the full range of applications, from areas such as Business Process Reengineering to the classic applications such as health care, manufacturing, and traffic. The size of the system to be animated is not an issue when using Proof Animation.

With Exceptional Features And Performance...

Proof Animation uses vector-based geometry to provide a large animation canvas and the ability to zoom in or out while maintaining crisp, clear images. Proof Animation's features include post-processing for maximum performance, built-in drawing tools and CAD import/export for ease of creating animation layouts, dynamic bar graphs and plots used for displaying statistics, a multi-windowing display, a unique presentation-making capability, and smooth, realistic motion for animations regardless of the size, complexity, or application. All versions run as 32-bit applications that require only a 386 or better CPU, a

math coprocessor, and at least a VGA-compatible video card.

Demo versions of animations can be prepared using the optional Demo-Maker feature. Copies of the demo can be reproduced and distributed free of charge and can be viewed by anyone. No licensed copy of Proof is needed to *view* an animation prepared with the Demo Maker.

And An *Open* Architecture

Proof Animation was built not only to work easily with Wolverine's GPSS/H simulation software, but also to provide affordable, powerful, easy-to-use animation software to modelers who use other simulation and programming languages. Proof Animation is driven by ASCII files. As a result, any software capable of writing ASCII text files can be used with Proof Animation.

In contrast to Proof, most animation software from other vendors is directly coupled to their simulation software. In other words, one cannot use *their* animation software without also using *their* simulation software. Worse yet, in some cases the simulation and animation software are sold only as a pair, so both must be purchased regardless of the needs of the user.

Although vendors of such tightly coupled packages often claim that their approach is the *only* way to add animation to a simulation, Proof Animation provides a mix-and-match option that allows software selection to be based on optimal functionality and price.

Post-Processing Makes The Difference

Post-processing means that the animation runs *after* the simulation has executed.

Three great advantages result from the post-processing approach. First, PC hardware is not shared between the simulation and the animation. This leaves the entire CPU for running the animation. Second, it provides the abilities to jump back and forth in time during the animation playback, to speed up or slow down the viewing speed, or show all or a specific portion of an animation. These features make it easy to investigate unusual system behavior or highlight points of interest. Third, you don't have to run the simulation model to see the animation. With small (or demo) models, this may not seem important. *But if your model requires, say, ten minutes or an hour to run and you need to show your boss (or customer) a certain part of the animation again, the importance of post-processed animation becomes quite apparent.*

Vector-Based Geometry Gives Realism, Allows Direct Import of CAD Files

One of the advantages of *vector geometry* is that an animation can be much larger than a single screen. With the ability to zoom in or out and pan side to side, larger layouts are easily navigated to show the "big picture" or zoomed in to whatever level of detail is necessary. Vector-based geometry also allows moving objects to realistically *rotate* around corners.

Another advantage of vector-based geometry is that if a CAD drawing already exists for the system to be animated, the effort of (re)drawing the system layout can be avoided. Proof Animation's built-in CAD Import/Export feature can convert industry-standard .DXF files into Proof Animation layout files, *and vice versa.* Credibility of the study is enhanced when viewers see the system's animated behavior taking place on a familiar CAD drawing of the system.

Smooth Motion Is Critical To Realism

The maximum-performance design of Proof Animation achieves *very* smooth motion by updating the screen 60-70 times per second. Other software can often sustain rates of only 5-10 updates per second. Objects that move smoothly across the screen are dramatically more realistic than those that jump across the screen.

Smart Paths Handle Accumulation

Proof Animation provides two kinds of motion: *absolute* and *guided.* Absolute motion causes an object to be moved directly between two points. In contrast, guided motion follows a predefined path. Such paths play an especially important part in transportation, product flow, and material-handling animations.

Using paths is very simple because Proof Animation does all the work. Once an object is placed on a path, it will follow that path until it visually comes to rest at the end of the path or until it is placed elsewhere or destroyed. All objects traveling on the same path can be stopped simultaneously and resume movement at a later time. Paths provide real animation power.

Accumulating paths provide even greater power. On accumulating paths, Proof Animation reflects physical reality by *visually* queueing objects when bottlenecks occur. This often makes a simulation model of the system much simpler to construct, because such queueing need not be explicitly represented in the model. Accumulating paths can be used to represent certain types of conveyors, cars at a traffic signal, customers in bank lines, and so on.

Multiple Display Windows Provide Complete Flexibility

The animation screen can be divided into separate windows. Within each window, the view can be independently manipulated using zooms, pans and rotations to include all or any portion of the animation canvas.

With this feature it is easy to maintain a window containing updated statistics in constant view while panning and zooming to different areas of the layout.

Proof Makes Presentations Easy

A professional-looking presentation can be built *completely* with Proof Animation. Its Presentation Mode lets users specify scripted sequences consisting of bit-mapped screen images or slides, full animations,

and/or segments selected from full animations. These presentation elements can be linked together using fades, dissolves, and other special effects to produce a polished presentation.

Slides can be created directly in Proof Animation or in any software package capable of exporting industry-standard ".PCX" image files. There are many such packages available, and virtually all of them can produce very high-quality charts, graphs, and slides. Proof Animation can both read and write these .PCX images, so one can save Proof Animation screen images as .PCX files and incorporate them into presentations as slides when animation is not needed.

Presentations can also incorporate selectable menus defined by the presentation developer. These menus can be set up by topic, giving the viewer or presenter complete control and flexibility of what to show.

SLX

SLX is a completely new layered modeling system, designed with the benefit of Wolverine's nearly 20 years of experience in the simulation industry. Its individual layers are as follows:

- *Level 0* is the SLX kernel, a language loosely modeled after the widely popular C language, including a run-time library. Unlike C, it provides complete run-time error-checking. It also directly supports a number of primitives (such as a generalized "wait until") essential to simulation.
- *Level 1* consists of data structures, subroutines, operators, macros, and statement-definitions, all written in SLX. These provide additional simulation primitives to support higher levels of SLX. A user can augment Level 1, adding similar capabilities of his or her own design.
- *Level 2* is the "new" GPSS/H, retaining many of the basic concepts while implementing them in a far more general way.
- *Level 3* is the level at which application-specific packages will be developed, e.g., manufacturing, telecommunications, health care. Wolverine will develop some of these packages, but we anticipate that many will be developed by third parties.
- *Level 4* will contain very high level, graphically oriented packages for use by non-simulationists.

Perhaps the greatest strength of SLX lies in its novel extensibility mechanisms, which facilitate the construction of higher layers from components contained in lower layers. Users of upper layers can ignore lower layers. However, if their requirements are not met at a given level, they can move down one or more levels, *without exerting extraordinary effort and without losing protection against potentially disastrous errors.* For example, a user wishing to add a new statement to the GPSS/H level of SLX has access to the same mechanisms we used to implement the built-in statements — a hallmark of the *open* architecture of SLX.

SLX retains an important concept from GPSS/H: its Transaction-flow world-view. This world-view has proven to be extremely flexible and powerful, *yet easily learned,* and lends itself well to graphical representation.

A second important concept retained from GPSS/H is its *total run-time error-checking* and complete reproducibility of run-to-run results. In a simulation, events take place in complex (usually random) circumstances, thus one expects to have unanticipated events. (That's a major reason for simulating: to have such events occur in a model instead of in the real system.) Determining the cause of an unexpected event can be quite difficult in a complex simulation. It would be totally unacceptable to allow undetected "programming-type" errors, such as referencing beyond the end of an array, to muddy the waters further. Such errors must be unfailingly trapped.

SLX is not a truly object-oriented language, although it has been influenced by the object-oriented programming (OOP) paradigm and it does make heavy use of objects. In SLX, objects are used in two ways. *Passive* objects are used for modeling entities which have no "executable" behavior. For example, a parking lot could be modeled as a passive object. *Active* objects have executable behavior patterns. Customers in a supermarket are a good example of entities that would probably be modeled as active objects. SLX active objects are roughly equivalent to GPSS/H transactions.

SLX objects can have a number of *standard properties.* All standard properties are comprised of explicitly identified sections of executable code. The *initial* property is invoked when an object is created. The *final* property is invoked when an object is about to be destroyed. The *actions* property specifies the behavior pattern for an active object. The *clear* and *reset* properties specify what should be done to an object when statistics are cleared or reset. The *report* property is used for the obvious purpose.

SLX presently operates in a highly interactive, window-based environment which features a fully integrated editor, compiler, and debugger, with work underway on tools to support building models graphically. On "typical" machines, "typical" models can be compiled at rates exceeding 1000 statements per *second.* Both compilation *and execution* errors are highlighted in the model source. SLX contains a number of truly impressive innovations in simulation technology. It provides the sophisticated capabilities that will be required to meet rapidly-changing modeling needs in the second half of the 1990s.

Building discrete event simulators with Create!

Michael Rüger

Dept. of Computer Simulation and Graphics
Otto-von-Guericke-Universität Magdeburg
P.O.BOX 4120
D-39016 Magdeburg, Germany
michael@isg.cs.uni-magdeburg.de

Abstract

Create! started as a project for developing a discrete event simulator for material flow and logistics systems. Since then it has been enhanced to form a multi-platform, powerful integrated development environment (IDE) for discrete event simulation.
This paper will show which steps where necessary to arrive at where we are now and how several quite different simulation and evaluation tools where built using this environment.

Introduction

Through recent years new graphical simulation tools have appeared on the market and gained a remarkable success in the simulation community. Most of these environments come with a set of prepacked components aimed at a certain application field, some provide a more or less general extensibility.
Solutions to the problem of extensibility range from simple programming language interfaces providing access to external C(++), FORTRAN or Pascal code to combined library and simulation language approaches for extending the set of simulation elements. Finding a both flexible and intuitive way of specifying the behavior of elements or processes is another challenge in this context.
Allowing a user to create new or modify existing elements requires the existence of a language or process specification of some sort to describe the desired behavior. Compiling/linking on the one hand or interpreting this code on the other both have their pitfalls. While the first generally produces faster models it introduces a potential danger of corrupting the system through unwanted side effects or simply programming mistakes. Interpreting user code allows trapping errors at runtime but is somewhat slower.
Another important issue is the range of supported platforms. Using an application and it's data on different platforms implies support for different filename conventions, character encodings and user interface look-and-feels.
By choosing Smalltalk as the underlying development environment, most of the above problems could be solved quite easily. ObjectWorks™-Smalltalk is binary portable across all supported platforms, thus allowing us to use Create! on workstations as well as on PCs or Macintoshs™. The incremental compilation and dynamic binding provides a way to add code at runtime while providing the safety of an interpreted system.

In the following we will show how these capabilities are used to solve the problems mentioned above and together with the modelling and development framework form a powerful environment with low threshold and high ceiling.

The Create! cloud[1]

Based on this underlying framework is a two step approach to the development of simulators. The first step is the creation of

- a generic runtime environment for building models, performing simulation runs and evaluating the results and
- an integrated development environment containing all necessary tools to create new libraries of elements and objects.

Combining the libraries with the generic simulation environment in the second step results in a new simulator for the end user (*Figure 1*).

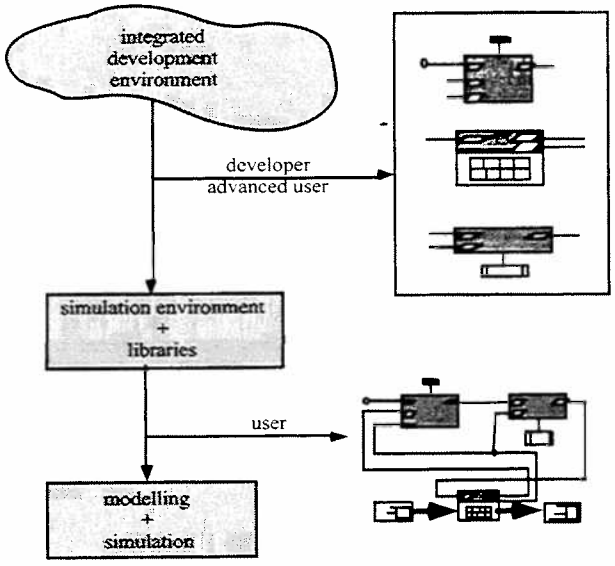


Figure 1: Create! IDE and runtime environment

1. The word *cloud* came up during the early Create! development when there was just this cloudy vision but no idea how to realize it.

Simulation kernel and modelling language

The Create! simulation kernel is based on extended finite automata. Each automaton is defined by a set of states, state transitions and functions attached to the transitions. Transition functions are defined (coded) using an object-oriented programming language which is syntactically close to C++ and conceptually close to Smalltalk-80. It supports inheritance for data structures and simulation elements as well as function polymorphism depending on function name and parameters.

The notion of finite automata is extended in so far as some transitions may be guarded, so, depending on the outcome of a boolean expression (condition), one of two alternate transitions will be executed. Each automaton is executed by a state machine which reacts to input signals and performs the appropriate transition depending on the current state.

This concept named SEC (State, Event, Condition) has later been extended to allow the modelling of concurrent processes based on the SEC automaton definition (Par-SEC: Parallel State, Event, Condition). Although the concept is named *parallel* SEC, processing of simulation events takes place in a strictly sequential manner. Nevertheless, as each state machine has its own execution space, one can think of these as executing in parallel, without worrying about mutual exclusion on variable access etc.

An example for utilizing this concept is a transport system with several vehicles, all sharing the same behaviour. The behaviour of a vehicle would then be described by the automaton definition. At runtime a state machine will be forked for each vehicle in the simulation model.

Elements are completely encapsulated. Communication between elements is realized by sending messages along connected in- and outputs. There are two different ways of communication: message passing with and without handshaking. The first one is used for information exchange like sending of orders and receipts, the second one for realizing the material flow. Figure 2 shows the general layout for an element with it's in- and outputs.

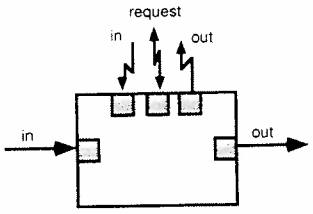


Figure 2: general element layout

Figure 3 illustrates the handshaked communication through a plug in combination with the automaton definition. Plugs define the communication protocol through a set of pins, transmitting a signal or message, and guards, promoting a boolean value.

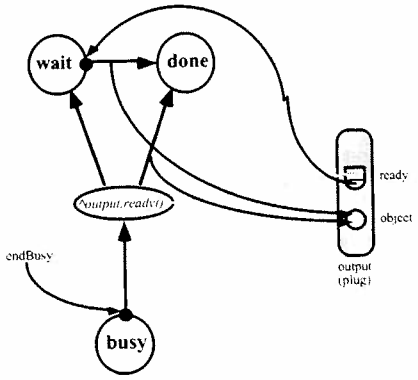


Figure 3: communication through plug

Element definition

Based on the concepts layed out in the previous chapter, the user can define Create!-elements using the element editor. The editor allows the interactive editing of constants (parameters), variables, in- and outputs as well as automata definitions and transition functions. Figure 4 shows the editor while working on the elements of the LogiChain!-environment.

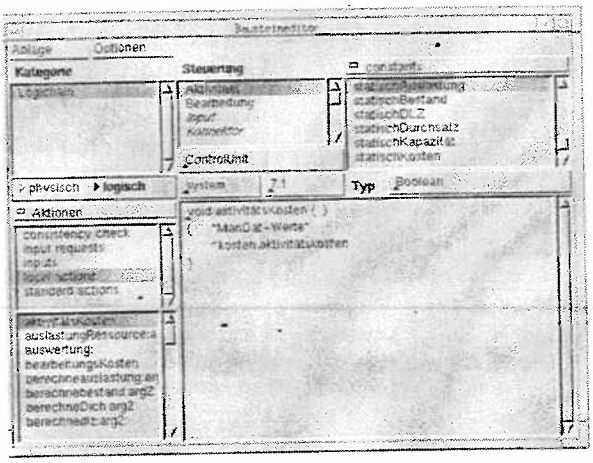


Figure 4: Element-Editor

In order to display the elements in the model editor, each element has one or more graphic symbol(s) (icon) associated with it. Icons are defined in a freely scalable vector graphic format based on the Office Document Architecture (ODA). By using the icon editor, the user may choose from one of the predefined icons from the libraries or define a new one.

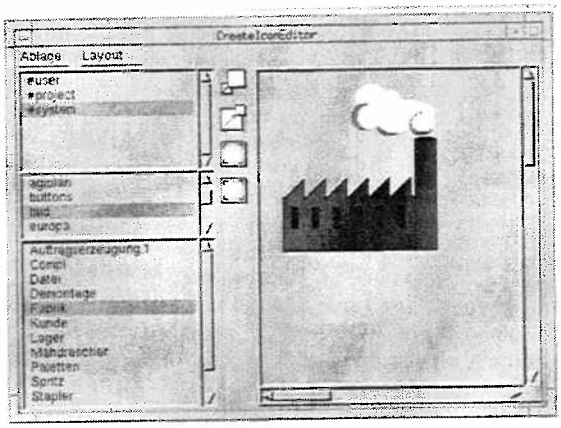


Figure 5: Icon Editor

Using the type information provided within the element editor, dialogs are generated to allow entry and modification of element parameters.

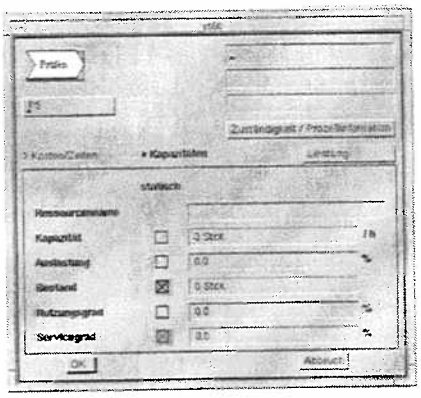


Figure 6: generated parameter dialog

Trace, evaluation and animation

Both evaluation and animation use a generic trace format. The trace is generated by generic calls coded into the elements during the simulation run. An early version supported element specific (load, state) trace events only. The current implementation is based on an ASN.1 compatible file format and supports both element and object specific (creation, attribute change, destruction) trace events.

Evaluation and animation depend to a large extent on the application domain. Nevertheless a set of basic evaluation and animation methods is desirable. Based on a generic framework, domain specific evaluations can be added when needed.

The solution choosen in Create! is to provide a method to generate a stream of events during the simulation and work on this stream online or offline. The object-oriented nature of the underlying Smalltalk system makes it easy to transparently handle both internal and external streams or pipes.

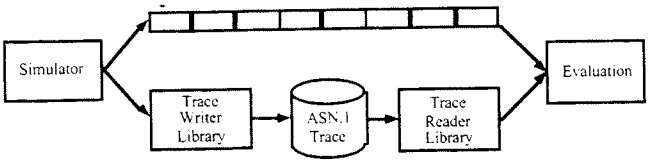


Figure 7: trace event stream

The tracer interface allows element related trace events like *state*, *load* or *object count*, as well as object related events like creation, modification of attributes or deletion.

All events are piped through an evaluation network which consists of single data processors realizing tasks as counting, averaging or other statistical filter functions. Connected to the outputs of this network are visualization components like graphs or animation views. As the evaluation sees events only, it does not depend on the kind of elements generating these events, thus allowing us to provide a generic set of evaluation and animation methods.

Figure 8 shows the results of an amount evaluation based on the increment and decrement trace events. In this particular case, the line chart displays the variation of amounts in a buffer. The same evaluation can be used for every element providing the appropriate trace events and thus allows for an easy integration of new elements into the existing framework.

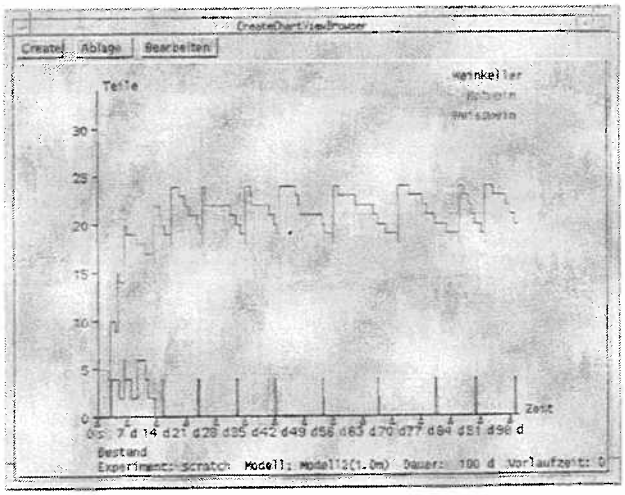


Figure 8: line charts of amounts in a buffer

Model and experiment management

When building models and performing experiments on them, the user in general has to use functions of the underlying operating system for organizing his data. Create! provides the concept of a simulation project to support the management of models, associated parameter sets and results of simulation runs(Figure 9). It relieves the user from coping with different file systems and name conventions on all supported platforms including DOS™ and UNIX™.

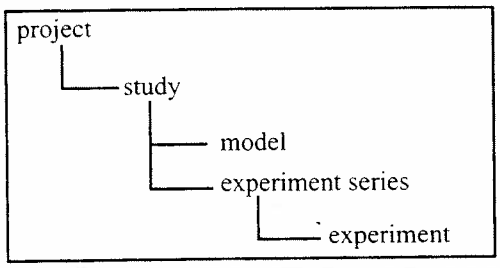


Figure 9: organization of a simulation project

A project is organized into studies which in turn contain models and experiment series. Models come in two flavors: working versions and frozen ones. Frozen models may not be modified and form the basis for experiments, which therefore can be replicated at any time.

Each of these project parts can be created, renamed, duplicated or deleted without the need to use the commands as provided by the underlying operating system.

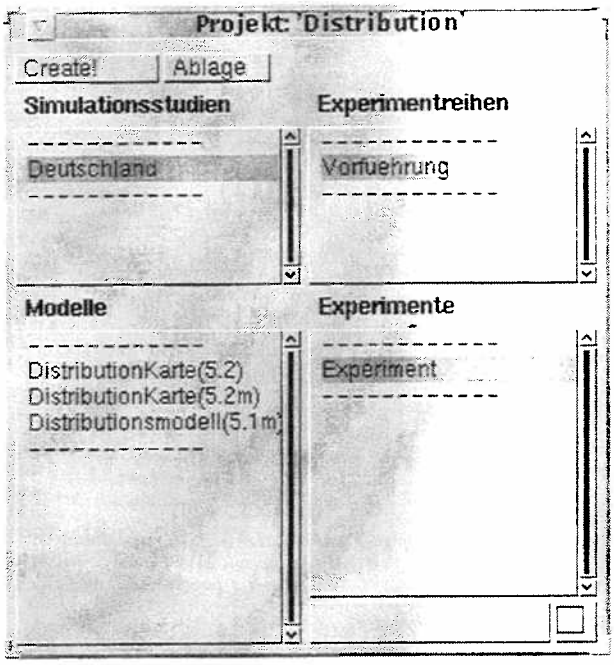


Figure 10: project management window

Projects provide yet another functionality. Each project works as a a container for the installation of element and type definitions, libraries and icons. In this way, parts of an environment which are specific extensions for a simulation project, are encapsulated within this project without cluttering the overall system space containing the default elements and libraries. When entering a project, these components are dynamically loaded into the current environment, and hidden again when leaving the project.

Simulation environments

Currently the following environments have been built using the Create! IDE:

Create!-LSG
simulation for strategic decision support

Create!-Batch
simulation for batch oriented production systems

Create!-Simple
simple single server environment for teaching purposes

Create!-Structure
simulation of logistic systems at the structural level

Create!-LogiChain
analysis of process chains (no simulation!)

All share the same runtime kernel with some minor simulator specific changes in the user interface. Figure 11 shows a snapshot of the model editor within the Create!-Structure environment. The same editor is used in the other tools as well.

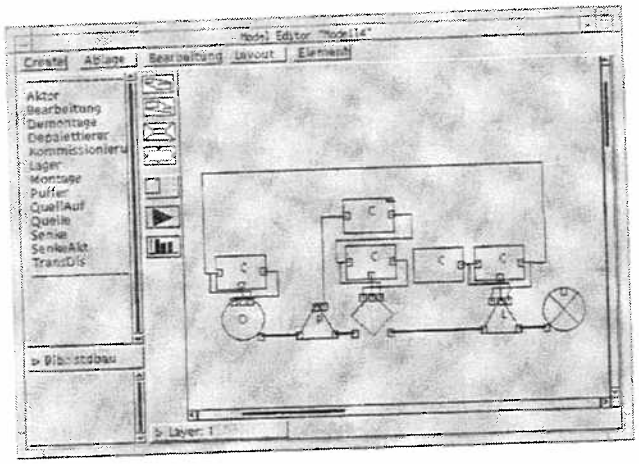


Figure 11: Create!Structure model editor

A somewhat interesting exception is the Create!-LogiChain environment which makes no use of the simulation kernel. It was built using the IDE's abilities to support graphical, element based modelling environments. Elements (activities) are placed on a regular grid. Controlled by parameters provided with the element definitions, elements of the process chain auto-connect to their next neighbors. After constructing the chain, a generic computation algorithm, which in turn calls functions defined within the elements, is run on the model and provides the results of the analysis.

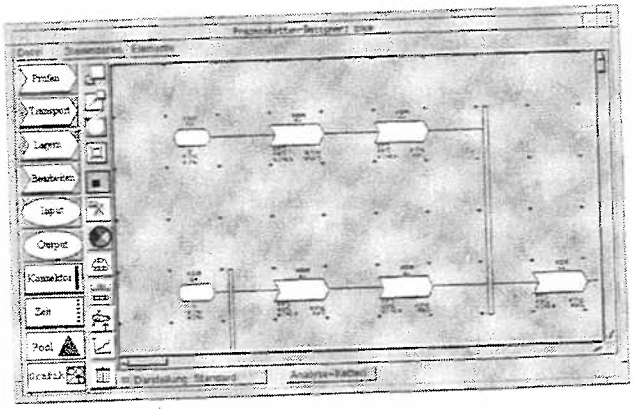


Figure 12: LogChain Designer

References

Carrie A., 1988. Simulation of Manufacturing Systems. Chichester: John Wiley & Sons Ltd.

Cota B.A., Fritz D.G. and R.G. Sargent, 1994. Control Flow Graphs as a Representation Language, in *Proceedings of the 1994 Winter Simulation Conference*, Lake Buena Vista, Fl, December 1994, pp. 555 - 559

Paul R.J., 1993. Activity Cicle Diagrams and the three Phase Method, in *Proceedings of the 1993 Winter Simulation Conference*, Los Angelos, Ca, December 1993, pp. 123-131

M. Rüger:, An object oriented divide-and-conquer approach for modelling control logic in logistic systems, In: SCS European Simulation Multiconference, Nürnberg (Deutschland), 1990

Rüger M., Hoppe U. und H. Kirchner, 1990. Objektorientierte Modellierung von Bausteinen innerhalb der Simulatorentwicklungsumgebung „Create!", in*Fortschritte in der Simualtionstechnik*, Band 1, Wien, 140-144

R. Schmidt, A. Schürholz, M. Rüger, CREATE! - Simulation aided development of control software, In: SCS summer simulation conference , Austin (Texas, USA), 1989

Schruben L.W. 1983. Simulation Modelling with Event Graphs, *Communication of the ACM* 26: 957-963.

Schruben L.W. 1990. Simulation Graphical Modelling and Analysis (SIGMA) Tutorial. in *Proceedings of the 1990 Winter Simulation Conference*, New Orleans, LA, December 1990, pp. 158-161

Som, T.K. and R.G. Sargent, 1989. „A Formal Development of Event Graph as an aid of Structured and Efficient Simulation Programs", in *ORSA Journal on Computing*, 1, 2, 107-125

SIMPRO 1984. *GBS - Die graphische Basissprache für das SIMPRO-System*, IMPRO Gmbh Berlin.

Zeigler, B.P. 1984. *Multifaceted Modelling and Discrete Event Simulation*, Academic Press.

World-Wide-Web

The Create! home page can be reached at
http://simsrv.cs.uni-magdeburg.de/~create

CAD-I++: Advanced Interface between CAD and Simulation System SIMPLE++

G. Kronreif
Unseld + Partner
Business and Marketing Consulting • Simulation!
Lerchenfelderstr. 44/9, A-1080 Vienna, Austria
E-mail: kronreif@unseld.co.at

1. Introduction

Attending to the increase in production automation and to the utilization of computer aided technologies in design, planning and manufacturing, simulation technique has been becoming an established tool for planning, realization and operating of technical systems. Initially a tool for providing cover against planning risks, simulation now is utilized in every particular stage of planning and realizing the system as well as for control of the manufacturing process. Because of the unquestionable advantages of this tool, computer aided simulation will gain more importance in future. Considering this, an improvement of the simulation systems in respect to flexibility, interfaces to particular (hardware-) components of the manufacturing systems, user-orientation for modelling and experimentation tasks as well as complete integration of simulation technique into the whole planning cycle will be necessary.

Above all, integration of the two essential CA-techniques for design and layout planning - Computer Aided design (CAD) and (Computer Aided) simulation - seems to be advisable. Generally the CAD drawing of the plant layout contains many data used by the simulation system, as there are information concerning the physical elements of the plant (e.g. machines), information to the flow of materials, transport routes, and others. Further data could be available in PPS (product planning systems) and/or in plant control systems. When opening the access to data stored in other CA-systems and using them for (partially) automated generation of a simulation model, the efforts and costs necessary for a simulation study can be decreased dramatically.

There are different ways to form a combination between CAD functionality and simulation features:

- ⇨ *Simulation within CAD-environment*
  The CAD-system serves as a carrier for an embedded sub-system with basic simulation features. This arrangement is used especially in simulation of robotics systems (e.g. CATIA and CATIA-Robotics).

- ⇨ *CAD functions within simulation environment*
  In general there are some CAD functions available in most simulation systems for design of the animation layout (i.e. animation background and facilities). Therefore making a simulation/animation model consists of two steps: drawing of the animation layout and definition of links between the simulation model and the particular elements of the animation layout.

- ⇨ *CAD-Simulation interface*
  The goal of such a CAD-Simulation interface is to use the full functionality of each tool during the appropriate planning stage. Previous approaches of a combination between CAD and simulation systems are limited in exporting the CAD-layout data for providing a more realistic graphical representation of the simulation model. Hence, the CAD layout is being reduced to a static animation background - the information included in the CAD layout, like transport routes, gets lost.

The CAD-simulation interface presented in this paper is more than an export function for CAD data. Beside drawing of the plant layout, the CAD software serves for definition of a hierarchical structure of

the plant components and for a static analysis of the plant layout. In addition to the data included in the CAD drawing file, information from other sources, like PPS systems or plant control systems, can be processed as well. After development of an appropriate design for the plant layout, a data exchange system creates the necessary structures for automatic generation of the simulation model. For a first prototype we used CAD system AutoCAD. Simulation is executed with SIMPLE++ - the object-oriented structure of simulation models in SIMPLE++ and the advanced features for automatic model generation provided with this system are forming the basis of an efficient data exchange. A second prototype using CATIA as CAD environment is in planning stage.

2. CAD Software AutoCAD

AutoCAD (by Autodesk Inc.) is the most common CAD software used in architecture, electronics, engineering. It is a powerful CAD tool with many applications available (e.g. AutoCAD Designer, AutoSurf, 3D-Studio...). AutoCAD supports various hardware platforms and operating systems (DOS, WINDOWS, WINDOWS NT, HP-UX, SUN, etc). Some important features of AutoCAD used for the presented interface system CAD-I++ are:

- ⇨ Powerful 2D and basic 3D drafting and designing tools.
- ⇨ Graphical user interface with pull down menus, dialogue boxes, accelerator keys, icon toolbar and floating toolbox.
- ⇨ AutoCAD menus can be customized to suit user needs.
- ⇨ AutoCAD Development System **ADS** supports a programming interface to C or C++ code for development of custom AutoCAD applications. Furthermore there are additional interfaces to other graphic standards and database systems (AutoCAD SQL Extension **ASE**).
- ⇨ Performance: fast redraw, pan and zoom speed - important for large drawings.

3. Simulation Package SIMPLE++

SIMPLE++ is the abbreviation for SIMulation in Production, Logistics and Engineering design and its implementation in C++. The simulation package SIMPLE++ is almost becoming a standard software for object-oriented, graphical and integrated modelling, simulation and animation. All the features - graphical user-interface, system architecture and implementation - are corresponding to the demands of object orientation.

A major strength of SIMPLE++ is the significant increase in productivity when building, changing and maintaining models. The most powerful features of description-block, list and language concepts are provided in a single, integrated simulation environment. Features of object-orientation like inheritance, hierarchy, reusable objects or even models as part of models make this simulation software to one of the most successful simulation systems nowadays.

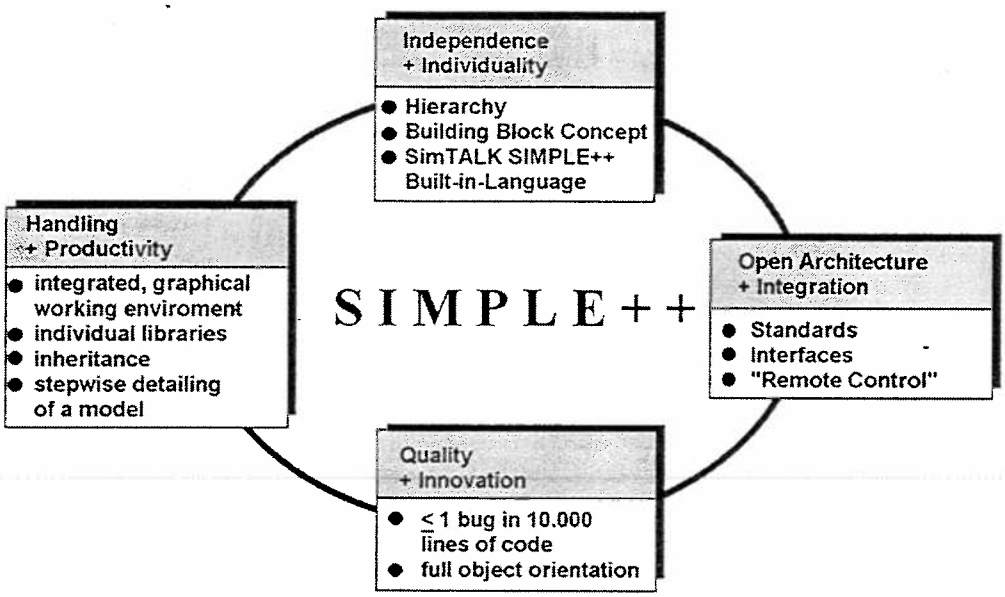


Fig. 1: Requirements in functionality for SIMPLE++

Graphical and intuitive model handling to increase productivity is a key requirement for the widespread adoption of simulation throughout the economy. In SIMPLE++ this is achieved with a unique integrated and incremental working environment, individual libraries (application templates), inheritance and a graphical object interface.

„Users should be independent" - consequently they are free to make an infinite range of tailored application objects which are serving themselves as templates for efficiently creating models. Individuality ensures that the simulation model has a lifelike animation and functionality.

4. Layout generation and simulation with CAD-I++

The presented system CAD-I++ is both, an extension of the features of the used CAD-system as well as an interface between CAD and simulation. To fulfil these tasks, CAD-I++ defines an additional link to a third 'component' (beside CAD- and simulation system): to a database system for storing and processing **non-graphic information (NIGI)**, like instruction sheets, list of orders, characteristic data of the particular facilities, and others.

4.1 Tasks performed within CAD environment

The main task of the CAD system is the generation of the plant layout. With CAD-I++ the user is endowed with all necessary functions for efficient layouting. After definition of the planning area and all existing cut-off regions, the next step is to define and insert the facilities of the plant (i.e. (tooling) machines, assembly systems, storage, etc.). The system CAD-I++ follows to a strictly separation between **graphic data** (dimensions of the facility, positions of the input/output unit) and **non-graphic data** (characteristic data of the facility, like mean time between failure MTBF, mean time to repair MTTR, ...). Graphic data are part of the layout drawing and have therefore to be defined during the layout generation (after input of the facility dimensions, a scaled symbol of the facility is inserted into the planning area). On the other hand, the non-graphic data are stored in the database system and can be displayed in special data sheets on request.

After (or during) the arrangement of the facilities, the planner can define a **hierarchical structure** in order to gain more clarity for the later simulation process. To set up a 'multi-layer' structure of the plant, CAD-I++ defines three groups: *Facility* (i.e. the plant 'hardware'; machines, storage, ...) - *Work Centre* (i.e. group of facilities with equal characteristics) - *Module* (i.e. group of Work Centres; e.g.

cost unit). The described model hierarchy is fully supported in later simulation because of the object-oriented structure of the simulation system SIMPLE++.

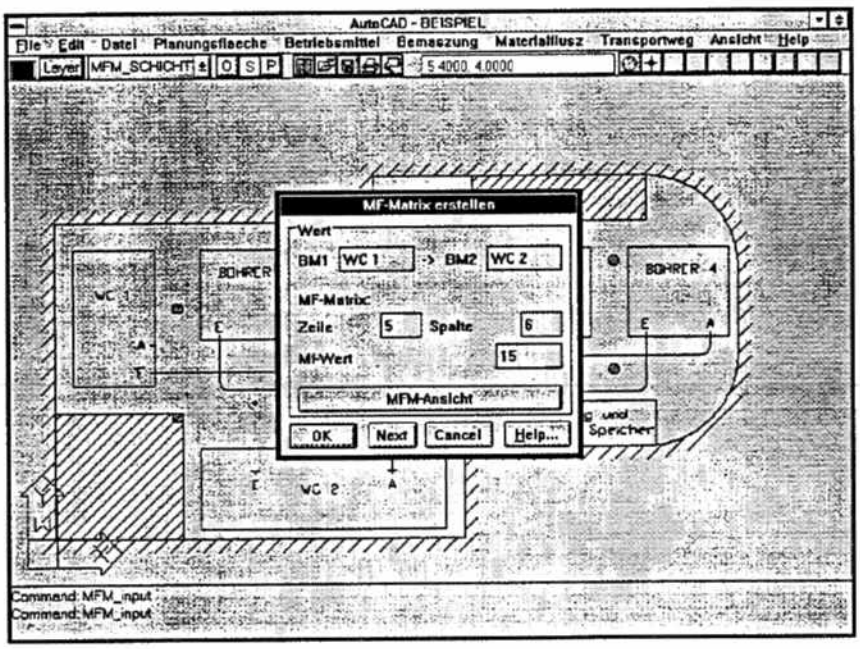


Fig. 2: Interface CAD-I++: Definition of the MFM

Prior to the dynamic analysis in SIMPLE++, the interface system CAD-I++ offers certain functions for static analysis of the flow of material (MF). To supply the system with the necessary data for this task, the matrix of the flow of material (MFM) either can be defined by means of data masks (see Fig. 2) or calculated using the non-graphic information. Combined with functions for graphical input of the transport paths by means of 'Point-and-Click'-Method, the valuation of the generated layout can be carried out by means of a detailed graphical display of the MF relations (Sankey diagrams).

4.2 Tasks performed within simulation environment

After static analysis of the layout, the appropriate CAD data can be exported to the simulation system SIMPLE++ via an interface file. A template model in SIMPLE++, including all the necessary functions for automatic model generation, transforms this data structure into a complete simulation model structured into the same hierarchy defined during layout generation. Beside processing the geometrical data stored in the interface file, the simulation model has access to the non-graphic information as well. No additional effort for modelling and definition of the input data for the particular simulation runs is necessary.

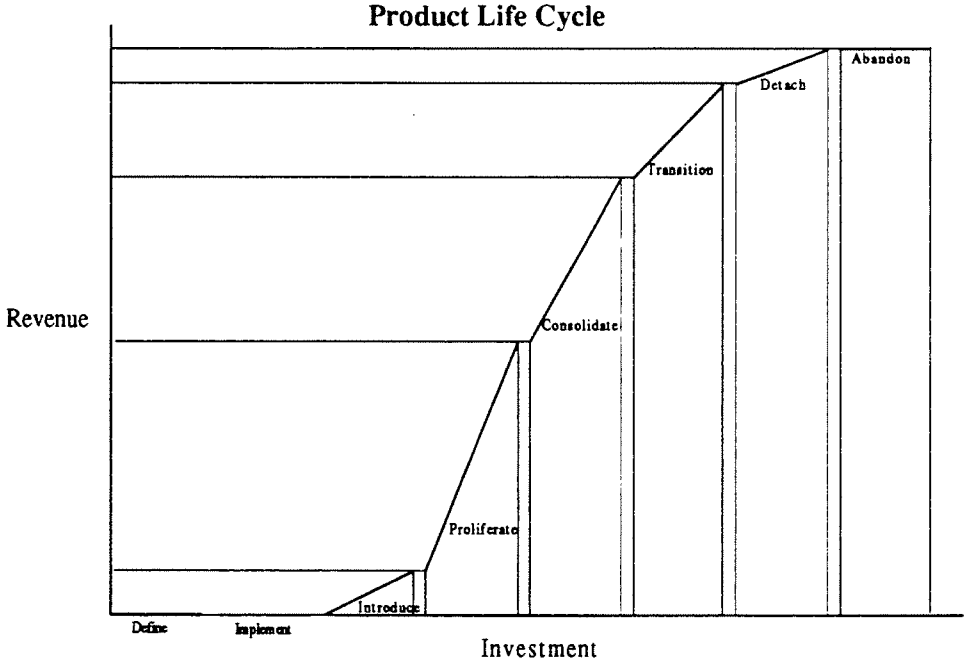
5. Summary

Using simulation technique, manufacturing systems can be designed, optimized and estimated. The plant layout, nowadays often developed by means of CAD-systems, stores the results of the layout planning in a concentrated way. The aim of the presented interface system CAD-I++ is to close the gap between these two planning tools in order to use the capacity of rationalization existing in manufacturing planning. With this interface system, all the planning data can be integrated and used by both systems. The arrangement of the facilities as well as the definition of the transport routes can be done within familiar CAD environment. After static analysis of the plant layout, CAD-I++ offers an automatic generation of an object-oriented, hierarchical structured simulation model for simulation system SIMPLE++ for further dynamic analysis of layout manufacturing process. Additional to the improved efficiency of planning, the integration of CAD information with CAD-I++ should gain more acceptance for the tool 'simulation'.

Insights into Successful Simulation Software Products

Joseph Annino, President
CACI Products Company
3333 N. Torrey Pines Court
La Jolla, CA 92037 USA
Tel: (619) 457-9681
Fax: (619) 457-1184

Organizations need new products just to survive. The need for new products is confirmed by the fact that the profits from today's technology leaders come primarily from products they introduced in the past five years. While the need for products is great, so also is the risk of product introduction -- most products never make it to market and those that do often make little or no profit. With failure so common and its cost so great, a strategy is needed for the successful introduction of products. It is helpful to consider such a strategy in the context of the product life cycle.

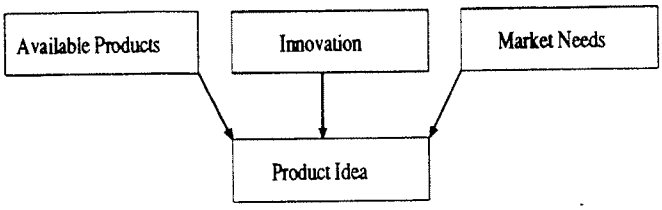


Define

It all begins when some visionary, drawing on talent, ingenuity and a predisposition to new ideas, conceives a product that fits the organization's strategic direction.

The creative idea encompasses an understanding of evolving technology, and a strong sense of what the market needs. Such good ideas are not uncommon, but thinking the idea through to an implementable product plan requires a combination of intelligence and commitment that is rare. In most organizations persistence is also essential. The intuitive visionary is often discouraged by premature requests for financial justification, such as Return on Investment, when little is known about either the needed investment or its potential return.

Product Definition



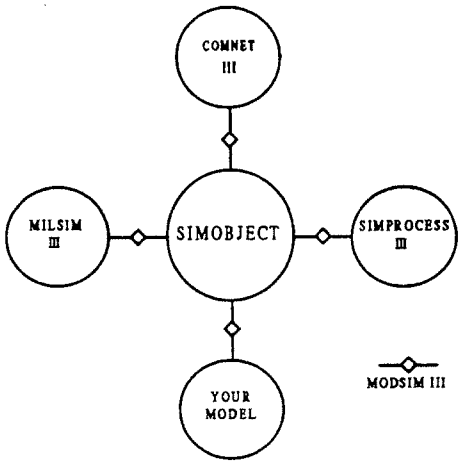
If the visionary has a history of success, the endless hand-wringing over whether to proceed with development can probably be dispensed with. The product idea must, of course, be tested. One excellent market test is to see if outsiders will put up money or other resources. You should contact organizations you regard as prime prospects, and make them a special "for-sponsors-only" investment offer. They get the opportunity to influence product functionality, and are its first users. Offer to repay their investment as a percentage of sales. If necessary, pay more than their investment.

Implement

Eighty percent of all products that are started fail at this stage. A company must have a great idea, a superior product development technology, a talented development team and strong management support, or it will be starting a game that it cannot win.

Time-to-market is an increasingly important competitive factor in high-technology products where the underlying technology advances rapidly, and the markets change quickly. One strategy is to develop a Platform that greatly facilitates the development of a family of products. By providing a jump start on product development, a Platform can sharply reduce the time, cost and risk of bringing a product to market.

Product Platform



A product Platform is collection of the common elements, especially the underlying core technology, that can be used across a range of products. In the above diagram, the SIMOBJECT Platform incorporating simulation and graphics, is specialized using the MODSIM III[1] language to develop members of the product family: COMNET III[2], SIMPROCESS III[3], and MILSIM III.

Differentiation, a way of distinguishing a product's value from competing products, is achieved both at the Platform level, and at the individual product level. The Platform provides advanced concepts and a consistent look and feel throughout the product line built upon it. When the Platform is enhanced, the entire family of products built on it is enhanced.

Of course what makes great products is not technology alone -- technology is perishable -- great products are built by talented hard-working people who are committed to the commercial success of the product. The most successful product developments are those by small, intensely focused teams, working closely and under a tight schedule. The mutual dependence of the team generates the high level of commitment needed to take product ideas to market.

The difficult creative work and long hours that go into successful product development must be rewarded with appropriate incentives. Technical people should be partially rewarded on the product's sales success -- just as salespeople are. This keeps them focused on what's important.

While product development is itself a consuming activity, marketing must begin **before** the implementation is complete. One technique is to organize a Charter Group of users by offering special benefits. Offer a special price and a money-back guarantee to those who commit to buy early. Work with these early adopters to perfect your interface, functionality, installation and support procedures, and sales literature. True, your charter users may point out a lot of things you don't want to hear. You may find that your sales literature is misdirected, or that your pricing is wrong, or your product has no appeal or that its appeal is different from what you thought.

It may be necessary to cut your losses.

Introduce

The best way to introduce a product is to try to sell it! Market research has been mostly poor at predicting the success of a truly new product. People don't know what they want until it exists. Once you have something to show -- even if it is only an initial version -- prospects and customers will let you know what they like and dislike about it. You can then make needed changes.

Even the greatest product cannot succeed if nobody knows about it. You must aggressively promote the product with advertising, direct mail, conferences and trade shows.[4]

Proliferate

Continually infuse the product with significant enhancements and promote relentlessly.[5] Enhancement ideas come from the imagination of developers, often stimulated by the needs of customers and prospects. From the long list of potential enhancements, choose those that expand the market.

As the product achieves success it will attract competitors who know of that success, can examine your product, and will seize the opportunity for low risk profits by entering the market. With an aggressive enhancement program, you can discourage these competitors by forcing them to play catch-up. You want them to look elsewhere for opportunities.

No matter what you do, however, at some point product proliferation will slow and you need to be positioned with a successor product. The pace of change is so great that a large installed base

offers little protection -- look at what happened to Lotus 1-2-3, dBase, Word Perfect, Turbo Pascal, ...

For this reason, technology companies must be willing to initiate the development of advanced products, that compete with their own successful but aging products, or their competitors will surely do it for them. In 1988 Microsoft wisely hired David Cutler, the visionary behind Digital's wildly successful VAX VMS operating system, to lead the development of its NT operating systems -- a superior competitor to their own already successful MS Windows.

Consolidate

Most high-technology products are eventually replaced completely by new technology. Because of the entry of effective competitors, loss of sales momentum, or market changes, revenues will slow down. Ironically, a large customer base also constrains product enhancement by demanding upward compatibility. This constraint creates an opportunity for competitors with a vastly superior product.

In addition, technological limits make successful products vulnerable[6]. For example, Intel based PC's appear to have reached a technological limit. **It costs a great deal to get a small performance improvement.** Intel's CISC (complex instruction set computer) is being threatened by IBM's new Power PC RISC (reduced instruction set computer). Just as the IBM Selectric typewriter, with its wonderful correction tape and type-balls, was threatened and then obliterated by word processors.

Transition

At this stage it is apparent that revenue will continue to decline. Cost cutting should continue in order to maintain profits. Be prepared to offer your customers a transition path to your highly-desirable successor product.

Detach

The decline of revenue continues to the point where the product is no longer profitable, but cannot simply be abandoned because there are customers using it. By raising the support fees and aggressively cutting costs the product may be made marginally profitable. Customers must be moved to the successor product.

Abandon

Customers who did not move to the successor product are notified that they can use the product "as is" but that it will no longer be supported.

---

[1] *A Quick Look at MODSIM III*, CACI Products Company

[2] *A Quick Look at COMNET III*, CACI Products Company

[3] *A Quick Look at SIMPROCESS III*, CACI Products Company

[4] Annino, Joseph, *How to Promote Your Own Products*, La Jolla, CA: Genesee Publishing, 1989

[5] Ibid., Annino

[6] Foster, Richard, *Innovation*, New York: Summit Books, 1986.

Metamodelling of Simulation Tools

M. Frank, Technische Universität Dresden, Fakultät Informatik

1. Introduction

Usually, three classes of discrete simulation software are distinguished

- simulation languages SL (SIMSCRIPT, SIMULA etc.)
- general purpose simulation systems GPS (SLAM, GPSS etc)
- special purpose simulation systems SPS (DOSIMIS, SIMPLE++, WITNESS etc.)

This classification is mainly based on the different levels of predefined means for model description. For instance, SPS offer predefined modules describing subprocesses of a special application area whereas GPS offer blocks describing events. The greater complexity of SPS-modules leades to decreasing expenses for aquisition but to a certain loss of flexibility for application, too. For this reason SPS designers frequently don't realize the modularity consequently. Generally, the modelling means offered force the user of SPS to describe the process to be simulated not only by (more or less complex) modules of typical subprocesses but also by additional statements of a programming language.

This hybrid way of model desciption confines the loss of flexibility but leaves the expense for aquisition and efficient handling of SPS on such a level that specialised users are required. For a more simplified utilization one should look for another way to ensure flexibility without renouncement of consequent modularity.

Obviously, there are two ways:

- the creation of SPS based on GPS on the one hand /1/
- the extension of SPS by means for descriptive module insertion.

2. Meta modelling

Some years ago we started investigations in this direction. They have shown that , to a certain extent, flexibility of a pure modular SPS can be guaranteed by metamodelling. Metamodelling means the description of all elements offered for model building on an abstract level. The subject of investigations mentioned is the simulation system TOMAS, the modelling conception of which is consequently modular. Referring to TOMAS, metamodelling is particularly the formal characterization of the modelling modules and functions.

The TOMAS' modelling modules are called operators and can be combined to an operator net flowed through by operands.

The entities which are called operands represent (passive) temporary elements (material parts, messages, jobs etc.) which after its generation move through the operator net in

order to be processed, transported, stored or distributed etc. by operator objects. An operator object is derived from a predefined operator type by specifying its parameters. The specification is supported by predefined functions.

Model building requires the following user activities, only:

- selection and denomination of suitable operators
- positioning of operator symbols on the screen
- drawing the needed connections dependent on the possible flow of operands
- specification of all denominated operators via its parameters.

The TOMAS metamodel describes formal features

- of models that can be build (global parameters).
- of the operators offered
- of the predefined functions

Global model parameters to be described are

- the maximal number of operator objects which are allowed for a model
- the maximal number of preceding and succeeding objects of an object
- the number of operator types offered

The operator description refers to its coupling conditions and its parameters including for instance:

- the minimal and maximal number of input respectively output valencies
- the number of freely usable valencies, (freely usable means that the connection with different operators is allowed; the complemental subset consists of valencies for special operators, only)
- the subset of allowed adjacencies (possible neighbour operators)
- the total number and the types of parameters (there are four types: alternative, scalar, function and list parameters
  - ∇ each parameter type has to be described by some further formal features respecting parameter dependences etc.

Metamodelling of functions includes for instance

- the function type and its general denomination
- the number of the assigned syntax graph
- the number of function parameters

  Per each function parameter are to be assigned
  - ♦ the type (real, integer, ...)
  - ♦ the assigned scope of values (to be selected from a predefined set)
  - ♦ a possibly given speciality (as for instance 3-σ-Intervall)

Dependent on the function type further features are to be fixed.
Summarizing one can say: the meta model defines the model room. That is the set of all models that can be build based on the defined metamodel.

3. System and surface generation

Based on this conception of metamodelling we designed an additional component for system generation and partial adaptation. Using this generation component the TOMAS administrator can derive a desired system version by

- changing the global parameters
- deletion respectively addition of operator parameters
- deletion respectively addition of operators or functions

The component is able to adapt the surface for the model building automatically except the type of parameters or functions added don't yet exist. The basis for this adaptation are predefined widgets of menu and formula windows (OSF/Motif).
All data describing the meta-model are stored in a comprehensive file called GENA.DAT. Under the precondition that the parameters (of operator or functions) added don't extent the already defined set of parameter types, the generation component is able to adapt the model building component automatically. The basis for the automatic adaption are predefined procedures for the generation of windows and window hierarchies by using predefined widgets (OSF/Motif).
Morever the generation component is used while the process of building a special model is running in such a way that all windows needed for this model in order to input the model data are dynamically generated. For this reason there is no additional expense for adaption the model building surface to changes of the modelling means as it would needed in the case of a static surface definition. On the other hand, only such windows are produced which really are needed for the special model. Only if a not existing parameter type is demanded the administrator has to extend the generation component.
Recently, such an extension has been realized in order to equip TOMAS with an interface to the post-run-animation-system PROOF. PROOF needs an ACII-Trancefile of all events to be animated. For this reason the generation component has been extended by

- a global alternative parameter allowing to decide whether post-run-animation is desired or not (This parameter is important for the dynamic generation of the model input surface)
- additional alternative parameters of the differant operator types describing all events which can reasonably be protocolled for animation
- a specification file describing the protocoll text of the animateable events in accordance to the PROOF owned syntax
- routines for the generation of windoes respectively window hierarchies allowing the choose of events which are really desired to be protocolled .

Of course, the simulator has to be exented, too, especially by routines for the tracefile generation based an the specification file, mentioned. Based on the concept of meta modelling under discussion the expense for the extension mentioned was comparitive small.

4. Concluding remarks

Up to now. the means for system extension has been focused to the surface for model development and manipulation, only. That's why our experiences have shown that functional extensions cause much higher expenses for adapting the model data files and the surface for model building than for changing the modular simulator program.
The idea of meta-modelling has been elaborated in detail by Stephan Strohmeyer /2/. Substantial contributios to its actualimplementation have been realized by Simone Bersiner /3/, Kathrin Sippach /4/, Alexander Hartmann /5/.

References:

/1/ Apsel, T.: Attributierbare Modellkomponenten zur Unterstützung von simulationsunerfahrenen Anwendern mittels eines universellen Simulationssystems.
in: Biethahn (Hrsg.): Simulation als betriebliche Entscheidungshilfe: Neue Werkzeuge und Anwendungen in der Praxis; Braunlage 1995, S. 63-74

/2/ Strohmeyer,S.: Theoretisch-methodologische Probleme der Entwicklung und Nutzung diskreter Simulationsmodelle und daraus resultierende Konsequenzen für die Gestaltung des Zugangs zu fachgebietsorientierten Simulationssystemen.
Dissertation A, Technische Universität Dresden, Informatik-Zentrum 1987

/3/ Bersiner, S.: Neukonzipierung der Generierungskomponente des Modellierungs- und Simulationssystems TOMAS/32.
Diplomarbeit, Technische Universität Dresden, Fakultät Informatik 1994

/4/ Sippach, K.: Untersuchungen zur Anpassung der Oberfläche der TOMAS-Modelleingabe an veränderte bzw. neue Modellbausteine.
Diplomarbeit, Technische Universität Dresden, Fakultät Informatik 1995

/5/ Hartmann, A.: Entwurf und Implementierung einer Schnittstelle TOMAS-PROOF zur Post-Run-Animation eines TOMAS-Modells.
Diplomarbeit, Technische Universität Dresden, Fakultät Informatik 1995

Modular Application Objects: Closing the Gap between Flexibility and Ease of Modelling

Dietmar Geuder
AESOP GmbH, Königsstr. 82
70173 Stuttgart, Germany
E-mail: geuder@aesop.de

ABSTRACT

Compared to pure simulation languages modern graphically oriented simulation environments offer a comfortable efficient user interface. This increase in modelling efficiency, however, came at the expense of modelling flexibility and scope of application. In this paper, we will propose an approach which offers the ease of modelling of building block oriented simulators while keeping the flexibility of simulation languages.

1 MOTIVATION

During the last 40 years simulation systems have come a long way. Starting with general purpose programming languages, explicit simulation languages optimised for modelling and simulation evolved. Nowadays graphical simulation systems and modelling environments are widely used (Schmid, 1994).

With pure simulation languages, modelling requires a comparably high level of programming skills and is therefore mostly performed by specialists. Implementing the model on this level provides for the substantial flexibility to model the real world to as much detail as necessary. Also, a vast variety of application areas can be covered with the same language.

Modern graphical simulation tools being targeted for specific application areas usually offer a standard set of building blocks or templates. With these a model can easily be built graphically on the computer screen with a few mouse clicks. The model is then adjusted to the real system by setting default parameters and selecting standard operation rules. To allow this, these tools are mostly domain restricted and are shipped with one or several sets of templates for specific application areas (Hillen, 1993; Banks, 1994). Obviously this approach is by far more powerful as far as speed and ease of modelling is concerned. Due to the graphical user interface, models can be built by practitioners with only limited simulation experience and no programming skills.

This increase in efficiency, however, leads to a significant loss in flexibility. The fixed set of application related templates restricts the variety of systems and the detail which can be modelled. With existing systems one can thereby either find high flexibility and a wide scope of application or a comfortable, easy to use modelling environment. An optimal solution would incorporate the advantages of both type of systems:

- an integrated graphical user interface
- application specific building blocks or templates

for the *ease of modelling* and

- application objects configurable and changeable by the user
- ability to create new application objects in an easy way
- availability of an integrated programming language for specific control strategies

for *flexibility and application scope*.

While conventional systems compromise in either efficiency in modelling or in application scope, an ideal toolset will provide the flexibility to cover different application areas as detailed as necessary by fully adaptable and reconfigurable application objects (see Figure 1).

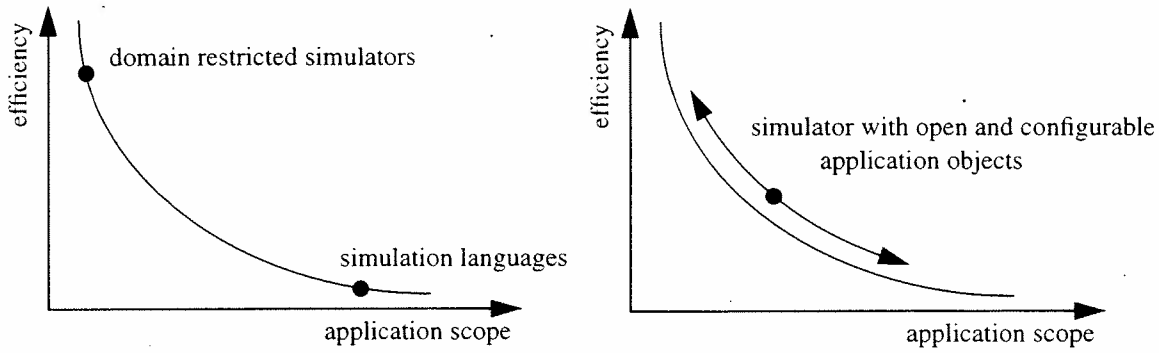


Figure 1: Configurable application objects open to the user combine efficient modelling with a wide application scope.

2 FLEXIBILITY BY MODULAR APPLICATION OBJECTS

SIMPLE++ an object oriented modelling and simulation environment offers hierarchical modelling, generic objects, an integrated programming language and a graphical user interface. This serves as a basis for implementing an approach which incorporates both types of features described above.

2.1 A Layered Approach of Open Objects

Instead of fixed high level building blocks we propose application oriented objects which are composed of generic objects on a lower level. Their functionality is determined by the default behaviour of the basic objects and their parameter settings. Application specific functionality can be added by high level program statements while domain specific data is represented by user defined attributes and variables.

By this process building blocks are created which are specific for a given application area. With definable control and data structures they can be as detailed as necessary. Generic basic objects together with configurable attributes and an integrated programming language provide for the required flexibility to cover a wide variety of application areas. With a hierarchical modelling environment the functionality can even be refined in an entire hierarchy of levels. Figure 2 shows a library containing basic objects and application objects. The internal structure and the dialogue mask of a buffered processor is depicted as an example.

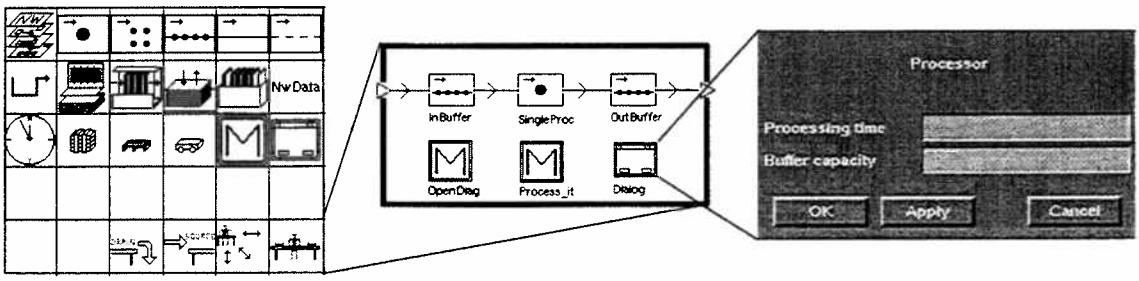


Figure 2: Example for a processor with buffers as an application object being composed of basic objects. With its dialogue window it appears as an integral high level application object.

These application objects are an integral part of the modelling environment and are represented as classes of objects in the library. Thanks to configurable dialogue boxes, these modular objects look like standard fixed templates to the user and can be employed accordingly.

That is, they can graphically be inserted in a model, and the parameters and control rules be set as necessary. This guarantees the user friendliness and ease of modelling modern domain restricted simulators offer.

2.2 Flexible Modification of Application Objects

While for modelling the application objects resemble standard parametrisable objects, they are nevertheless open for the user for modification. Should at any time additional parameters or operating rules be required or should the overall structure of the building block need to be changed, this can easily be done to any detail as the internals of each application object are fully accessible. The same graphical integrated modelling capabilities apply for all hierarchical layers. Changes like adding, deleting or redefining the structure or functionality are easily possible and there is no need to fall back to a different description mode.

Starting with a given set of application objects users can thereby actually create their own objects targeted to their individual modelling purposes by modifying the internal structure, the number and type of parameters and control strategies. In this way one is not restricted to a given set of parameters or attributes being predefined for domain restricted building blocks. Also, additional hierarchy levels may be introduced with this concept which allows incremental refinement of existing models.

In an object oriented modelling environment as offered by SIMPLE++ these changes will automatically be propagated to the instances of the objects in the model by inheritance. That is, added or modified functionality of a class in the library will automatically become active for all instances of that class in a model. This provides for very effective and productive modifications as necessary for common ‚what if' scenarios. Also, a full set of variants can be created for these application objects by deriving child objects and making the necessary modifications.

2.3 User Interface for Different Skill Levels

The hierarchy and the configurable dialogue boxes as described above make the user interface adaptable to different skill levels. Depending on the expertise of the user, he can simply use the application objects as they are and adjust them by setting parameters and selecting operation rules. He will thereby stay on a comparably high level requiring only limited practise. More experienced users can also change the internal structure of the application objects by opening the building block and graphically changing it in the same way the global model is changed. If necessary, the entire functionality of the building block can be adapted by progressing to the next lower level and programming new control structures with the built-in language.

Different views of the simulation environment and degrees of changeability can thus be distinguished as depicted in Figure 3. Information hiding of data irrelevant to certain user groups is directly supported.

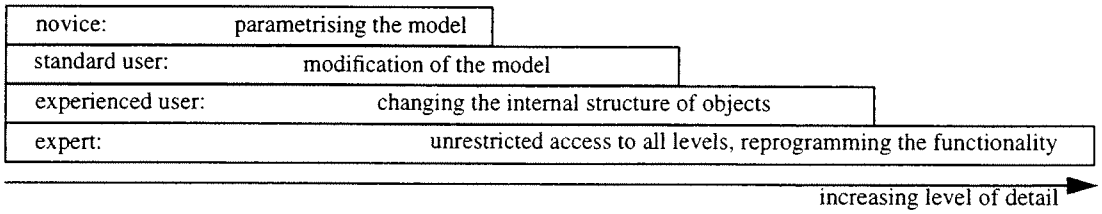


Figure 3: Different views and access levels of a model are supported by hierarchy and user defined dialogues. Information hiding and access restrictions can thus be implemented.

3 APPLICATION EXAMPLES

Various sets of modular application objects were created using the above approach. These include libraries for manufacturing, warehousing, business process reengineering, and staff and shift patterns. In Figure 2 two sets of application objects for chemistry and transportation are shown as an example.

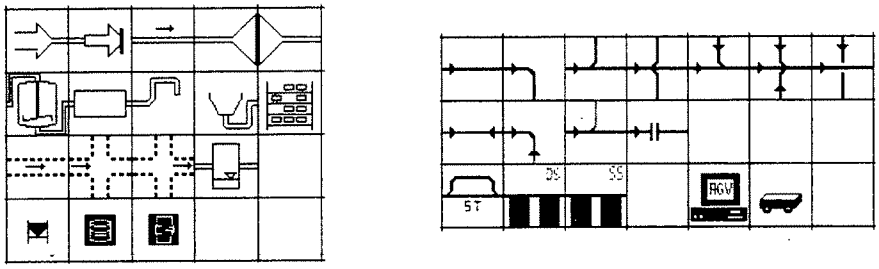


Figure 4: Application objects for modelling of chemical processes (left) and for an AGV system (right).

The application objects in the above figure were created with the SIMPLE++ modelling environment. Based on its graphical user interface for modelling it offers the possibility to create new application objects (AESOP, 1995). For this purpose generic basic objects are available which fall into different categories such as processing elements, entities, control structures, data structures, displays and dialogues, and interfaces. An integrated high level programming language with a variety of data types guarantees the implementation of specific functionality.

The application objects shown above are based on the same generic objects. These are aggregated in 1 up to 3 hierarchical layers depending on the complexity of the object in question. The functionality of these objects was specified in collaboration with engineers from the respective field and the graphical representation (icons) was chosen accordingly. By a double click on any of these objects, a dialogue box will open which displays only the parameters and menus which are relevant for the planner or engineer using the model.

Sets of application objects for different fields are shipped with SIMPLE++ and were used in various projects. Being modular and open for the users, they were also modified and adjusted to the specific requirements of simulation studies. In addition they serve as templates for variants and for new objects to be created.

4 RESUME

Based on a hierarchical approach and generic objects a variety of high level application objects can be created for different domains. With definable dialogue masks these objects are as easily usable and configurable as the fixed templates modern domain restricted simulation systems offer. However, being modular and open for the user, they offer the flexibility to change any part of them as far as necessary to customise them for the specific systems to be modelled. The modular application objects thereby combine the ease of use of domain restricted simulators with the flexibility of programming languages.

REFERENCES

AESOP GmbH (1995) *SIMPLE++ Reference Manual Version 3.0.* Stuttgart, Germany

Banks, J. (1994) *Software for Simulation.* Proc. 1994 Winter Sim. Conf., Orlando, FL

Hillen, D.W., Werner, D. (1993) *Taylor II Manufacturing Simulation Software.* Proc. 1993 Winter Sim. Conf., Los Angeles, CA

Schmid, B (1994) *Simulationssysteme der 5. Generation.* In: Fortschritte in der Simulationstechnik vol. 6, Berlin 1994

Micro Saint Simulation as a Means of Evaluating Optimal Conveyor Management Strategies

by K. Ronald Laughery, Ph.D.
Micro Analysis and Design, Inc.
Boulder, Colorado USA
Work was performed while Dr. Laughery was
at the University of Nottingham, Nottiingham, United Kingdom

In recent years there has been a great increase in the use of simulation as a means of evaluating manufacturing investment alternatives. Dunlop Cox Ltd., designs, develops and manufactures vehicle seats and seat mechanisms in Nottingham, England. The factory is structured on flow line, cellular manufacturing basis and it operates using Just In Time principles. Simulation was chosen as one of the tools to improve the performance of the factory. Of particular interest was the conveyor and paint plant that was used in the manufacturing process. This was seen as the limiting factor on total plant productive capacity. Several conveyors are used to transport seat components on carriers (flight bars) from the production cells to the paint plant, which is a bottleneck of the factory, and back to the cells again. The aim of the project was build a model which could then be used as a tool to discover ways that the conveyor hardware, software, and or usage practices could be changed to improve the efficiency of the plant.

Of particular interest in the study reported here is the use of dynamically programmable flight bars. The current system used a movable bolt on each flight bar to indicate to which particular production cell it was assigned. An alternative was to make an investment in a hardware and software system that would allow the allocation of flight bars to production cells to be dynamic as the requirements of the cells vary on a moment-to-moment basis. However, this would require a substantial investment. The question addressed in this study was how great an increase could be obtained by this potential investment.

The Conveyor System

The ten production cells at Dunlop Cox Ltd. share common paint plant facilities. Several conveyors transport the seat components from the cells into the paint plant, through the painting process and back to the cells again for assembly. The overall layout of the conveyors is illustrated in the Figure 1. At each cell, components are manually loaded onto load jigs hung form carriers (flight bars), which stop on a siding conveyor. From here they are released manually onto the main factory conveyors when the operator pushes the release button. A flight bar takes the components throughout the entire system and returns the painted components to the siding of their home cell, where they are unloaded and new parts take their place.

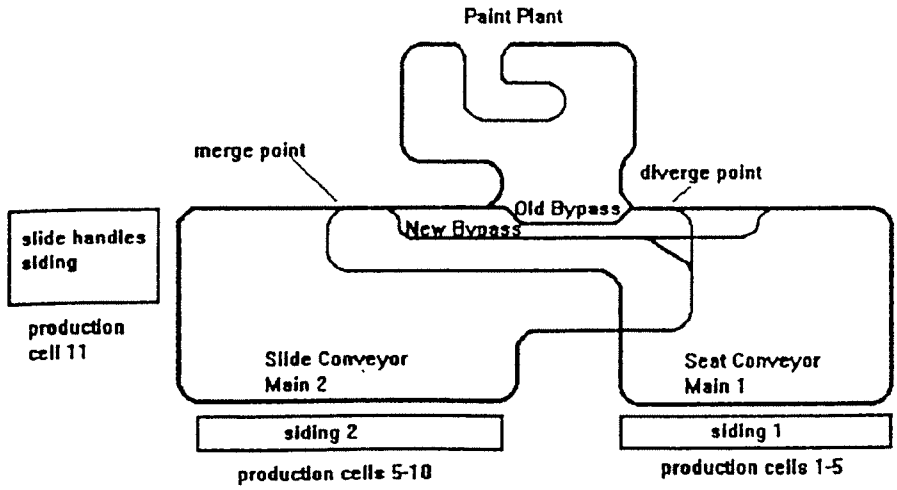


Figure 1. Overview Layout of the Conveyor System

To add to the system's complexity, there are two main conveyors in the system, each dedicated to one half of the factory. The conveyors merge before the paint plant and diverge again after the painting area as shown in Figure 1. The paint plant itself has its own conveyor. There is a siding conveyor on one side of the shop for five cells, four of which are involved in production and one in maintenance. The other half of the factory has two siding conveyors, one for five adjacent production cells and the other for a remote cell which requires a different siding structure. The conveyor system also has many other small aspects of operation that are too lengthy to mention here but are important to the efficient operation of the system. These aspects of the system have evolved over several years and are important contributors to plant performance and, as such, needed to be included in the simulation analysis.

Each cell has a number of flight bars allocated to it and no other flight bars can enter the cell. The conveyor control system is able to direct the flight bars into the correct cell by sensing the fixed bolt on the side of the bar. The investment being considered was a method of dynamically allocating flight bars to each cell during the course of operations based upon moment-by-moment supply and demand.

The Simulation Project

The steps followed to build the simulation and answer the question of "Would a dynamically programmable system be worth it?" were as follows:

- define the system to be modeled
- build the model
- collect data
- validate the model
- run experiments using the model
- revise the model and repeat experiments

Below, these steps are discussed as they were followed in this project.

Defining the System to be Modeled - Some previous work done on modeling the conveyor system provided a starting point for the Micro Saint modeling effort. This was supplemented with the plan of the electrical services layout of the conveyor system. Additionally, we make frequent visits to the company. Conveyor maintenance personnel also turned out to be helpful in gathering information about the system.

Building the Simulation Model and Collecting Data - Once a general understanding of the conveyor system had been attained the model was developed and tested gradually using the Micro Saint computer modeling system. The first version of the model including only one of the two factory conveyors and the paint plant. The model was extended as progress was made in developing the conveyor logic and collecting information about the system.

Once the building of the physical structure of the conveyor system into the model was completed, the main concern was how the system is operated by each individual cell. To ensure that the model adequately represented the real system, it was necessary to study the behavior of each individual cell by interviewing the cells leaders. The following aspects of each cell's behavior were included in the simulation:

- shift working hours
- beginning/end of shift practice (actual period of loading/unloading the flight bars)
- flight bar allocation
- maximum number of flight bars that fit onto the siding
- number of operators (loaders)
- criteria to release an empty flight bar

The development of the logic for dynamic flight bar allocation was carried out during the last stages of model development. The strategy used for assigning flight bars dynamically was based upon input

from the Managing Director of the plant as well as insights gained during model development. Also, during model testing, insights for better allocation schemes were gained and subsequently implemented in the model that was used for experimentation.

As we were gaining this knowledge and data about the system, we were concurrently building and refining a Micro Saint model of the system. In the Micro Saint model of the system, the conveyors are represented as chains of tasks, which are sections of the conveyors. Tasks form the top level network diagram of which the final version is shown in Figure 2. Some of these tasks were actually decomposed into subnetworks in the final model.

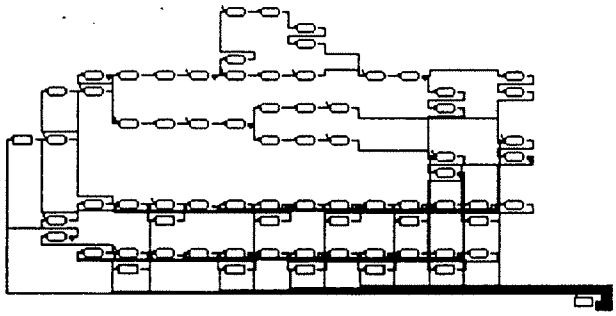


Figure 2. Micro Saint Network Diagram

The model simulated:

- The flight bars as they flowed through the system
- All flight bar queuing rules and release as in the actual conveyor.
- Routing based upon the system logic (which could be readily changed in the model).
- The potential for dynamic flight bar allocation to the work cells.

Data generated by the model included 1) the percentage utilization of the paint plant, 2) the number of loaded flight bars that were painted per unit time, 3) the number of flight bars that went through the paint plan either empty or with the same load twice, and 4) the average flight bar cycle time.

Validation and Verification - Throughout model development, validation and verification practices were carried out on a continuous basis. The validity of the model was improved as knowledge of the system increased through several interviews and visits to the shop floor.

Experimentation - The behavior of the model in eight different scenarios was studied varying three factors, 1) the existing fixed flight bar allocation systems vs. dynamic flight bar allocation, 2) the current production rate vs. a 25% increase and 3) a consistent material flow onto the conveyor vs. a variable material flow.

Results - The simulation results are presented in Table 1. The first three columns of this table describe the experimental condition. The fourth column of the table titled "paint plant utilization" is an aggregate measure of the number of cells in the paint plant vs. total paint plant capacity if used 24 hours/day. Utilization of the paint plant remains at around 50-60% because the shut-down hours are included in calculations. The fifth column entitled "number of painted flight bars" shows to the number of flight bars painted excluding empties and those flight bars going through the process twice. When a flight bar is either painted empty or painted twice because the cell was full upon return of the flight bar, this is reported in the sixth column titled "number of flight bars painted empty/twice." The seventh column refers to the overall average cycle time of the flight bar.

We were also able to plot measures over time, allowing comparisons such as that shown in Figure 3.

Table 1. Statistics Generated by the Simulation

Production level	Material flow variation simulated?	Flight bar allocation scheme	Paint plant utilization	Number of painted flight bars	Number of flight bars painted empty/twice	Average flight bar cycle time
Current	No	fixed	52.64	946	45	98.5
		dynamic	50.69	948	5	97.7
	Yes	fixed	52.64	946	45	98.5
		dynamic	50.59	948	5	97.7
Current +25%	No	fixed	61.67	1138	24	127.3
		dynamic	61.67	1147	15	130.8
	Yes	fixed	61.51	1122	37	119.7
		dynamic	61.67	1147	15	127.6

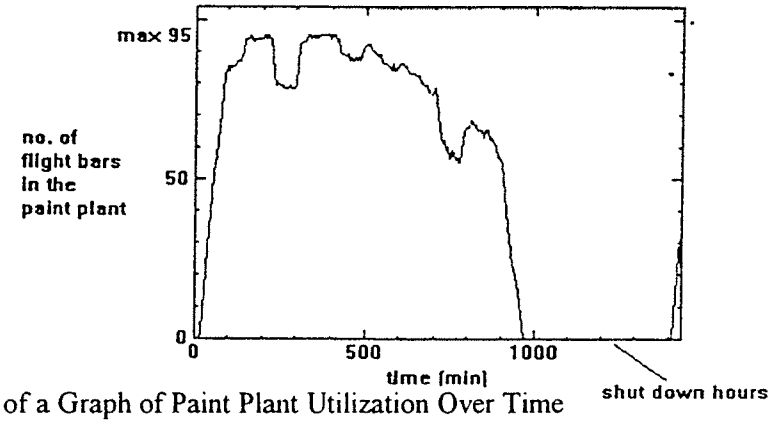


Figure 3. Example of a Graph of Paint Plant Utilization Over Time

Discussion of the Results

With the existing flight bar allocation, output goals are not fully achieved in the +25% production models. Using a dynamic flight bar allocation scheme gave somewhat better output figures, but the number of loaded flight bars released still did not reach the necessary value to sustain this production rate.

The logic used for dynamic flight bar allocation resulted in some improvements to the operations of the conveyors. The dynamic flight bar allocation decreased the proportion of empty flight bars compared with that of all four scenarios in the fixed allocation models - the current and +25% production models with and without material flow variations. There were significantly fewer empties released from the sidings and the flight bars passed by the home cell more rarely. Thus, the results indicate a more efficient use of the conveyors is possible.

However, the increased formation of queues in +25% production models, especially the one at the end of the slide conveyor, occurred even in a greater extent in dynamic flight bar versions compared with that of fixed allocation models. Hence, final conclusions about the usefulness of dynamic flight bar allocation cannot be made without a more detailed consideration of the queuing effects and potential modifications to the flight bar allocation scheme.

Summary

In all, this study provided both useful data and insights regarding the potential effectiveness of a significant capital improvement to the Dunlop Cox conveyor system. Of course, the decision to implement the dynamic flight bar allocation scheme will take into account many other factors such as projected demand, other potential plant limitations, and business constraints. However, this study did provide the decision makers with valuable projections on which these decisions can be based. Additionally, if the scheme is implemented, the simulation model can be used to fine-tune the allocation scheme at virtually no cost.

Quantitative Design of Material Handling System Using Predictive Simulation Modeling

Chris J. J. Lu, K. H. Tsai
Advanced Technology and Research Corp., Springpointe Executive Center
15210 Dino Drive, Burtonsville, MD 20866-1172, USA

and

Drs. Jackson J. S. Yang & Yu Michael Wang
University of Maryland, Mechanical Engineer Dept. College Park and Baltimore, MD

Abstract

The design of material handling system (MHS) includes the interface with existing automatic material processing machines (AMPMs), the comprehension of entire facility operation, the development of system control architecture, and the integration of all elements in the system. The staging buffer size, type of staging equipment, and required throughput of transportation links among machines are considered as the main design parameters of MHS and should be addressed first. This paper provides a generic numeric method to determine the above design parameters using predictive simulation model. Moreover, other interested system characteristics can be analyzed by using this method as well. All the material handling devices met the found requirements will be capable of handling the load of material in the automated system and may be considered as the candidate equipment. The criteria of selecting the final design and facility from these candidate equipment are based on the allowance of capital, floor space constrains, equipment reliability, and designer's preference.

1. Introduction

The plantation of automatic material processing machines, the installation of material handling equipment (MHE), and the integration of automation system are considered as the three major design elements in any type of facility automation. A generic method using the accelerated simulation was proposed to determine the type and number of AMPMs and to optimize the operation schedule [1]. The traditional design of MHS involves manual recording and experts consulting. This type of analysis is usually too subjective and unsystematic. Simulation methods are developed to overcome this obstacle.

A numeric determinative method, predictive simulation, is used in this study to find the required design parameters of MHE, such as the required minimum buffer size of the staging device and throughput capacity of the transportation equipment. Moreover, the type of staging device, FIFO or intelligent retrieval (I.R.), may be determined as well. Once the parameters of MHS and AMPMs are obtained, the theoretical optimized facility layout can be achieved by the topology optimization concept [2, 3]. A life cycle automation system can be then integrated and implemented by applying the accelerated simulation, the predictive simulation, topology optimization concept, Real-time Control System methodology (RCS), and facility layout CASE tool [4, 5].

2. Predictive Simulation

A successful system design must start with a set of good data. Thus, to acquire fundamental knowledge of the entire factory is the first step while designing the MHS. Certain common characteristic information may be categorized as a standard format for all types of MHS. The required design parameters can be determined once the standard dada set is ready.

The predictive simulation model applies the Markov chain simulation that predictively determines the paths of material from machines to machines. The predictive simulation can be run faster than the accelerated simulation and real-time simulation [1]. The assumption of the predictive simulation is that every object in the model will behave as its pre-assigned characteristics. Every automatic machine is operated at regular condition and each material is sent to an appropriate destination and is processed at the earliest period as soon as this material is handled by the AMPM. The predictive simulation is composed of a series of discrete sampled simulation period. However, the simulation is pseudo-continuous within each sampled period. Predictive simulation is perfectly suitable for analyzing a complex and large scale system while applying the traditional simulation is too tedious. The merits of predictive simulation are its extremely high simulation rate and simplicity of simulation algorithm.

2.1 Nodes Modeling

In this simulation model, AMPMs and MHE are considered as nodes and links in the network graph, respectively. Each node serves as an individual device emulator to simulate quantitative characteristic of the hardware device in the system. N different input buffers are assigned to each node if there are N operation plans exist in the designed facility. Only one operation is allowed to run at a node within one sampled period. Each node handles the appropriate material according to its operation plan. The simulation of handling event is called node operation and is modeled by applying the Monte Carlo method and Queuing algorithm. After the material has been handled, the characteristic of that material will be updated by the node. According to this updated characteristic, the material is then sent to the appropriate input buffer of the next destination node according to the earliest availability criteria (EAC).

The data structure of a single node, $\mathbf{Node}_{j,k,l}$, is a three dimensional array, where j, k, and l represent the operation plan ($\mathbf{OP}$), sampled period ($\mathbf{P}$), and simulation properties, respectively. Four entries, material processing capacity, received material volume, proceeded material volume, and miss-cutoff material volume of $\mathbf{OP}_j$ at $\mathbf{P}_k$, are included in the simulation property. The data structure of the nodes model is a four dimensional array and can be expressed as $[\mathbf{Node}_{j,k,l}]_i$, where i is the number of nodes.

The maximum volume that can be handled by a node (automatic machine) during a single simulation sampled period is called the capacity of that node. The processed material and the received material volumes are updated as soon as the material is handled by the source node and assigned to the new destination node. The node simulation mechanism increases the processed volume in the source node and the received volume in the destination node by one unit.

The transportation time for every material between nodes is not considered because the perfect material handling system is assumed [1]. Therefore, to achieve the objective (maximize the throughput of the material flow) is to send the material to the earliest available automatic machine resource. Because the volumes of throughput, capacity, and the processed material of each node are "real-time" updated in the simulation world, the destination node may be determined as the node with minimum queue length. Such a pre-determined algorithm is used to determine the destination node as soon as the material has been handled by the source node in this model. Moreover, the volume of processed material should be equal to the volume of received material of each node at the end of each period if the perfect MHS is applied. This criterion may be used to detect simulation errors.

Two important features, miss cutoff and FIFO (First In First Out), are implemented in this simulation model. First feature is to test if the material will miss its cutoff time for the destination operation. Cutoff time is the latest time that one operation is allowed to run without delaying the downstream operation during a unit operation cycle (one day). Cutoff time can be mathematically derived by using Eq. (1) once the final dispatch and the throughput rate of each operation are known.

$$\begin{cases} P_{cur} \le P_{des} \le P_{cut} & : \text{Make the Cutoff Time} \\ P_{cut} < P_{des} & : \text{Miss the Cutoff Time} \end{cases} \quad (1)$$

The second feature is to analyze the FIFO queuing property for the input buffer of the destination node. According to the queuing theory, a node does not need such a device if its arrival materials are in a FIFO order. When the material is sent to its destination node, the destination operation period has been determined. Thus, a node requires an automatic intelligent retrieval type device if there is any material in its queue with a later destination period than the destination period of a new arrival material. If a material is handled and sent to the new destination node, $\mathbf{Node}_{dest}$, with a destination operation period, $\mathbf{P}_{dest}$, the type of input buffer device of $\mathbf{Node}_{dest}$ can be determined by using Eq. (2),

$$\begin{cases} \sum_{i=Pdes+1}^{Max\_P} Rec_i = 0 & : \text{FIFO Type} \\ \sum_{i=Pdes+1}^{Max\_P} Rec_i > 0 & : \text{Intelligent Retrieval Type} \end{cases} \quad (2)$$

where $\mathbf{Rec}_i$ is the received material volume of $\mathbf{Node}_{dest}$ at $\mathbf{P}_i$.

Another important quantitative design parameter is the minimum required buffer size of each node and can be determined if we have the time series of the peak number of the material volume in the buffer at all sampled periods. The volume of material in the buffer ($\mathbf{Buf}_{i,j}$) of $\mathbf{Node}_i$ at $\mathbf{P}_j$ is given in Eq. (3).

$$Buf_{i,j} = (Rec_{i,j} - Pro_{i,j}) + \sum_{k=j+1}^{k=Max\_Period} Rec_{i,k} \quad (3)$$

where $\mathbf{Rec}_{i,j}$ and $\mathbf{Pro}_{i,j}$ represent the received and processed material volumes for $\mathbf{Node}_i$ at $\mathbf{P}_j$, respectively.

The minimum buffer size of a particular $\mathbf{Node}_i$, $\mathbf{Min\_Buf}_i$, is the peak number of all the required buffer size of $\mathbf{Node}_i$ at all simulation sampled periods, as shown in Eq. (4)

$$Min\_Buf_i = \text{Max.}(Buf_{i,j}) \qquad j = 0,\ldots,Max\_Period \quad (4)$$

2.2 Links Modeling

The minimum required buffer sizes of nodes and throughput capacities of links which connect nodes are the two major design parameters of MHS. The throughput capacity in this simulation model is defined as the volume of handled (transported and delivered) material within a time unit (sampled period). $\mathbf{Link}_{i,j,k,l}$ is used to represent the characteristics of all links in this predictive simulation model, where i, j, k, and l represent sampled period, source node, destination node, and simulation statistic properties, respectively. Three entries, minimum buffer size of destination node, material volume received by destination node at destination period, and material volume sent from source node at current period, are included in the simulation statistic property. A delivered material is staged in the local input buffer of its destination node until its destination period.

The values of both $\mathbf{Link}_{i,j,k,1}$ and $\mathbf{Link}_{i,j,k,2}$ are updated (through a function call) when the material is handled and sent to the new destination node. The minimum required buffer size for $\mathbf{Node}_k$, $\mathbf{Min\_Buf}_k$, can be determined by Eq. (5) once $\mathbf{Link}_{i,j,k,l}$ is found.

$$Min\_Buf_k = Link_{i,j,k,0} = \sum_{p=i+1}^{p=Max\_P} Link_{p,j,k,1} \quad (5)$$

This simulation model is a closed system, therefore no material will leave or disappear from the system. Eq. (6) shows a useful balance relationship to check for leaking error of simulation models.

$$\sum_{p=0}^{p=Max\_P} Link_{p,j,k,1} = \sum_{p=0}^{p=Max\_P} Link_{p,j,k,2} \quad (6)$$

The simulation time will be increased by the amount of "time factor" for every simulation cycle. The behavior of the simulation model is simulated based on the simulation time and the elapsed time. The larger the value of time factor is, the faster the simulation is. However, the simulation resolution will be decreased as the time factor is increased.

2.3 Simulation Modeling Summary

The capacity of each node at every sampled period is determined at the initialization stage using "Op.data" and "Param.data". Materials are then introduced into the simulation model by a system input device simulator. This simulator is modeled as a function and is called whenever the sampled period changes. Each material is simulated to be handled and assigned its coveted operation plan by applying Monte Carlo method and Queuing algorithm. The appropriate destination node of

[1] A perfect material handling system is an intelligent system which will automatically deliver every material in the system to a desired location at an appropriate time. Moreover, the capacity of the system is assumed infinite.

this handled material is also determined using the EAC. If no unexpected malfunction during the operation is assumed, the earliest node that will be able to handle certain materials in a period is the node with the smallest quantity in its operation buffer. This unique predictive property is suitable for the purpose of system design and analysis. The limitation of the predictive simulation is that one material can not go through two nodes within one sampled period. Thus, the sampled window effects the fidelity of the simulation result. The entire simulation flow chart is shown in Figure 1.

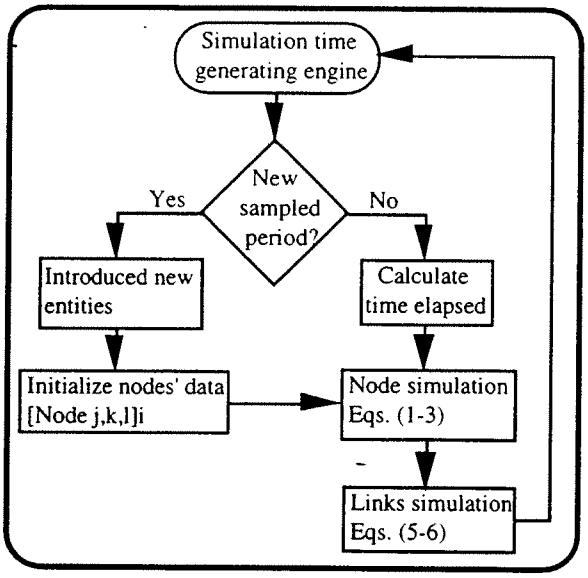


Figure 1. Predictive Simulation Algorithm Flow Chart

3. Example - USPS P&DC, San Diego, CA

The U. S. Postal Service San Diego mail processing and distribution center (P&DC) is a large scale facility in California. This facility is planned to extend and re-layout as an optimal P&DC by 1996. The predictive simulation is used to analyze the mail flows, staging requirements, and several other design parameters. This facility handles three to four million mail pieces every day, including letter and flat mail which have separate independent properties. In this paper, we use the letter mail analysis as an example. There are at least five major types of mail processing machines needed to be used to process the letter mail. They are Optical Character Readers (OCRs), Bar Code Sorters (BCSs), Letter Sorting Machines (LSMs), Distributed Bar Code Sorters (DBCSs), and Letter Mail Labeling Machine (LMLMs).

3.1 Standard Design Data Set Acquisition

The first step of this methodology is to acquire the standard data set that describes the characteristics of the entire system behavior. The standard data files of the facility provided by the San Diego field experts and are shown in Figure 2.

3.2 Simulation Modeling & Scenarios Setup

The simulation model of the San Diego P&DC was implemented in Language C and ran on Silicon Graphic Onyx Reality Engine-2 computer. Input data files were automatically read in while this software was running. A tray was used to represent the unit of a material (entity) and contained 300-450 of mail pieces. The simulation was setup to run a five day scenario. The output report files were generated at the end of the simulation. The analyzed data in the output report were extracted from the middle three days (second, third, and fourth day) for the accuracy purpose.

1. Volume.data-material volume of each input operation plan.
2. Profile.data-input profile distribution over 24-hour period.
3. Param.data-operation plan characteristic parameters, such as throughput, bin used number, bundleable, flush tray number, and order of operation plan.
4. Dens.data-bin density distribution of each operation plan.
5. Op.data-operation plan schedule over 24-hour period.
6. Cutoff.data-cutoff time of each operation.
7. Flush.data-machine flush schedule over 24-hour period.
8. Manu.data-units of manual sorting at each sampled period.
9. Center.data-served operation plans of center storage devices.
10. Node.data-general information of each node (AMPM).

Figure 2. Standard Data Set of San Diego, CA, P&DC.

3.3 Simulation Results

In this study, simulation searched all possible paths that mail would travel within the P&DC. The data describes each path which includes not only its source and destination, but also the throughput volume at all sampled periods.

3.3.1 Minimum Required Buffer Size & Types

One of the main goal of this simulation analysis is to find the minimum required buffer size of the MHS. Table 1 shows the detail data of the minimum required buffer size of every OCR extracted from the output report file of the simulation. Two different analysis methods, Node analysis - Eq.(4) and Link analysis - Eq.(5), are both applied in the simulation model. The result of these two methods are similar to each other. The larger buffer size of the results derived from these two methods should be used as a design base parameter for the safety purpose. Safety factors should be also applied to these parameters when the buffer device is designed and implemented.

Machine (Node)	Min. Buf (Node Analysis)	Min. Buf (Link Analysis)
OCR-0	154	153
OCR-1	150	144
OCR-2	129	130
OCR-3	102	84
OCR-4	102	82
OCR-5	121	100
OCR-6	180	166
OCR-7	200	180
OCR-8	150	132

Table 1. Minimum Required Buffer Size of OCRs.

Because of interchangeable properties, it is logical to arrange the same type of machines into like-groups, similar to a farm layout. All of the paths that previously went from machine to machine are now combined with similar paths to form new paths that carry trays of mail from group to group. Figure 3 shows the 5 day buffer size time series of OCR-0, OCR group, and P&DC. Another useful information for designing the staging equipment is the type of retrieval device, FIFO or intelligent retrieval (I.R.). Table 2 shows the required volume of buffer size and staging type of all groups in the P&DC. If the staging design is a centralized storage device system, the

minimum buffer size of the center storage device is 5950 trays applying the same method. OCR group is the only group that does not require an intelligent retrieval staging device. This is because the OCR group is the first mail processing station that all types of incoming mail may be processed in a random order.

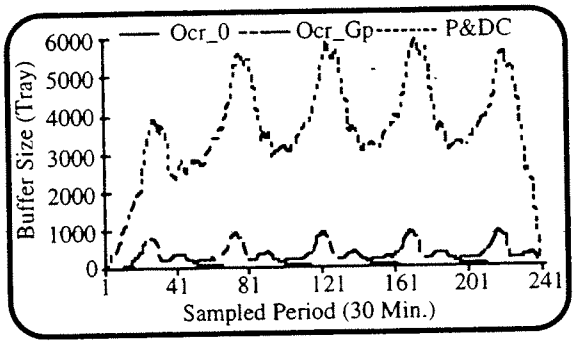


Figure 3. Five Day Buffer Size Time Series of OCR_0, OCR Group, and P&DC.

Ocr	Bcs	Dbcs	Lsm	Lmlm	Manu	P&DC
938	925	2997	919	260	747	5950
FIFO	I.R.	I.R.	I.R.	I.R.	I.R.	N/A

Table 2. Min. Required Buffer Sizes of All Nodes' Groups.

3.3.2 Maximum Throughput Capacity Requirement

Another important design parameter of the MHS is the required throughput capacity of each link that connects nodes. The predictive simulation provides the complete throughput time series of every link. Figure 4 illustrates the throughput time series for BCS group. The minimum required throughput capacity is the peak value of the time series. Table 3 shows the minimum required throughput capacity matrix for all the machine groups in the P&DC.

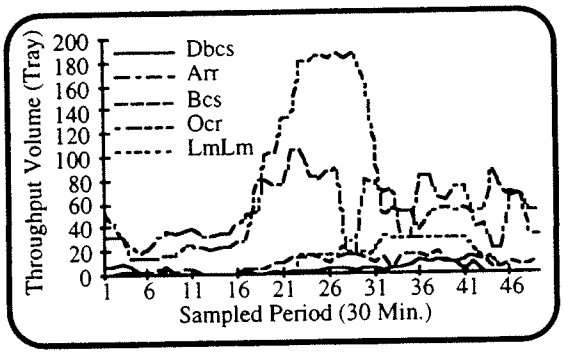


Figure 4. Throughput Time Series for Destination: BCS Group .

3.3.3 Operation Characteristics

The steady state status of volume of miss-cutoff tray and the volume of turn around mail are the most important aspects in the P&DC operation. The phenomenon of increasing volumes of miss-cutoff trays and turn around mail over time is considered as a result of a poor design because it means the machines are not capable of handling all the input mail. The steady state status of both miss-cutoff mail and turn around mail volumes are observed from the result of simulation.

From \ To	Arr	Ocr	Bcs	Lsm	Mamu	Dbcs	Lmlm	Disp
Arr	0	320	105	59	57	78	0	0
Ocr	0	0	187	28	0	156	0	227
Bcs	0	35	18	37	0	254	19	328
Lsm	0	0	0	245	40	0	0	1651
Manu	0	0	0	0	34	0	0	158
Dbcs	0	34	11	40	0	266	0	532
Lmlm	0	0	31	0	0	0	0	0
Disp	0	0	0	0	0	0	0	0

Table 3. Throughput Requirements for All Groups in the P&DC.

4. Conclusion

The ultimate purpose of this study is to provide a generic method to design an optimal automated facility. The material handling system is usually the most important component of a modern system automation. This paper proposes a quantitative analysis method, predictive simulation, to synthesis the required design parameters for the MHS. These parameters include the minimum required throughput capacity of every path, minimum required buffer size of every staging device, the types of staging buffer devices, and analyzed data of the information of every automatic machine and every operation.

The facility layouts found by applying this method are the best layout that can be derived from mathematical model. The system behavior should be simulated in its entirety before any components are installed on the floor of the facilities. A CASE tool called CSAT provides all these perfect features for integrating the entire automation facility and supporting the life cycle system design. Once all these stages are accomplished to the satisfaction of the end users of the system, the implementation of the final design can be initiated with the confidence that the system will be reliable and efficient.

Acknowledgments

This research was supported by the USPS Headquarters under Task Order #104230-88-D-2546. The authors wish to thank Dr. Anthony Barbera, Mr. Clyde Findley, Mr. Phil Feldman, and Mr. Keith Breadford for their advice and discussion.

References

[1] Lu, J. C., Tsai, K. H., Yang J. C. S., and Wang M. Y. "Real-Time Operation Plan Optimization Using Accelerated Simulation Modeling", International Journal of Modeling & Simulation, Accepted for publication.

[2] Findley, C., Feldman, P., and Lu, J. C., "An Analysis of Material Handling and Mail Processing Equipment Configuration in USPS Processing & Distribution Centers", Final Report of USPS Automation Research Project, Washington DC, May, 1993.

[3] Yang, J., and Lu, J. C., "Optimized Design For Automation Systems Using 4-D Simulation", Proceeding of International Workshop On Functional Modeling Of Complex Technical Systems, Ispra, Italy, May 13-14, 1993.

[4] Yang, J., and Lu, J. C., "A Life Cycle Automation Toolset Using The Real-Time Control System", Proceeding of US/ROC Workshop On Automation, Taipei, Taiwan, ROC, July 1-4, 1993.

[5] Yang, J., and Lu, J. C., "Developing A Life Cycle Toolset For The Modern Automation System", Proceeding of Sino-American Technology & Engineering Conference, China, October 11-15, 1993.

Dipl.-Math. Veit Popp, ExperTeam SimTec GmbH, Dortmund

Simulation for Production Scheduling: Integration of Simulation Techniques into Planning and Scheduling

1. Why simulation...

The problem of production planning and scheduling is characterized by a simultaneous scheduling of all resources and materials needed. Traditional MRP systems use scheduling algorithms that do not represent the finite capacity of a manufacturing system precise enough and do not provide a means for analyzing a manufacturing system's ability and agility to satisfy diverse customer demands through the use of different planning strategies and scheduling technologies.

This is the point where event based simulation comes in and provides techniques for a total coordination of the elements of the production process, giving a clear understanding of the details of all processes that influence the manufacturing performance. Simulation predicts the impact of each order on manufacturing performance and the production process. As all characteristics of the production system and all capacity constraints are represented in the simulation model (strategic, operating and procedural aspects), a better understanding of the dynamics of the production system is achieved. Comparing results of different simulation alternatives provides a better communication based on facts for planners and schedulers.

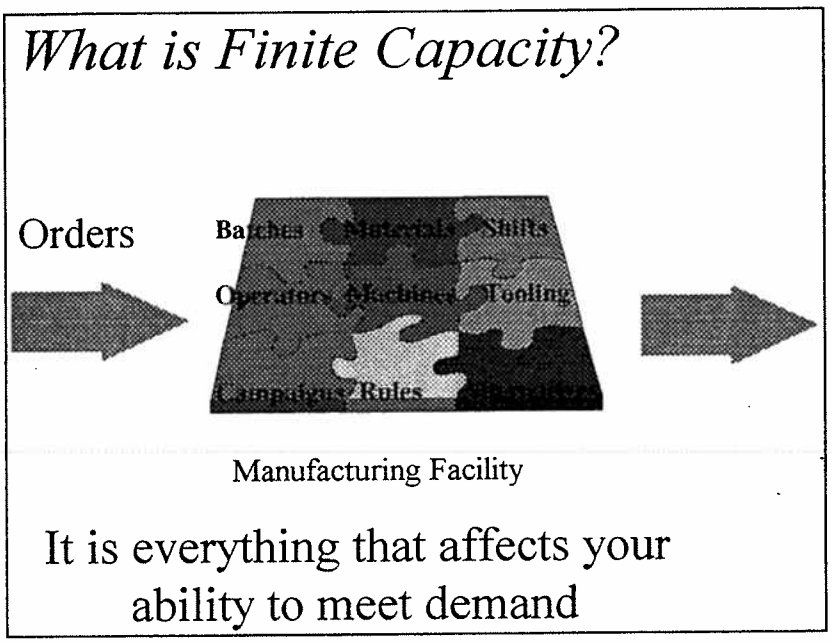
The problem of the representation of the complexity of the production process in a model - the "factory in the computer" - is simplified if the organizational and structural models developed in a capacity design system can be transfered and reused for finite capacity planning and scheduling for production control. Especially simulation models for evaluating capacity design questions and production strategies can help to understand the dynamics of the production process and provide a template for implementing a scheduling model.

The benefit of event based simulation for scheduling is well acknowledged, therefore one of the main demands on MRP systems in the future is providing "real" simulation techniques.

2. Modeling

A simulation of a manufacturing system provides a realistic portrayal of its detailed operations from which its performance can be accurately estimated. Because all aspects of the manufacturing system, including management's operating philosophies and procedures, are contained in the computer model, a common understanding of the planning and scheduling functions is established. This allows for comparisons of alternatives and facilitates communication to managers and executives of any problem areas and improvement possibilities. Simulation implements the operation strategy and creates feasible, coordinated and intelligent schedules.

Because manufacturing systems are complex and large, simulation systems have been specifically developed for them. A good manufacturing simulation system provides the capabilities as follows: allows diverse scheduling-development philosophies, demand-driven simulation using actual orders and build-to-plan forecasts, bill of materials to any level of detail, multiple resource requirements and constraints for processing steps, detailed process planning procedures, menu-driven interactive query functions to support reporting and problem analysis using manufacturing terminology, future projections of order status and displays of operation contention in specific time periods, advanced built-in graphics with automatic chart generation, RDBMS with complete data structure specificity, manufacturing specific functions, problem configuration and data collection specification, capabilities for multiple performance measurement and estimation.

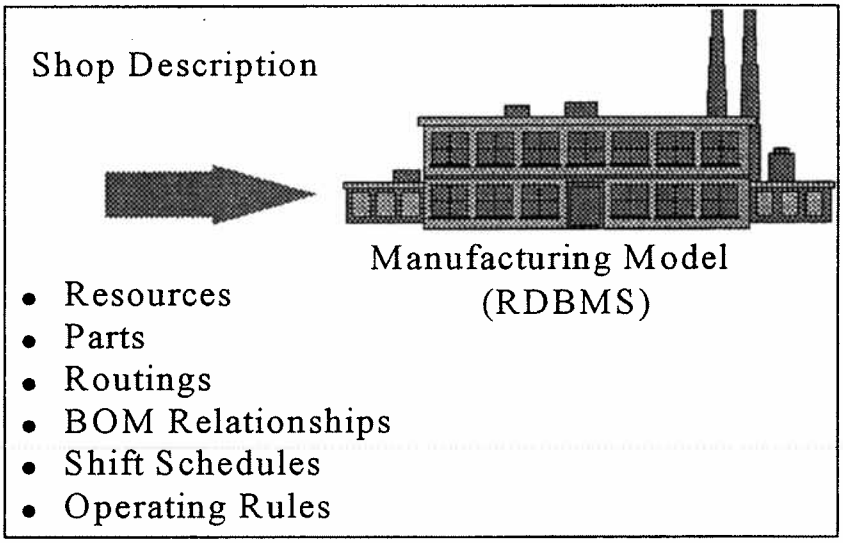


Traditional simulation systems need great expenditure of modeling work and often are inflexible to support permanent model modifications. Because of the amout of data that has to be processed (orders, process plans, jobsteps) the computing performance does not allow to use a software implementation based on a simulation language to be interpreted at runtime. A new approach supports both requirements: easy but accurate modeling and runtime performance. The event based simulation - also called time dynamic simulation - is used to combine the production data model available in MRP systems with the well known rule-based queuing algorithm.

One of the main advantages of the simulation system *FACTOR Production Manager* is that is makes use of the all the data relevant to the manufacturing process available in MRP and shop flor data collection systems.

The underlying model consists mainly of data already available from MRP systems: orders, bill of materials, process plans, resource data, shift calendars, maintenance intervals, personal with qualification for specific jobs etc. Building the model is reading that data through an interface from the MRP and shop flor data collection system into the simulation system. The interface has to be "intelligent" to provide some kind of modeling tasks: bill of materials and process plans are connected by assigning material needed to specific job steps, fixed set-up times have to be converted to part-depending set-ups and rules when to apply, fixed waiting times are eliminated (perhaps considering transportation time) as the time a load is waiting in a queue for processing is evaluated by the simulation depending on the dynamic situation.

In the simulation model generated, all aspects of the specifics of the shop decription is taken into account: operating strategy, multiple finite capacity constraints of job steps and technological sequencing rules.



3. Simulation

Simulation can now be performed by reading the MRP order data and the actual status of the manufacturing system (by a data collection system) on a day-to-day basis. The simulation takes into consideration all technological, strategic and manufacturing rules that were implemented at every resource. Sequencing is done by selection rules at resources that can be controlled by global rules (Earliest Due Date, Least Dynamic Slack, etc.) or resource specific rules (Minimum Set-Up, etc.). By using the multiple constraints defined in the job steps the simulator determines the best fit of strategies selected based on the actual jobs to be performed.

Simulation can be repeated with parameter adjustment as often as desired to satisfy the actual goals of manufacturing (meet due days, minimize set-up, decrease inventory level etc.). Simulation is rerun if new order data is available, a severe breakdown conflicts with the actual schedule or the actual status varies considerably from the schedule. Especially for the last item a degree of deviation from the schedule has to be defined and a threshold that indicates the necessity for reruning the simulation based on the actual status of production. This is caused by accumulation of small breakdowns or a conflict between strategic goals and their feasibility.

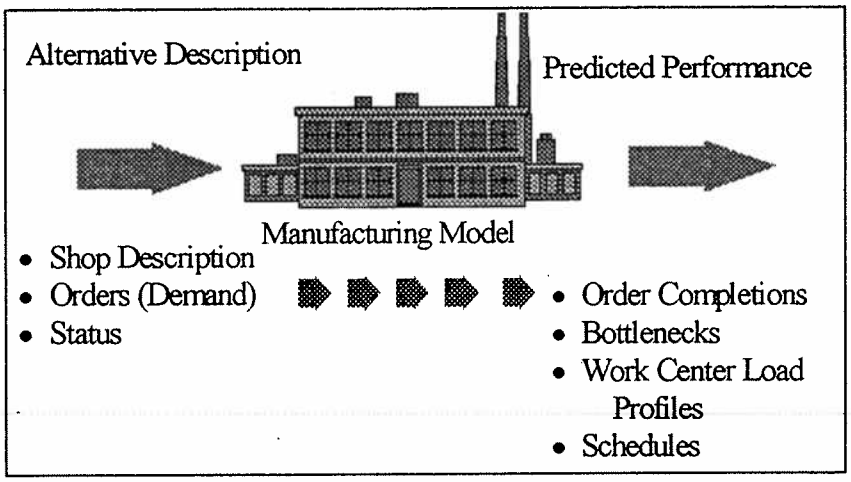
4. Results

When a relational database system is part of the scheduling system, it is easy to work with results of the simulation to obtain information on why the manufacturing system representation behaved the way it did, that is, produced a specific plan. Global and specific information is available: "Are there orders late, which orders are late, which job steps waited for the resource that caused the lateness, which are the bottleneck resources and when, did the job step wait for a resource or was material not available" etc.

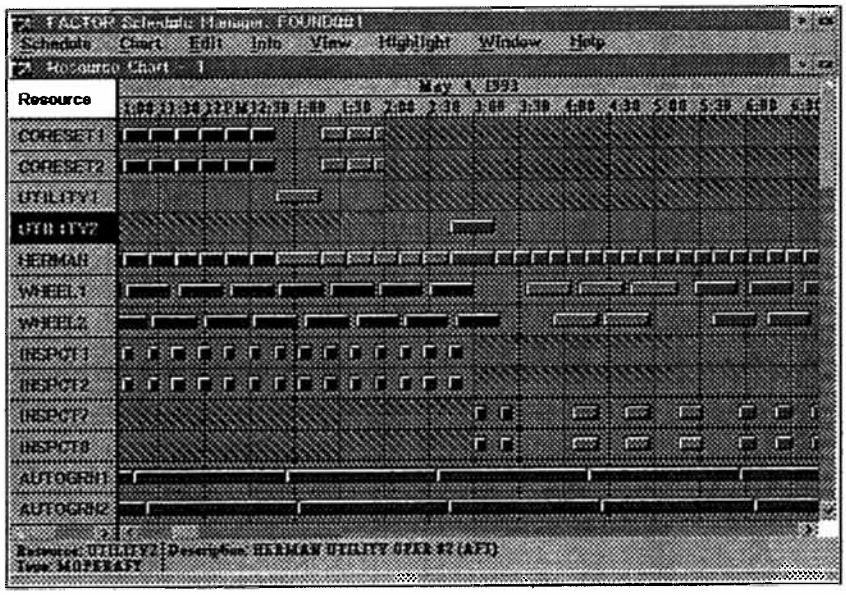
Also very detailed information is available: for each job step the simulation determines

- when are resources and materials allocated by requests of specific job steps
- when starts the operation, e. g. are all resources and materials needed available
- when is the operation finished and frees all resources not needed any more.

Simulation provides a feasible and intelligent schedule not only for resource allocation but also for material requirements, thus supporting JIT concepts.



Alternatives allow for what-if-scenarios to explore more advanced operating strategies. The final schedule then is used for execution, detailed material plans and feasible promise dates.



5. Summary

The advantages of event-based simulation for planning and scheduling are:

- simulation allows realistic and technological scheduling and the ability to perform various what-if-scenarios to develop the best strategies that meet the company's goals
- no complex modeling is needed as the model is generated mainly in the data interface to the MRP system and the sequencing rules at resources
- simulation always produces a feasible and intelligent schedule
- produces all necessary data to get full transparency of the manufacturing process and to feed back results to other systems involved in the manufacturing and decision making process
- considers the actual status of the production system when simulation is performed and therefore provides a full control loop for the manufacturing process
- schedulers are free to solve real problems, not ones we can predict ahead of time.

Top Down Design with VHDL-A

Richard Trihy and Kenneth Kundert
Cadence Design Systems
trihy@cadence.com

Abstract

Traditionally circuit simulation has seen significant application in the verification of integrated circuits and systems. Its application to system level design has been limited by the need to express circuit and multi-disciplinary model behavior in terms of a limited number of built-in circuit primitives. However the new Analog Hardware Description Languages (AHDLs) provide the needed flexibility and functionality to describe behavior at the architectural or system level.
This paper discusses the benefits of AHDLs in the design of circuits and systems. By bridging the gap between higher level behavioral or system level descriptions and circuit level descriptions an AHDL, such as VHDL-A brings structure and continuity to the design process. Language constructs that "shield" the user from the underlying simulator complexities allow designers to efficiently utilize the behavioral modeling capability. The top down design of a 9-bit A/D converter using Cadence's prototype implementation of VHDL-A is presented as an example[1].

1 Introduction

Simulation of analog circuits and systems as performed by traditional simulation strategies is time-consuming and cpu-intensive. The system to be simulated must be described in terms of a limited set of built-in primitives, consisting of resistors, capacitors, transistors, controlled sources etc. While the list of primitives may include higher level behavioral macromodels, the design or architecture must still be expressed in terms of this limited set of building blocks. This requirement acts to restrict the designer when generating a higher level description of a system's behavior.

The cost of circuit simulation is primarily affected by the size or number of nodes in the circuit as well as the complexity of the component models that must be evaluated at each iteration. Hence in order to speed up the simulations the size of the circuit should be reduced and its models simplified.

AHDLs allow the user to describe the essential behavior of a system more easily than is possible with traditional circuit simulators. By reducing the circuit and model detail, faster simulations result. The ease with which system level behavior can be expressed and then simulated opens up a number of useful applications that were previously cumbersome or even impossible.

This paper addresses the capabilities of SpectreHDL[3], a mixed abstraction tool that combines different AHDLs, e.g. VHDL-A, Verilog-A, and circuit level descriptions. There are two key aspects to SpectreHDL's languages. SpectreHDL employs an equation formulation methodology that treats across and through (voltage and current) sources symmetrically and independently of the simulator's underlying matrix formulation. This makes the language easier to use and understand[2].

The SpectreHDL langauges support a set of built-in functions that can be used to model a variety of different behaviors. A set of functions for shaping waveforms and controlling the underlying simulator engine are provided. This paper illustrates by example the application of SpectreHDL to the top-down design of circuits and systems.

2 Top Down Design

Traditionally post layout verification of integrated circuits has been the dominant need of high performance analog simulation. Analog circuit simulation does find application late in the design process but only after most of the detailed design decisions have been made. Other more behavioral approaches are often taken at the architectural level of the design. These tools do not extend to circuit level simulation.

The evolution of flexible AHDLs, such as VHDL-A, that permit the merging of accurate circuit level analog simulation and mixed signal simulation with higher level behavioral descriptions has introduced new applications for these tools in the design of analog and mixed signal circuits and systems.

This uniformity in the design tools between system and circuit level descriptions opens up new possibilities for the design process.

- The system under design can be described and simulated at a high level of abstraction early in the design process. This aids design experimentation and model reuse. The essential system behavior and its dependence on component nonidealities can be identified with the aid of simulations that would be prohibitive at the circuit level. This is illustrated in the case of a 9-bit A/D converter below.

- These higher level descriptions serve as a communication tool between the circuit and system level designers. They document the interface between system modules. Furthermore the AHDL descriptions can be used by the test engineers to debug the early versions of their testbenches.
- Functional level models for the system elements can be used to verify the connectivity of the circuit. Errors in circuit connectivity is a common and potentially very expensive mistake. It can be avoided with inexpensive simulations that employ AHDL models that characterize the interface of the system elements.
- AHDL can be used to construct measurement blocks or testbenches for analyzing components or circuit behavior. For example a standard testbench for opamps might perform a number of analyses to measure offset, slew rate etc. of different opamp circuits. For top-down system design the measurement blocks perform measurements of interest and they can be reused across abstraction levels. As the circuit details of the system are refined and expanded the testbench serves as a check on the behavior. Figure 1 shows the block diagram for an A/D converter and it includes a testbench to measure INL and distortion. It will be demonstrated in the next section.
- By mixing levels of abstraction it is possible to speed up the resulting simulations. Key pieces of the design can be described at the circuit level, while the remainder is modeled at the behavioral level.
- Finally AHDL allows for analyzing complex systems that employ multi-disciplinary models. Models of different disciplines can be combined in the system description to investigate the overall system (and not just electrical) behavior.

3 Design Example: 9 bit A/D

This section describes the top-down design of a 9 bit A/D converter. The example is designed to illustrate the desirability of developing testbenches early in the design process as well as to give a flavor of the types of analyses that can be inexpensively performed with behavioral models of the system.
Five levels of design are implemented for this A/D component. At its most simple level it consists of a single AHDL module description and the most complex version mixes behavioral and circuit level descriptions of the entire system.
The converter is a parallel pipeline implementation. It consists of four parallel paths that are time multiplexed to increase throughput. Each of these paths consists of a four stage 9 bit pipeline A/D converter with digital error correction. A block diagram of the circuit is shown in figure 1.

3.1 Simple Model

Using the most simple system model, suitable measurement blocks can be developed and tested. These measurement blocks are reused repeatedly as the design advances. They serve to ensure that the design is correct, and they identify the effects of component nonidealities.
The measurement blocks that are implemented are an ideal 9 bit D/A converter which allows the reconstructed output analog waveform to be compared with the input waveform and an integral nonlinearity (INL) measurement block. The INL block calculates the error between the input analog signal and the reconstructed output waveform. This integral nonlinearity error is expressed in terms of least significant bits (lsbs).
Furthermore we can perform fourier analysis on the reconstructed output waveform to investigate the distortion effects of various nonidealities in the system.

3.2 Analysis of pipeline converter

The first implementation of the pipeline converter consists of three 3-bit stages, which are combined to produce a single 9-bit code. In addition an implementation of the converter that uses an additional stage and employs digital error correction[1] to remove error due to component nonidealities is created. Diagrams of the 4-stage pipeline converter and the expanded 3-bit stage are given in figure 2. AHDL descriptions are written for the 3-bit stage, as well as for the input sample and hold and digital combinational logic.

To test this error correction feature, both the correcting and non-correcting A/D's were simulated with A/D nonlinearity error infused into the models. A VHDL-A module for this nonlinear 3-bit A/D is shown in template 1. The nonlinearity is modeled by introducing random offsets at the A/D transition points.

Figure 3 shows a portion of the output waveforms from the INL measurement block (expressed in lsbs) for 3 different transient simulations. The first plot shows the quantization error for a pipeline converter with no nonidealities. The second plot is for the case of a nonlinear A/D converter. The third plot is the result of simulating the circuit with nonlinearity and additionally digital error correction. This shows that the digital error correction does indeed remove some nonlinearity in the system's components. It would be prohibitively expensive to perform these simulations at the circuit level.

3.3 Parallel pipelines

The 4 bit stages can now be combined in parallel with an analog time domain multiplexer to complete the overall system. The interesting nonidealities that can be examined here are the effects of path mismatch on the harmonic content of the output waveform.

The effects of gain and offset mismatch were investigated. For a sampling frequency (Fs) of 10MHz and an input sine wave with frequency of 625KHz (Fs/16), the harmonics produced from gain mismatch are shown in table 1 below. This data was obtained from a transient simulation followed by a fourier analysis of the reconstructed output waveform.
It can be seen that the dominant harmonics occur at 3Fs/16, 5Fs/16, 7Fs/16, 9Fs/16, which is the behavior predicted in[1].

Next the circuit was simulated with a D.C. input, and gain mismatch was replaced by offset mismatch. In this case the dominant harmonics were seen to occur at frequencies of Fs/4, Fs/2 and 3Fs/4, i.e at multiples of the multiplexer sampling frequency, as expected.

3.4 Mixed abstraction representation

The final representation of the circuit replaces the AHDL description of the sample and hold with a detailed circuit (transistor) level representation, see figure 4 The transient simulation results for a 200KHz sine wave and an A/D clock frequency of 10Mhz are

shown in figure 5. The first plot shows the input sine wave and the reconstructed output waveform, which is generated from the ideal D/A probe measure block. The second plot shows the reconstructed waveform for one of the A/D channels and the last plot shows the reconstructed waveforms for all four channels. The true output of the A/D is a composite of these waveforms.

Table 2 compares the simulation times (on a SPARCstation 5) of the mixed abstraction circuit with a purely AHDL implementation. There is an order of magnitude speed difference.

4 Conclusion

This paper has illustrated the benefits of an analog hardware description language for top down design of integrated circuits and systems. The ease with which circuits and systems can be described and simulated with AHDL opens new possibilities for design methodologies. Investigation of design trade-offs can be performed efficiently at the architectural level and measurement modules can be written to quickly analyze the circuit or system behavior across different levels of abstraction. In addition standardization will increase model availability, interchange and reuse.

References

[1] Cormac S. G. Conroy, David W. Cline and Paul R. Gray. "An 8-b 85-MS/s Parallel Pipeline A/D Converter in 1um CMOS", IEEE journal of Solid State Circuits, vol 28, No. 4, April 1993.

[2] Daniel FitzPatrick and Kenneth Kundert. "Simplified approach to formulating Analog Behavioral Models", ICEHDL Las Vegas, Nevada, January 1995.

[3] SpectreHDL Reference Manual. Cadence Design Systems.

Table 1: Harmonics for Gain Mismatch

Harmonic	Relative Magnitude
1	**0 dB**
2	-280.573 dB
3	**-32.0978 dB**
4	-283.165 dB
5	**-33.4639 dB**
6	-295.362 dB
7	**-35.2253 dB**
8	-289.381 dB
9	**-37.3804 dB**

Table 2: Comparison of Simulation times

Abstraction	Simulation Time
Mixed Circuit and AHDL	36m 15s
AHDL only	3m 2s
Simple AHDL A/D model	1m 24s

```
NATURE electrical IS
  ACROSS V;
  THROUGH I;
END electrical;

ENTITY adc3 is
GENERIC(mismatch : REAL := 0;
        risetime : REAL := 5e-9;
        vref : REAL := 1.0;
        falltime : REAL := 5e-9;
        hi  : REAL := 5.0);
PORT(vin, d0, d1, d2 : electrical);
END adc3;

ARCHITECTURE basic_adc3 OF adc3 IS
  VARIABLE s2 : REAL := vref/2*(1.0 + mismatch*(random(2) - 0.5));
  VARIABLE s1 : REAL := vref/4*(1.0 + mismatch*(random(1) - 0.5));
  VARIABLE s0 : REAL := vref/8*(1.0 + mismatch*(random(0) - 0.5));
  VARIABLE out0, out1, out2, x : REAL;

BEGIN
RELATION BEGIN
    x := vin.V;
    out0 := 0;
    out1 := 0;
    out2 := 0;

    IF x > s2 THEN
       out2 := hi;
       x := x - s2;
    END if;
    IF x > s1 THEN
       out1 := hi;
       x := x - s1;
    END if;
    IF x > s0 THEN
       out0 := hi;
       x := x - s0;
    END if;

    d2.V <= transition( out2, 0, risetime, falltime );
    d1.V <= transition( out1, 0, risetime, falltime );
    d0.V <= transition( out0, 0, risetime, falltime );

END RELATION;
END basic_adc3;
```

Template 1: 3 bit A/D model with nonlinearity

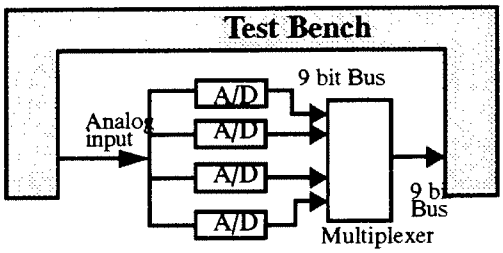


Figure 1: Parallel Pipeline A/D and Testbench

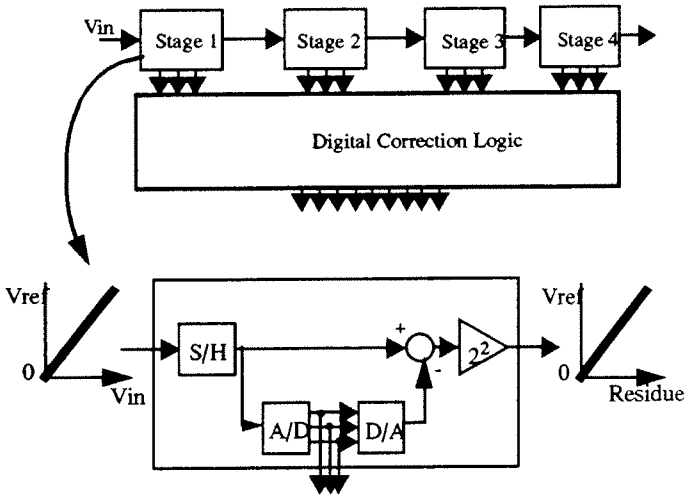


Figure 2: Pipeline A/D and Digital Correction Logic

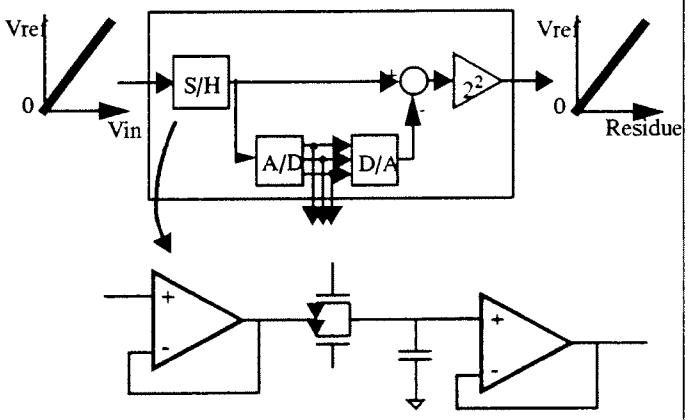


Figure 4: Mix of transistor level schematic and AHDL

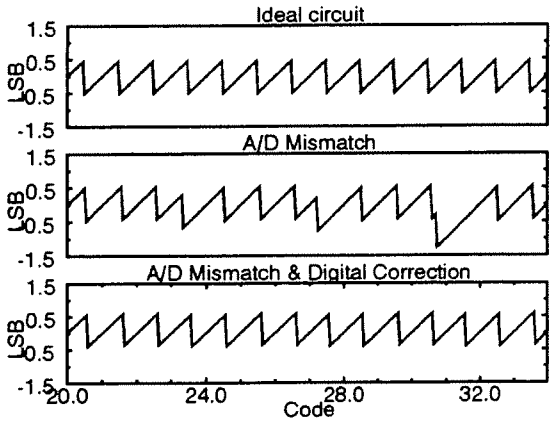


Figure 3: Simulation results with and without correction logic

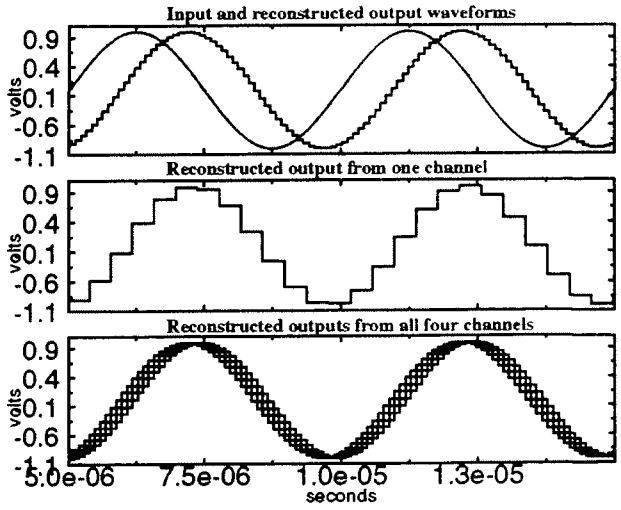


Figure 5: Transient Simulation results for mixed abstraction circuit model

VASIMS: A software package for validating and simulating FMSs modeled by Petri nets

C. AMER-YAHIA[1,2] Y. HADDAB[1]

[1] Université de Tizi-Ouzou, Institut d'Electronique, LCSP, B.P. 17 R.P. 15000 Tizi-Ouzou, Algeria
Fax number: 213 3 21 29 68

H. ACHOUR[1] N. ZERHOUNI[2]

[2] Ecole Nationale d'Ingénieurs de Belfort, LMP, 8, Bd Anatole France, 90016 Belfort Cedex, France
Fax number: 33 84 58 23 42

Abstract: VASIMS, a software package to assist the drawing , analysis, and simulation of flexible manufacturing systems is presented. The Petri net description, performed by the software user, follows a textual or a graphical approach. The system validation is based on the interactive use of the three well-known methods of analysis which allow behavioral and structural properties of Petri nets to be verified. A new algorithm which calculate all minimal invariants is used for the first time. All the programs constituting this computer-aided tool use the BORLAND C++ language. This object-oriented language makes the tool flexible and easy to use.

Key-Words: Petri net, FMS , Validation, Simulation, Object-oriented language.

1. Introduction:

Once a Petri net (PN) model of a physical system is constructed, a qualitative analysis may be performed. The analysis aims to investigate the behavior of the model and provides an assurance that the whole process is well defined. There are three main methods of analysis which allow behavioral and structural properties to be verified, for example liveness, boundedness, deadlock, and so on [1],[2].

VASIMS, a software package to assist the drawing, analysis, and simulation of flexible manufacturing systems (FMSs) is presented. The PN description, performed by the software user, follows a textual or a graphical approach. The system analysis is based on the interactive use of the three methods above-mentioned. The available processing allows the analysis of autonomous PNs (ordinary and generalized) as well as non-autonomous PNs (timed and synchronized).

All the programs constituting this computer-aided tool use the BORLAND C++ language. This object-oriented language makes the tool flexible and easy to use.

VASIMS is organized as it is shown in figure 1. It is constituted of two separate parts : the editor part and the processing part. A file serves as an interface between the two parts.

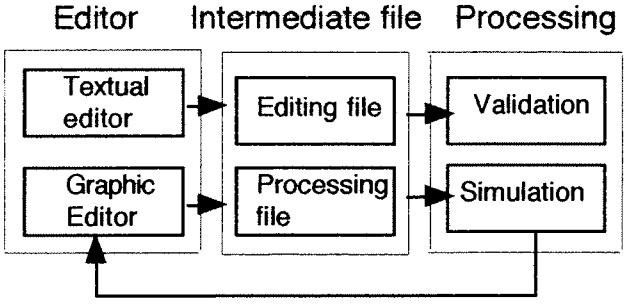


Fig.1 VASIMS'software structure

The resulting properties, such as liveness, boundedness,deadlock,safeness, reachability, persistence, and so on, when the different methods of analysis are applied to a manufacturing system model lead to the validation or the invalidation of the system. In the case where the manufacturing system model does not present the required properties, the proposed computer-aided tool makes it possible to find out and correct the design mistakes.

2. The PN editor

The implementation of a PN model on a computer uses two different approaches. It is possible to synthesize the PN model by means of a textual editor which allows to describe the PN structure and initialize the system model. It is, as well, possible to describe the manufacturing system, one hopes to design and control, by a graphic editor. However, because of the graphic nature of PNs, a particular care has been taken in the design of the graphic editor.

The net is drawn on a squared area which appears on the computer screen.

This area may occupy all the screen. When the menu or a sub-menu is needed, it is called and it appears on the screen. Once the needed function has been executed, the menu desappears, leaving the whole screen to the drawing operation.

A mouse may be used. It provides a flexibility in the access to PN components or menus.

In addition to the ordinary operations such as components displaying, deleting, names changing, and so on, the editor offers certain possibilities such as the zoom operation, the arcs drawing operation, etc... .

For a better PN drawing, an arcs drawing method is proposed. The method decides of the structure of the arc with regard to the relative positions of the place and the transition to join. The arc drawing is a part of an ellipse with the point $O(x_1, y_2)$ as a center (see figure 2).

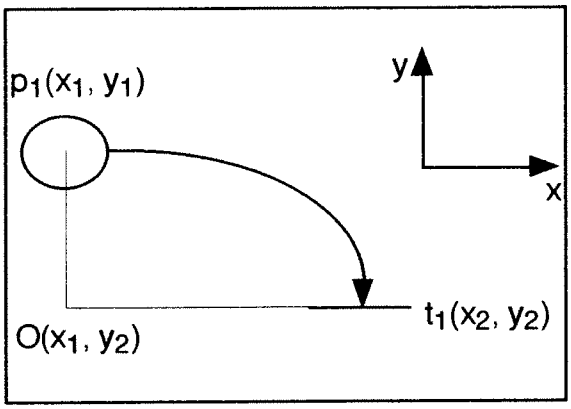


Fig. 2 Arcs drawing strategy

A PN modeling an FMS is constituted of a number of components that may represent a machine, a stock, or any entity of the system. In order to avoid confusions, references and names, displayed on the screen, are given to every place and every transition of the net. This way of doing seems to be very useful, particularly when a PN simulation is concerned.

Building large PNs is possible with VASIMS. Indeed, the authorized building space is not limited to the computer screen, but may be extended to a virtual screen whose dimensions, fixed by the user, are much larger than the real screen dimensions, and inside where the computer screen may move.

Because an object oriented language (the BORLAND C++) is used for writing VASIMS, the editor, from the conception point of view, is very different from editors that use ordinary languages [6],[7]. VASIMS proposes the following conception structure:

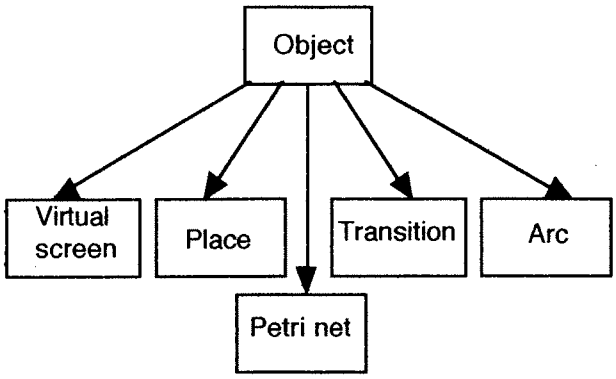


Fig.3 VASIMS'conception structure

The above structure is constituted of six classes. The object class contains all the data and methods that are common to the derived classes. All the classes inherit properties of the object class. The PN class contains all the pointers that are necessary to the manipulation of the net components. The virtual screen is a class which allows to manipulate the virtual screen with regard to the real computer screen.

The object-oriented language enables every place, every transition, and every arc of the net to be represented by an object. VASIMS introduces the idea of a PN by defining three sequences of objects, including all components of the net.

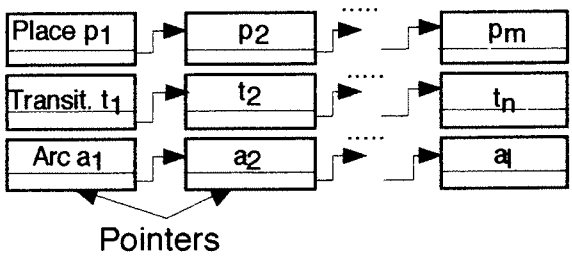


Fig.4Sequences of places, transitions, arcs

Each component of a sequence pointes on the following component by means of a pointer. Thus, the substraction or addition of components are widely facilitated. The object "PN" contains all the pointers that are necessary to the manipulation of the sequences. This way of doing presents a real advantage: the number of places,

transitions, and arcs is not limited. Thus, the size of a PN is only limited by the memory size of the user computer.

3. The intermediate file:

The PN drawing operation has led to the creation of two files: the editing file and the processing file.

The editing file receives all the data concerning the editing operation, such as the position of the different PN components on the virtual screen, the name of every place and every transition of the net, and all the PN drawing informations. It allows a repetitive call of any memorized PN in order to display it on the screen and, eventually, make some modifications. This file is used when a simulation of a PN is concerned.

The processing file is built with informations taken from the editing file. These informations are set in a way to be directly used by the processing algorithms, as it is shown in figure 5.

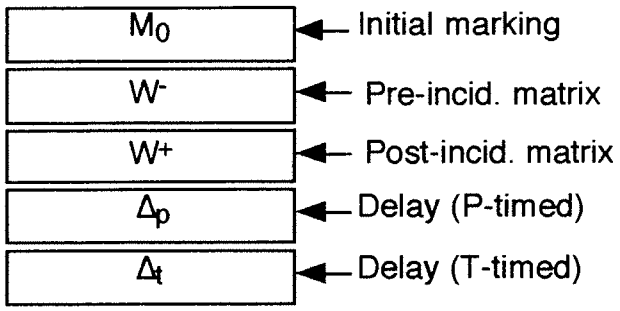


Fig.5 Processing file

4. Validation

This operation of analysis verifies the behavioral and structural properties of PNs by means of a number of tasks that are performed on the data contained in the processing file. the results of these processings validate or not the PN model.

VASIMS uses several methods for analysing PNs. Firstly, it makes it possible to draw the reachability tree (if the net is bounded) or the coverability tree (when the net is unbounded) of a PN according to its initial marking. This operation is done on the file in order to enable the analysis of large systems. Secondly, it allows to obtain all marking and firing invariants (P-invariants and T-invariants) which provide powerful tools for studying structural properties of a given PN independently of its initial marking. A new and powerful algorithm is used, in this case, for the first time [3]. Thirdly, the study of structural properties, as mentioned above, is done by means of a number of algorithms implemented in the proposed computer-aided tool. Finally, in order to facilitate the analysis of very large systems, VASIMS enables the reduction of the system model to a simpler one, while preserving the system properties to be analysed.

5. Simulation:

The computer-aided tool presented may perform, as well, the simulation of flexible manufacturing systems by enabling and firing transitions. The firing of transitions causes an evolution of the marking which corresponds to an evolution of the state of the system. Questions like the type and the number of machines one may buy, stocks capacities and system architecture one may use, and so on, are accurately answered by the simulation operation.

VASIMS allows the simulation of both synchronized and timed PNs.

In a synchronized PN [4], an external event is associated with every transition of the net. The firing of a transition will occur under the following conditions:

i) the transition is enabled,

ii) the associated event occurs.

The evolution of a synchronized system is described by the PN fundamental equation in which a new matrix is inserted, namely the events matrix E. The fundamental equation of PNs:

$$M_i = M_0 + W \cdot S$$

becomes:

$$M_i = M_0 + W \cdot E \cdot S$$

where M_0 and M_i represent respectively the initial and the actual markings, W is the incidence matrix, and S is the firing sequence. The event matrix E is an n x n matrix such that $E = \text{diag}(E_i)$, with $i = 1,2,...,n$, and where n represents the number of transitions of the synchronized PN. E_i ($i = 1,2,...,n$) represents the event corresponding to the transition t_i, with

i = 1,2,...,n. The simulation of externel events is done in the two following manners. When the number of transitions is small, a keyboard key is associated to every transition of the net. Pressing a key means the occurence of the event of the corresponding transition. When the number of transitions is large, the external event is simulated by pressing on the corresponding transition by means of the computer mouse.

In a timed PN [5], the functionning of a system is time dependent. The timing may be associated with the places (the PN is said to be P-timed), or with the transitions (the PN is said to be T-timed). Timed PNs are well suited for modeling systems in which a certain time may elapse between the start and the end of an operation. They are useful for evaluating the performances of a system. For example, the simulation of a P-timed PN is done in the following manner:

a) A time delay d_i is associated with each place p_i.

b) A counter C_i, with an initial zero value, is associated with each place p_i.

c) When a token is deposited in place pi, the following operations happen:

i) The token remains in this place for, at least, a time delay d_i. This token is said to be unavailable for this time.

ii) The time delay d_i is transmitted from the place pi to the counter C_i.

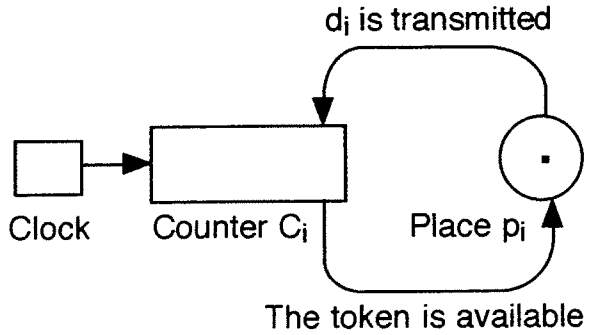


Fig. 6 Simulation of a P-timed PN

iii) The counter counts down at each clock pulse.

iv) When the d_i has elapsed the counter is reset to zero, a pulse is transmitted to the place pi in order to make the token available (see figure 6).

6. Conclusion:

We have presented a computer-aided tool which assist the drawing, analysis, and simulation of flexible manufacturing systems. The tool presents several interesting characteristics. The most notable characteristics are the introduction, for the first time, of the algorithm proposed in [3] which allows to obtain all minimal invariants of PNs, the utilization of an object-oriented language (the BORLAND C++) which makes the tool flexible and easy to use, and the possibility of drawing very large pNs because the authorized building space is not limted to the computer screen but may be extended to a virtual screen. Some improvements of this computer-aided tool are steel going on.

References:

[1] DAVID, R., and ALLA, H., 'Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems'. Prentice-Hall, London, 1992.

[2] MURATA, T., 'Petri nets: properties, analysis and applications'. Proc. IEEE, Vol. 77, 1989, pp. 541-580.

[3] AMER-YAHIA, C., and ZERHOUNI, N., 'A mathematical method for calculating minimal invariants in Petri nets'. (In preparation).

[4] MOALLA,M.,PULOU, J.,and SIFAKIS,J., 'Réseaux de Petri synchronisés'. Journ. RAIRO, Automatic Control Series, Vol. 12, 1978, pp. 103-130.

[5] RAMCHANDANI, C., 'Analysis of Asynchronous Concurrent Systems by timed Petri Nrets'. PHD Thesis, MIT, Cambridge, MA, 1973.

[6] LE MER,E.,'OVIDE: A software package for verifying and validating Petri nets', IFAC Software for Computer Control, Madrid, Spain, 1982, pp. 255-260.

[7] COLOM, J.M., MARTINEZ,J., and SILVA, M. 'Packages for Validating Discrete Production Systems Modeled with Petri nets'. In Applied Modelling and Simulation of Technological Systems, Elsevier Science Pub. B.V., 1987, pp. 529-536.

Simulation-Aided Investment Analysis for Manufacturing

Florian Klug
Institute for Operations Research and Systems Theory
University of Passau
Innstraße 33, 94032 Passau, Germany
E-Mail: klug@moni.fmi.uni-passau.de

INTRODUCTION

During the planning phase of a manufacturing plant, a main objective of the designer of the system is choosing between different design options. This implies working out a cost function which minimises the overall cost of the planned manufacturing system over its lifetime at a given, expected performance level. Due to the long timescales required for large technical projects the resulting decision problem can be extremely complex in its dynamic behaviour and underlying uncertainties.

Simulation is one of the decision support technologies for the planning phase of such systems. It gives the designer a tool for validation of technical and logistic options at an early point in the project, which reduces risk and - by enabling early recognition of errors - shortens the planning cycles.

Traditionally, the values for the relevant technical variables - obtained by simulation runs - have been used for decision support. This approach, however, leads to inefficiencies due to cost factors being neglected in favour of technical factors. While the method is usually sufficient to ensure that the chosen design will satisfy all the technical restrictions and boundary conditions, the final choice of design (among several sufficient ones) is often down to intuition. If we want to optimize this final selection then we need to introduce a cost-related target.

SIMULATION-BASED COSTING

To support such cost-based target functions a costing system has been developed, which is suitable for simulation and which consequently, could be integrated in the existing simulator for manufacturing systems, OSIRIS.

The new costing approach is required to support assessment of the design alternatives. Furthermore, it needs to allow integration of features which are specific to a particular simulation model, a requirement which made traditional costing system inapplicable.

Traditional costing methods which were developed with different requirements in mind tend not to meet the requirements of simulation based costing and must therefore, be excluded from the list of candidate costing methods. Some, more recent approaches to costing attempt to achieve a more realistic mapping from the real world to the costing model and therefore, need to be investigated as to whether they can be integrated into a simulator. The following, four methods have shown initial promise:

- Riebel's unit cost calculation
- machine hour accounting
- logistics costing
- activity based costing

Since each of the methods only fulfills some of the requirements for a fully operational cost simulator we need to be selective and re-combine as well as modify elements from each of the approaches in order to arrive at a costing system which can be used for simulation. This new approach was developed application-independent, and was then integrated into the OSIRIS manufacturing simulator, for verification. There are some general requirements for simulation-oriented costing methods - we included the following in our implementation:

- a realistic, cost-based representation of the manufacturing process
- user friendly input mechanisms for cost parameters
- re-use of existing accounting data in order to avoid duplication - acquisition of cost data at source
- cost attribution according to Riebel's relevance criteria.

The following applications are possible:

a) calculation of unit cost for a segment of the manufacturing plant
b) economy studies for single components as well as for the manufacturing plant as a whole are enabled by input/outputanalyses.
c) sensitivity analysis of cost drivers through simulation leads to better planning solutions.
d) cost based control of the manufacturing process is enabled. Up to now, control strategies for flow of materials has been based primarily, on quantity-related or time related parameters (e.g. load factor, throughput-time). In order to improve the quality of micro-economic decision making, these need to be transformed into cost factors.
e) analyses of overall value enhancements, based on individual value enhancements by cost factors or groups of cost factors
f) simulation support for logistics-based costing

SIMULATION-AIDED INVESTMENT

In economics, the evaluation - and assessment of comparability - of different options is regarded as a classical investment problem. Investment theory approaches decision problems by looking at cash flows between input and output categories. Charting the cash flow will not only show the current financial situation of a company and its solvency over time, but will also give indications about the profitability potential of alternative investment plans.

Practitioners who were forced to deviate from the rigorous notions of accounting as used by investment theorists, paved the way for the use of cost-oriented parameters in investment analysis. Cost-Volume-Profit models (CVP) have cost-theoretic foundations whereas techniques such as net present value (NPV), payback-method, accounting rate of return, or internal rate of return (IRR), as well as some modern Operations Research models are based on the theory of capital budgeting.

We believe that it is possible to use cost-oriented investment analysis (which is not based on the theory of capital budgeting) for solving known types of investment problems. Historically, these cost-oriented procedures were developed for purely practical reasons: it tends to be very difficult for companies to conduct sufficient data acquisition for assessing future input and output. This leads decision makers to turn to existing cost data in order to support their decisions.

One might feel that because of its cost-theoretic foundations, cost-integrated simulation is just another kind of cost analysis technique. However, traditional CVP techniques are static methods

where observations are made at a particular point in time and where there is a very limited time horizon. They are therefore often termed "subgoal-oriented techniques". Cost-integrated simulation by contrast, simulates the life-cycle of the plant which is to be built. We therefore propose that the taxonomy extended by a new class of "dynamic investment calculations". The great advantage of this method is its practical usefulness while at the same time, overcoming virtually all of the shortcomings associated with static techniques.

Having established that simulation based costing can be used for investment analysis we investigated in what respects results would deviate from those of classical input/output analysis if investment simulation was used.

A theorem by Lücke shows that one can indeed use cost terms as the basis of an investment analysis technique and that, if certain preconditions are satisfied, there will be no deviations as long as relevant costs of project capital are considered over the time period in question. This also proves that we were justified in accounting for these costs in our cost analysis model which was developed prior to the work reported here.

REFERENCES

Cooper, R.: Activity-Based Costing - Wann brauche ich ein Activity-Based Cost-System und welche Kostentreiber sind notwendig?, in: *Kostenrechnungspraxis* (5/90), p.271-279

Gausemeier, J.; Erfolgspotentiale integrierter Ingenieursysteme (CAE), in: Gausemeier, J.(Hrsg.); *Proceedings CAD '94*; Paderborn 1994, p.1-25

Gutenberg, E.: *Grundlagen der Betriebswirtschaftslehre, Band I: Die Produktion*, 23. Auflage, Berlin, Heidelberg, New York 1979

Hartberger, H.: *Wissensbasierte Simulation komplexer Produktionssystem*, München 1990

Hummel, S.: Die Forderung nach entscheidungsrelevanten Kosteninformationen, in: Männel, W. (Hrsg.): *Handbuch Kostenrechnung*; Wiesbaden; Gabler 1992, p.76-83

Kaplan, R.S.: Ein einziges Kostensystem ist zuwenig, in: *HARVARDmanager* 3/1988, p.98-104

Klug,F.: Kostenintegrierte Fertigungssimulation, in: *Simulation in Passau*, 1994 Heft 1

Klug, F./ Fortmann, Ch.: Do Simulation Models In Manufacturing Make Any Sense Without Cost Analysis, in: *CISS - First Joint Conference of International Simulation Societies Proceedings*, hrsg. von Halin, J./ Karplus, W./ Rimane, R., Zürich 1994, S.415-419

Klug, F./ Fortmann, Ch.: Kostenintegrierte Simulation als betriebswirtschafltiches Bewertungsverfahren fertigungswirtschaftlicher Systeme in: *Simulationstechnik 9. Symposium in Stuttgart Oktober 1994* hrsg. von: Kampe, G./ Zeitz M., Braunschweig/Wiesbaden 1994, S.579-584

Plinke, W.: *Industrielle Kostenrechnung für Ingenieure*; Berlin, Heidelberg, New York; Springer Verlag 1989

Riebel, P.: *Einzelkosten- und Deckungsbeitragsrechnung; Wiesbaden*; Gabler 1990

Schneider, A.: Prozeßorientiertes Controlling und Rechnungswesen auf der Basis der I-Linie; in: Reichmann, Th. in Zusammenarbeit mit Sema-Group Paris(Hrsg.): *DV-gestütztes Unternehmens-Controlling*; München Vahlen 1993, p.24

Spur, G.: Simulation zur Auslegung und Optimierung von Produktionssystemen in: *Zeitschrift für wirtschaftliche Fertigung* 77 (1982) Nr.9, p.446-452

Witte, T.: *Simulation von Produktionssystemen mit SLAM*; Bonn, Paris, Reading; Addison-Wesley 1994

A Scalable 3D Animator with Open Interfaces

Heiko Kirchner, CePLuS GmbH Magdeburg
Ralf Helbing, Universität Otto von Guericke, Magdeburg

Introduction

The animation system AniPLuS was originally developed by the CePLuS GmbH Magdeburg. It is intended as a post-run animation tool for three-dimensional modeling and visualization of dynamic processes.

Animation is often used to provide a visual understanding of an abstract model that is used in a simulation system.

Today's simulation software allows the creation of very sophisticated models of real systems. However, this can only be done using abstraction techniques to formulate the reality in terms the simulator can understand.

Based on the same simulation model, several animations can be created, each of them tailored to it's specific purpose.

Since three-dimensional animation becomes more and more realistic, a user can intuitively apprehend the simulation and imagine how the real process works. This is especially useful in a presentation where some of the audience may be unfamiliar with either the simulation or the real system.

Furthermore, the animation can help the simulation process itself by revealing possible flaws in the simulation model.

Of course, creating the animation means additional effort, but this can easily be offset by the benefits mentioned above.

AniPLuS

Components

Our animation system consists of four major parts:

- the kernel which manages and processes all animation data and contains an interactive 3D modeler for the animation model and an interface to import geometry descriptions for the objects and their layout
- a rule editor to transform the simulation run into a series of movements and other visible alterations of objects
- an underlying graphics system to render the animation frames in rapid succession
- a graphical user interface

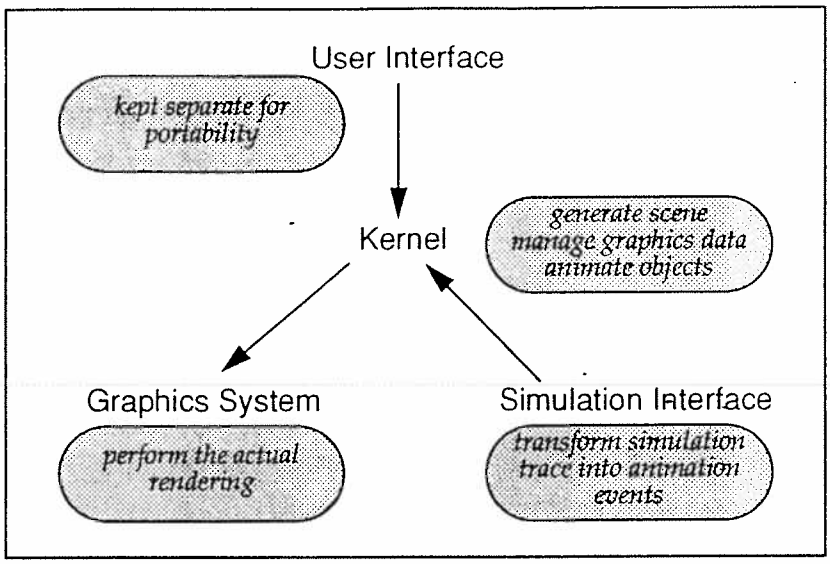


Fig 1: AniPLuS architecture

AniPLuS is available on several types of UNIX workstations, including SGI, SUN, HP, and IBM as well as the popular PC based LINUX system. If available in a fast (hardware accelerated) implementation, the OpenGL graphics interface will be used. Otherwise a very simple PHIGS-like implementation based on generic X11 is the graphics system. Although the OpenGL version is expected to be faster especially for models with a large number of small polygons, all[1] rendering functions are available in both versions. Providing two versions with essentially the same feature set allows to meet user-specific requirements for rendering quality and speed while still having a cost effective solution.

Ports to other systems like OS/2 are planned.

Creating Animations

In order to create animations, the animation system must solve at least three problems:

- import of geometry data
- transformation of simulation results into visible animation events
- running the animation

Thus, an animation consists of a static scene describing the initial configuration of the animation model and an ordered list of animation events.

The scene contains all the geometric information needed to display the objects. There are two kinds of geometry information: layout data and shape data.

[1] with the exception of things like gouraud shading and antialiased outlines

Layout information is stored in configuration files and defines the position, orientation, relative size, color, name and reference ID of each object. Shape information is defined externally in a DXF file. An object can have more than one shape. This is useful to indicate different conditions of an object in a more interesting fashion than only with colors, e.g. a transportation vehicle with or without load.

In order to create a scene, the user must:

- build a DXF description for each class of objects using a geometric modeler of some kind
- import these descriptions into AniPLuS
- create and place/edit instances of objects

Steps 2 and 3 are performed in the integrated scene editor. Once an object is placed in the scene, it can be selected and edited with respect to its position, orientation, size and color. The scene can then be saved for later use with an animation.

The list of animation events is what brings the scene to life. Each event can be thought of as a transition that affects an object in a certain aspect. At present, there are seven kinds of events: move, rotate, scale, color, shape, hide and show. The first three events can be arbitrarily long in duration whereas the rest of the list occurs immediately at the given time. Several data entries are common to all events: type, start, length and the targets object-ID. The rest of the event information varies with the type. Move/rotate/scale events have their respective final state as parameter. Color and shape events need the name of the new color. Hide and show events need no further information.

Interfacing the Simulator

As we visualize the behaviour of simulated objects, the creation of the animation model starts with the simulation. Currently, the focus is on discrete event simulators. There are five steps to build an animation from a given simulation model:

- identify which objects can be and must be visible
- build a scene with these objects using the scene editor
- set up a number of rules that transform simulation events and actions into animation events using the rule editor
- use these rules to automatically perform the transformation

Here step three is the most interesting one. It allows to create different animations based on one simulation run. Each animation can be focused on specific parts of the model and neglect less important issues.

Usually, each class of simulation objects (from the first step) will have a visible representation. Objects from that class can then be instanced in the animator, that is made visible at certain places in the layout. Simulation events implying the move of a specific object from one place to another will have a corresponding rule to rephrase this action in terms of animation events mentioned earlier. There can be a series of animation events generated from a single simulation event (an object moving on a path would have to perform a number of moves and perhaps turns to reach its target).

All these animation events can either be written to a file for later execution or be performed on-line.

Future work

AniPLuS is still in an early phase of its development and many improvements must be added to make it a stable and useful tool.

Fields of interest are:

- providing various levels of detail (adequate for the current hardware and desired frame rate) for the creation of high-quality animation with detailed objects vs. simple animations for previews and low-end hardware
- addition of an interface to the element-oriented simulator Create!
- a clean, easy-to-use GUI
- an OS/2 version of AniPLuS
- accelerated movements, text output and textures

Current work is focusing on two fields: the Proof™ interface and the use of cinematic knowlegde in the creation of asthetic animations.

The Proof™ interface will allow the use of many existing 2D Proof™ animations, layouts and objects in a 3D world. However, it requires a cleanly defined general simulation interface which can be used for other simulators, too.

Film techniques as used in movies would help the viewer better understand the simulation, since the language of film is intuitively known by almost everybody. Furthermore, their use allows the creator of an animation a much better control on how to fulfill his communicative goals and therefore provide an effective animation in terms of time spent by the viewer to „get the message".

Several models from different application fields with varying levels of detail and size have been built.

Current research about the integration of simulation and animation continues at the University of Magdeburg.

The Next Generation of Simulation Tools: A Focus on Usability in Micro Saint

by K. Ronald Laughery and
Catherine E. Drury
Micro Analysis and Design, Inc.
Boulder, Colorado USA

Simultaneously, the world of discrete event simulation software is both mature and, yet, in its infancy. It is mature in that the necessary modeling power of simulation tools is largely defined and supported by existing commercial packages. Discrete event models of virtually unlimited size and complexity can now be developed with commercial software packages on computers that sit on virtually every engineer's desk. However, it is in its infancy in that the number of *actual* users of discrete event simulation is a small fraction of the number of *potential* users. If one simply looks at the number of examples of discrete event simulation studies performed by the engineering community, it is obvious that relatively few engineers are aware of the potential for discrete event simulation, and far fewer are frequent users. Furthermore, discrete event simulation has wide potential use outside of the engineering community. For example, the management sciences could use simulation in many aspects of business process design and development. Yet, simulation is almost unknown to most managers.

One could simply attribute this lack of awareness and application of discrete event simulation to many potential factors that are beyond our control including:

- *It takes time to raise awareness and build a market for new software technologies.* However, if one considers what happened when computer spreadsheet software was introduced twenty years ago, it is obvious that users will quickly accept new technologies if the technology can be used to solve a wide variety of problems.

- *Simulation is inherently too complicated.* Certainly, one can create examples of system issues that require highly complex model designs. However, there are many straightforward problems that lend themselves to simple formulations with simulation. By and large, these straightforward problems are still not being addressed with simulation.

- *The simulation software market has not had sufficient time to develop mature tools.* Computers were invented to do simulation.[1] There has certainly been time.

Certainly, the above factors have contributed to slow growth in the use of simulation. However, we suggest that *the primary factors slowing the use of simulation is the same today is it was twenty years ago - most software still demands too much of the user in the way of computer programming skills*. While many commercial software packages have provided a more usable environment for specific modeling problems (e.g., robots), once the user must represent something out of the ordinary, some form of programming is usually required.

The last few years have seen an increase in the availability of graphically-based model building tools such as Micro Saint™, ProMod™, and Arena™. Also, more sophisticated windowing environments are making the models more transparent to the developers as they build and run their simulations - another key to enhanced usability. However, there are still many aspects of discrete event simulation packages that could be substantially improved to enhance usability.

From the outset, our mission in the development of Micro Saint has been to enhance usability without sacrificing modeling power. Many innovations over the past ten years have resulted in leaps forward in usability. However, we still struggle to develop the product and market education strategies that will lead to

[1] ENIAC, the computer that is widely perceived to be the first electronic digital computer, was developed in the 1950s at the University of Pennsylvania to simulate ballistics trajectories.

the mass market for simulation use. What are the factors that have kept the mass market from our doorstep? The remainder of this paper presents the three factors that we see limiting the growth in the market and some of the initiatives we are undertaking to address these factors at Micro Analysis and Design.

Limiting Factor #1 - The wide variety of skills of the potential user base - Simulation does require that the user have significant analytical skills. Even if we were able fully automate the process of translating a process flowchart into a functioning simulation, analytical skills would still be required to determine how best to chart the process and the appropriate level of detail for the problem at hand. Users must understand the basic concepts of systems analysis and measurement and many potential users do not. To some degree, the problem lies with the schools and universities. While we teach children the methods and issues when adding columns of numbers, as in a spreadsheet, we do not teach the methods and issues of system function decomposition. Accordingly, when new users of our products show up for training, we have a wide diversity of systems analytic skills. For example, in a recent Micro Saint training class, we had individuals with advanced degrees in engineering and computer sciences as well as individuals with no prior programming experience. This was a typical class.

The wide variety of skills in the potential user base creates a conflict to the developer and designer of simulation software - do we create software for the systems analyst or do we create software for the broader market which demands less analytical skills? Often, software elements that facilitate use by the novice simulationist will impede the experienced analyst. We propose two solutions to this conflict that are being used increasingly by modern software; layering and wizards.

Layering - Complex computer software today is often designed to provide many layers of usability and, hence, complexity to the user. For example the novice user of a word processor can create letters and other basic documents without learning how to format and write macros. Simulation software can do the same. For basic systems analysis problems with straightforward process flow and resource constraints, the software can provide usable tools. For complex systems, the user could have access to more aspects of the model of the systems behavior, such as complex decision behavior and constrained resource modeling.

An example of the concept of layering is presented in figures 1 and 2. Figure 1 presents a task description in Micro Saint where a complex set of resource requirements, as defined in the release condition. Figure 2 presents a new feature we are developing which will allow the user to define simple resource requirements associated with tasks.

Wizards - For the past several years, MicroSoft™ has been incorporating wizards into their software products. By invoking a wizard, the user is presented with a series of questions that helps him to use a feature of the software (e.g., set up a form for data entry into a database). Experienced users need not use wizards to use the features.

Figure 1. Defining complex resource requirements in Micro Saint

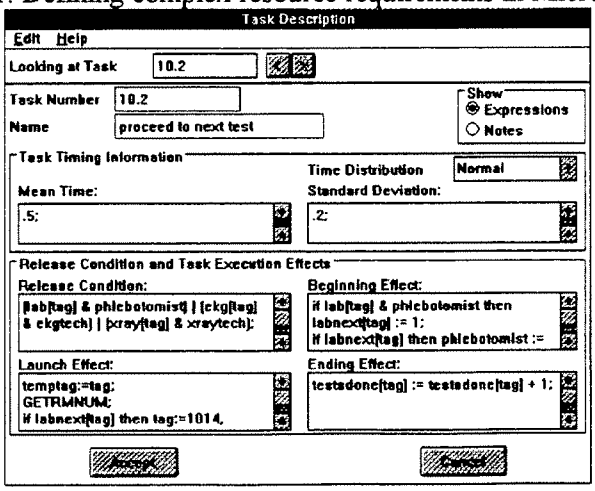
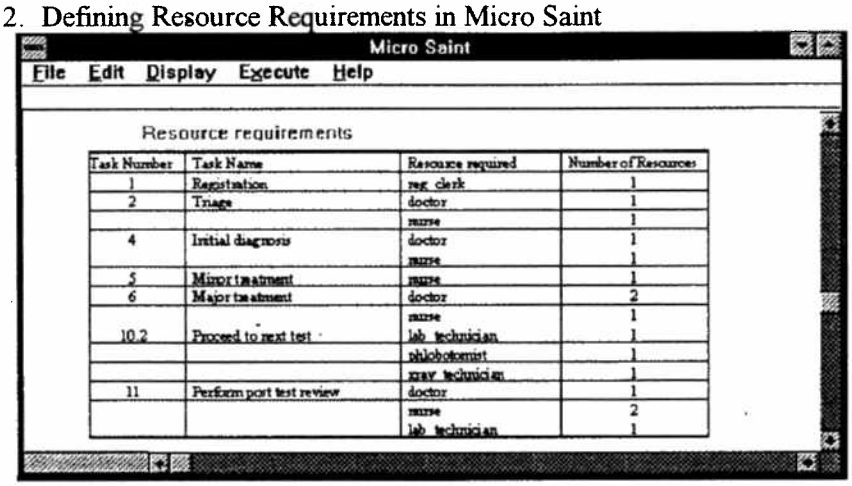
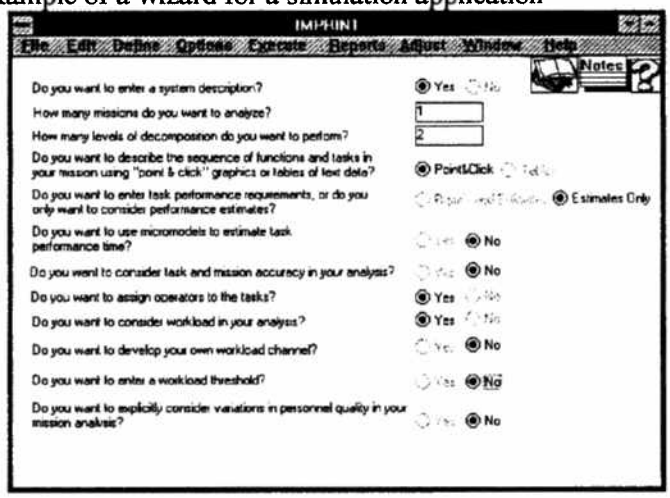


Figure 2. Defining Resource Requirements in Micro Saint



Many aspects of model development lend themselves to the use of this type of technology. These are, in a sense, expert systems that can assist users in a range of activities - from determining the appropriate level of detail for the model to building custom model data collection. While we have not incorporated wizards into our commercial software, we have developed wizards for several custom simulation packages. An example of a wizard user interface is presented in Figure 3.

Figure 3. An example of a wizard for a simulation application



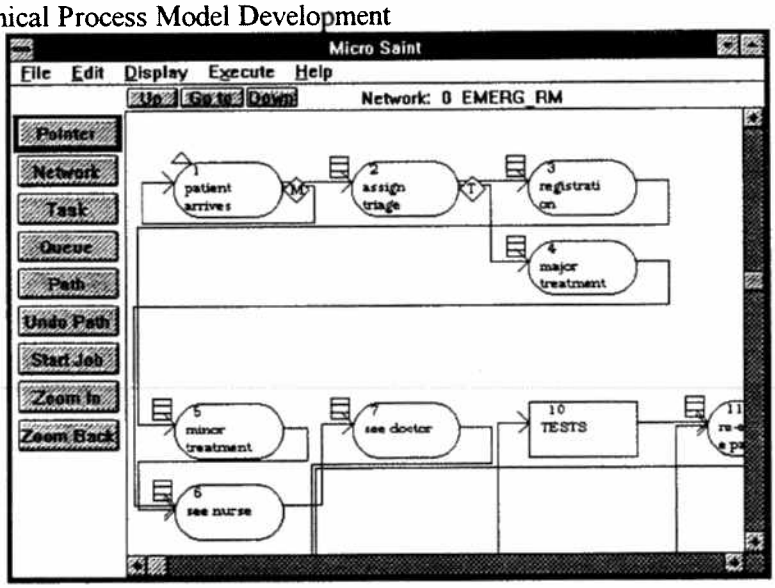
Limiting factor #2 - Dramatic differences between simulation tools, both structurally and in look and feel - A common problem we hear from potential users of simulation tools as they review the suite of tools that are available is that the tools all look so different. This leads to confusion on the part of the potential customer and makes it difficult to make a purchase decision. Potential customers like to think that they are comparing "apples to apples," which in today's market is not always easy to tell. These differences also limit the degree to which experienced users are able to change simulation systems. This is in contrast to, for example, spreadsheet software where all products are quite similar.

Why is the difference between simulation tools so dramatic? We suggest that it is the long history of discrete event simulation that has created these differences. Since the early 1960s, models have been built using higher level languages. A set of the simulation tools evolved using this language-based approach. In the mid 1980s, a set of tools emerged using modern software techniques such as language parsers and graphical development environments. These approaches are inherently different .

Time and the marketplace will allow the best approaches to emerge as standards. This is beginning to happen now. For example, many software packages provide a graphical process model development environment as exemplified in Figure 4 as the foundation for model design. As those of us building and

selling simulation software listen to the features that the marketplace likes and does not like, other aspects of different simulation tools will look increasingly alike.

Figure 4. Graphical Process Model Development



In the future, there will be forces beyond the marketplace that will encourage increased commonality. A Simulation Software Vendors Group now exists to direct and promote simulation software technology. This will result in some standardization of software functionality and look and feel.

Limiting Factor #3 - Price - Simulation software is still quite expensive in comparison to software that is targeted at similar markets (e.g., computer aided design software). This is part of a "chicken and egg" problem. Until there is a larger market, the fixed costs of developing the simulation software must be paid by a smaller number of users, thereby keeping costs high. However, the high costs are keeping the market small.

To expand the market, some software vendors, including ourselves, are promoting the use of simulation through the Universities. By providing simulation software at a low cost to the emerging user, the student, we hope to expand the base of users in the long run, thereby allowing us to lower our price. Additionally, the market must be sensitive to price and the vendors must continue to hear from potential buyers that price is a significant factor in deciding which software package to buy.

Summary - The wide variety of skills of simulation users, the differences in structure and look and feel of the software and the vast price differential are all areas that are driving our product development process. Our long-term goal is to increase the base of simulation software users by enhancing our products' usability, working with Universities to create a new market and eventually lowering the price to make simulation software a product that is on everyone's computer. In order to achieve this goal, we will continue to solicit ideas from the simulation community on new methods and constructs that will make simulation more intuitive to the experienced user as well as the vast number of users that we have not yet reached. Please contact us via email at sales@madboulder.com to comment.

X'': A Vision for Worldwide Simulation

Andy Symons, X-Prime Ltd.
Orteliusstraat 176-1, 1057 BK AMSTERDAM, NL
tel :+31 20 689 2022, fax: +31 20 6168142, email: andysymo@xs4all.nl

Keywords: modelling, simulation, client-server, object-orientation, internet, networks

Abstract: *a client-server architecture is proposed for the cooperative establishment of powerful, interactive, user-friendly modelling and simulation facilities in a worldwide network environment.*

Introduction and overview

X' (read "X prime") is an experimental system which has been under development since 1991 [1]. It has proved some important basic assertions about interactive modelling and simulation:

- that models can be captured directly *in user terms* (such as mathematical equations)
- that a compute-ready internal model can be *generated directly* from such a user model without use of a programming language
- that this generation (compilation) can be done *incrementally,* i.e. as the model is entered
- that experimentation on a model can be separated from the modelling process itself
- that the whole modelling and simulation process can be usefully controlled by an *object-oriented* graphical user interface (OO-GUI)

Further, it has been shown theoretically [2] that the automatic generation (compilation) of models can easily be extended to *parallel* computer architectures

However, as a development group of 4 people part-time, we had a big problem: we had too much work to make useful progress... so we did the only thing possible: we made the system concept even bigger!

The new concept described in this paper (dubbed, X'' or "X prime prime") is based on the idea of interacting clients and servers operating locally or over networks such as the Internet. The great advantages of this approach are:

a) separate clients and servers (of various hue) can be developed concurrently and independently by separate organisations
b) the potentially usable computing power for simulation is unlimited: any resource, connected by a network, can be employed

This paper is an introduction to the architecture and an invitation to universities and similar institutions to participate in the X'' programme by developing your own contributions in the form of client and server software. You may for example like to develope a better model editor client with direct translation of mathematical formulae, or a graphical input method such as bond graphs or process diagrams. You may like to implement a powerful compute server based on a parallel computing platform connected to the Internet. The *quid pro quo* is the possibility to combine this with our software or that from other institutions in the programme.

This is no unambitious plan, but it is eminently practical. We have already solved the 'difficult' problems. Everything required by the X'' specifications is based on technology that is either already available elsewhere or has been proven by us in the first X' project. The novelty lies in the new *combination* of the technologies.

Client-server principles

The basic idea of a client-server architecture is that the programs concerned with interfacing to the user (the *clients*) are separated from the programs concerned with the organisation of bulk data and, in our case, also the powerful computations required (the *servers*). In principle, clients and servers can be implemented on any platform. Usually (and easiest to visualise for X''), clients will be implemented on PC's or workstations with a windows-oriented graphical user interface (GUI); examples are a PC with MS-Windows, an Apple Macintosh or a UNIX workstation such as a SUN with X-Windows. Servers will usually be implemented on bigger machines such as UNIX mini-computers or mainframes, although small implementations will be made for test purposes and for use with small models, e.g. for teaching. Clients and servers are connected by some kind of network such as the Internet.

Those wishing to know more about client-server architectures are referred to the abundant literature on the subject.

So how can client-server ideas be applied to modelling and simulation? ... (see figure 1)

The X'' Clients

We identify three types of X'' client:

a) *Model Editor* clients
b) *Experiment Control* clients
c) *Results Analysis* clients

A *Model Editor* client edits *Model Components*. Model Components (synonyms: 'model objects', 'sub-models' or 'model block') are descriptions of items in the real world. Instances of Model Components can be invoked in other Model Components and can be characterised by (not necessarily constant) *parameters*. Thus an hierarchy is formed which we call a *Model*. Note that the Model is kept quite separate from numerical solution methods, which in X'' are part of an *Experiment* (see below). New Model Components can be *inherited* from other Model Components with full function overloading; therefore, the X'' Model Description Language can be said to be truly *object-oriented*.

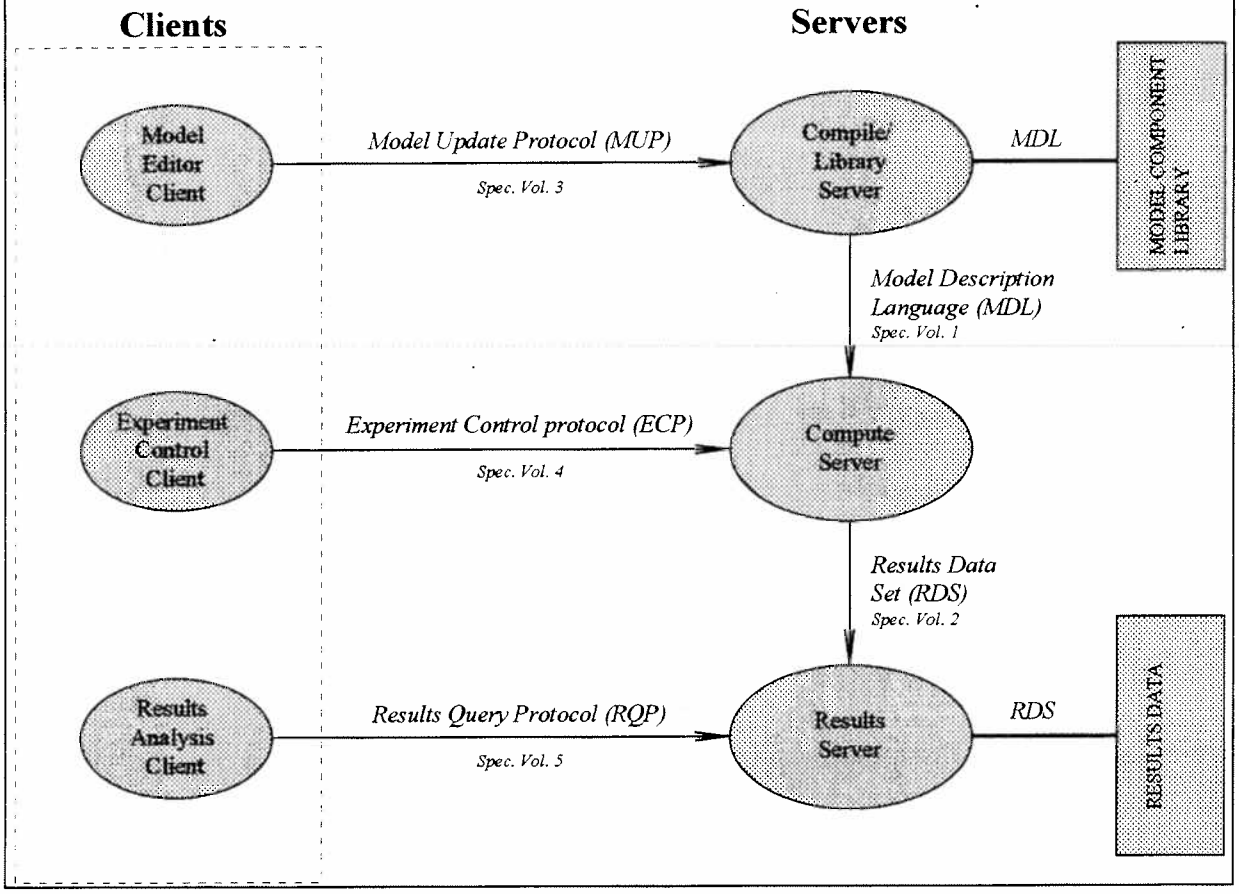
An *Experiment Control* client interacts with the user to ascertain what the user wants to *do* with the Model. This includes setting initial values, specifying simulation begin and end times, manipulating operator parameters and, if required, influencing the numerical methods which will be employed to calculate the results. The results can also be displayed as they are generated, if required. Running simulations can be 'paused', 'rewound', 'forward wound', influenced by operator parameters and restarted.

A *Results Analyzer* client gathers the results of one or more experiments and represents them in a variety of graphical forms. Results of experiments on one or a *number* of different models (including different versions of the same basic model) may be combined in one graph. Graphs can be displayed, printed or exported to other documents in, for example, a proprietary word processor.

The X'' Servers

Servers are used to store data and do computations 'behind the scenes'. They only interact with users via clients. They will ultimately be implemented on middle range or large machines, including massively parallel configurations.

Figure 1: X" Client-Server Architecture



We identify three types of server:
a) *Model Compile/Library* servers
b) *Compute* servers
c) *Results* Servers

A *Model Compile/Library* server converts the (external) model description entered by the user via a Model Editor client into an internal form suitable for (parallel) computing. It also keeps information on the model topology such as where variables are defined, the class and type of each variable and so on. A Model Compile server can act in an interactive mode so that a user is continually warned of syntactical and semantic errors and is continuously informed of the completeness of the model as it is built. Model Components can be saved and retrieved in a library. For this, existing file transfer protocol (FTP) technology can be employed. Model components include a text description to 'advertise' them to other users. These descriptions could be used to form search (e.g. WAIS) databases.

A *Compute Server* receives an experiment specification, which refers to a particular model[1]. The Model is already compiled as far as is possible without knowing the target compute platform. An *Assembly* step maps this internal compiled model onto the machine code and architecture of the target platform, then the calculations are done according to the experiment specification, using suitable numerical

[1] References could, for example, be based on the Universal Reference Language (URL), so the model components can be anywhere on the Internet).

methods. Several experiments can be carried out on the same model. The nature of X'' Model Descriptions is such that the computations can be carried out on massively parallel machines without any intervention from the modeller. The results are saved using the X-Prime Results Data Set (RDS) standard.

A *Results Query* server provides access (for a Results Analysis client) to a potentially vast wealth of results calculated from any number of experiments on any models. Preliminary location, extraction and sorting functions are provided, which can be refined further by the client.

The X" specifications

The key, of course, to a successful global client-server environment is a common definition of the data. Data in this case means the formats of the files used to save models and results, and the protocols used to transfer instructions and data between the clients and the servers. The principle is that all parties can develope software independently with the assurance that they can use each other's products, provided they adhere to the interfacing and data standards. To this end X-Prime is preparing the following specifications[2]:

Vol. 0: Introduction and Overview. The Architectural relationship between all the other documents. Read this first!

Vol. 1: Model Description Language (MDL). The detailed description of the internal file format for a model component.

Vol. 2: Results Data Set (RDS). The detailed description of the internal file format for the calculated results data.

Vol. 3: Model Update Protocol (MUP). The protocol used between a Model Editor Client and a Model Compile and Library server.

Vol. 4: Experiment Control Protocol (ECP). The protocol used between an Experiment Control Client and a Compute Server.

Vol. 5: Results Query Protocol (RQP).
The protocol used by a Results Analysis client to query the results database.

References

[1] "X-Prime: An Interactive Simulation Package". J.A. Burton, M.H.H. van Dijk, A. Symons.
Published in "Computational Systems Analysis 1992" - Proceedings of the 4th International Symposium on Systems Analysis and Simulation, August 25-28, 1992, Berlin, Germany. Elsevier Science Publishers B.V., P.O. Box 211, 1000 AE AMSTERDAM, The Netherlands. Editor: A. Sydow. ISBN 0 444 89780 1, pages 457-462.

[2] "What's the point in Parallelisation?". Author: A. Symons.
Published in "Massively Parallel Processing Applications and Development" - Proceedings of the 1994 EUROSIM Conference on Massively Parallel Processing Applications and Development, Delft, The Netherlands, 21-23 June 1994. Elsevier Science Publishers B.V., P.O. Box 211, 1000 AE AMSTERDAM, The Netherlands
Editors: L. Dekker, W. Smit, J.C. Zuidervaart. ISBN 0 444 81784 0, pages 321-331.

[END OF DOCUMENT]

[2] Documents can be retrieved from the X-Prime WWW homepage on *http://www.xs4all.nl/~andysymo/x-prime*

SIMULATION USING SITA

G.Jonin, J.Sedols, Dz.Tomsons
Institute of Mathematics and Computer Science, University of Latvia
Rainis Boulvd. 29, LV-1459 Riga, Latvia
Phone: +371-2-227815 ; Fax: +371-7820153 ; E-mail: jonin@mii.lu.lv

ABSTRACT

Simulation of any real system aims to some destination. If a new system is under development, the simulation is used as a tool for the investigation how the system will act under certain conditions. For existing systems, the simulation is applied in case when something should be changed. A great amount of simulation tools, like GPSS, ACSL, SIMULA, SITA, etc., are built in the world [1]. Any of them has certain advantages for the use to some of research directions - system design and specification, prototyping, testing, calculation of characteristics, data processing and analysis, hypothesis testing, forecasting, training, etc. The features and advantages are described that can be got using simulation system SITA in order to solve the above mentioned problems. The research branches, where SITA has been used, are illustrated with corresponding examples.

1. SYSTEM SPECIFICATION

SITA (Simulation and ITeration Algorithms) is offered for simulation of discrete event systems. It implements it's own simulation and specification language on the IBM-compatible PCs. The language contains a small set of basic elements that provides possibility to describe model's action scheme in a graphic form. Thus, it is easy to learn and to use the language for the system specification. The set of the basic elements is enough for description of large and complicated systems. New elements, more useful for some models, can be derived from basic elements. The usage of derived elements is helpful in the specification process. The description becomes more understandable. Thus, SITA is suitable both for teaching and learning, and for scientific and practical research.

The specification and simulation language SITA has it's software implementation - simulation systems SITA/B, SITA/C, and ITA (ITeration Algorithms) [2]. These systems provide the possibility to describe the models in different forms. So, SITA/B allows to build the action schemes of quite simple models by graphic elements. In ITA, anyone can describe quite simple models in a table form. ITA builds the system of equilibrium equations on input data, and solves it by using iteration method. By analytic calculations, as ITA does it, it is possible to get the system characteristics more quickly than by simulation.

The system SITA/C provides the model description in C-like programming language supplement by some simulation operators and pseudo-graphic notation. The simulation and analytic modelling by the same simulation program can be performed using SITA/C. Thus, the system is suitable to validate either how perfectly a model simulates the real system, or how accurate the given formula describes the behaviour of the real system.

The system SITA has been widely used for specification and simulation of teletraffic systems. In the current paper, a case study on simplified model of a factory supply department illustrates the SITA facilities. The department has N vehicles taking required elements from warehouse to workshop. Factory has K warehouses. In every of them V workers have been working to load the vehicles. The distance between any warehouse and the workshop differs. In the all warehouses, the average service time B is equal. The vehicles form the Poisson flow into warehouses with

parameter L. They choose i-th warehouse with probability P_i (i=1, 2, ..., K; $P_1+P_2+...+P_K=1$). There is possible to get the required elements with probability R in every warehouse. If the necessary elements are not in the warehouse, the vehicle is going to next one. Fig.1 shows the model's action scheme, built by SITA, for this system. Performing the simulation program, the information on system characteristics can be got, for example, to calculate the time consumed by a driver of vehicle in order to get the corresponding elements.

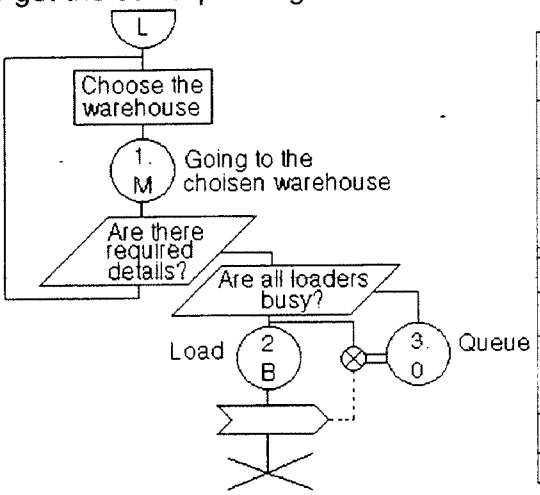


Fig.1. The model's action scheme for the factory supply department.

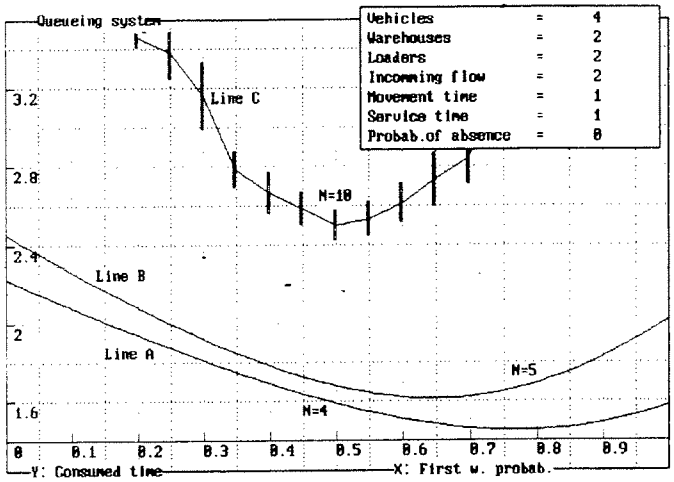


Fig.2. Simulation results.

2. CALCULATIONS AND DATA ANALYSIS

SITA contains a great amount of tools for processing and analysis of simulation results - for calculation of statistics, for building of histograms, for graphic building, etc. The usage of the tools provides possibility to apply the simulation as numerical method by which it is possible to estimate approximately the characteristics of the real system.

For example, Fig.2 displays the coherence between the average time, consumed by a driver of vehicle in order to get the required elements, and the probability by which the drivers choose the nearest warehouse. The parameters of the model are shown in the right upper corner of the Fig.2. Looking on the line A, someone can find: if there are four vehicles (N=4) and drivers are choosing the nearest (first) warehouse with probability 0.8, they spend the least average time for elements supply. Such information is helpful for the supply manager of the factory managing the elements deliver for the workshop.

3. NUMERICAL METHODS AND HYPOTHESIS TESTING

The system SITA provides both simulation and analytic modelling. SITA/C allows to do that by the same simulation program. However, the analytic modelling can be applied only under corresponding requirements. Thus, SITA can be used to validate how perfectly a formula or a statistical model describes the behavior of the real system. Besides, SITA provides possibility to save results of various trials. That allows to estimate how essentially the change of some parameters influences the model's behaviour and so on.

In Fig.2, the lines A and B display results calculated by analytic modelling, but the line C shows those got by simulation. Looking on the figure, someone can claim the following hypothesis: in the described model, if the number of working vehicles is quite large ($N \geq 10$), the less average time,

consumed for elements deliver, is spent in the case when the drivers are choosing the corresponding warehouse with probability 0.5. In order to examine that, the simulation results can be transferred to some statistical programs, for example, MINITAB, SPSS, etc. [4]. Subsequently, its tools can be used for testing of the hypothesis.

4. PROTOTYPING AND DEBUGGING

It is proved that the simulation system SITA can be a suitable tool for design and debugging of real-time systems [3]. SITA/C is useful for prototyping of these systems. In such system, both program modules, and technical devices can be substituted by their prototype. Besides, the "external environment" can be simulated using prototype. For this purpose, several actions should be done. At first, the model of the system must be built by SITA/C. Then the simulation program had to be debug and run; the results should be analyzed, and the efficiency of the simulated system must be estimated. Often, it is necessary to build some alternative models.

When the most appropriate model has been chosen, it's parts are replaced with corresponding programs and devices. SITA/C uses Turbo C (or Borland C) compilator. Thus, the system is more suitable for prototyping of systems built by programming language C (or C++). In order to SITA for debugging and prototyping of other systems, additional interfaces should be implemented between SITA/C and corresponding programming language.

5. TRAINING

Simulation has taken an important place in the training. Usually, visualization, economy (time-consumption, job-consumption, materials, etc.), safety, etc. are meant as the main advantages of the simulation in the instruction. SITA is a suitable system for training purposes, too. The system is easy-to-learn. Using SITA, students can describe the same model in various forms. That is why the system is suitable for training programming and modelling.

The system has a wide set of tools for data processing and analysis. Thus, SITA can be used to simulate the experiments and the results can be utilised for teaching and learning methods of the theory of probability and mathematical statistics. [4].

SITA animation facilities permit to use the system for demonstration of different processes [5]. Simulation animation is an efficient way how to show the development (dynamic) of the simulated system. At the same time, it is also appropriate tool for model validation and debugging. In SITA/C Trace mode, it is possible to form the file where the system records all the simulated events. A special SITA program translates this file to the ProofAnimation trace file. Subsequently, animation of simulation results can be performed by ProofAnimation.

Besides, SITA/C includes dialogue-building tools. That provides mutual interaction between either SITA and user, or SITA and another software system. The last feature is essential for contemporary simulation and animation. The facilities of simulation animation and dialogue-building are the main condition for implementation of simulation games by SITA. The games take especial place in the teaching and learning process. They are mainly used for training the management and decision-making. The games can be used both for student training, and for further education of specialists.

For example, the game has been designed based on the above described model. The player can play the game in one of two modes. In the first one, he is the dispatcher of the factory. His main duty is to deliver the elements in the appropriate time giving corresponding instructions to every driver,

which warehouse to attend. In the second mode, user plays the driver's role. The simulation program controls the remaining drivers. In both modes, SITA dialogue means provide the user's control over the corresponding drivers. When the game is over, he can compare efficiency of his strategy to the one computed by computer. Playing the game, user can get experience for analysis of different situations and for decision-making under uncertainness. Besides, the player better understands how the stochastic systems work.

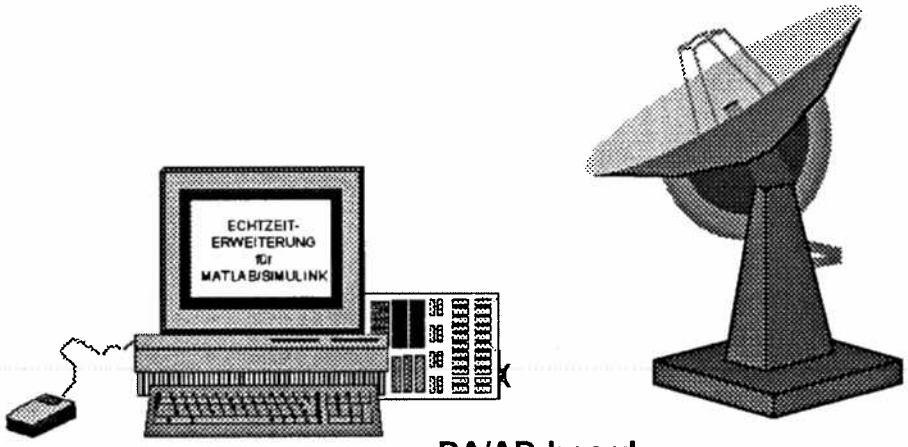
6. CONCLUSIONS

Simulation is a powerful tool for the solution of various tasks in different application areas - science, hardware, education and other areas. The mentioned above enumeration of the application fields is not closed. It is only showing those classes of the tasks, where simulation system SITA is used.

REFERENCES

1. Kindler E. (1985) "Simulation languages" (in Russian), Energoatomizdat, Moscow, Russia, 288 p.
2. Jonin, G.L., Sedol, J.J. (1991) "SITA - The Language for the Description of the Teletraffic Systems Simulation Algorithms" in *Proceedings of 13th International Teletraffic Congress*, Copenhagen, Denmark, pp.219-224.
3. Jonin, G.L., Kalnina, D.A., Sedols, J.J. (1991) "The Application of the System SITA/C for Simulation of Package-Line Communication Networks" (in Russian) in *Proceedings of International Conference on Functionability Problems of Communication Networks*, Novosibirsk, Russia, pp.143-149.
4. Jonins, G., Sedols, J., Tomsons, Dz. (1994) "The Application of Simulation System SITA for Statistical Analysis and Education" in *Proceedings of 15th Nordic Conference in Mathematical Statistics*, Lund, Sweden.
5. Tomsons, Dz., Galibeckis, J. (1995) "The Animation Facilities within Simulation System SITA" in *Proceedings of European Simulation Multiconference*, Prague, Czech Republic, pp.837-840.

ECHTZEIT-ERWEITERUNG
for SIMULINK™ on PCs



DA/AD-board

The ECHTZEIT-ERWEITERUNG is an extension for SIMULINK to allow **realtime simulation** on a PC under MS-DOS and MS-Windows.

No additional software is necessary. You can use **directly** SIMULINK models for your realtime simulation. No code generation and compilers are required. Models with MATLAB- and S-functions in MATLAB-code or as MEX-files can be simulated in a realtime environment without any changes. Only discrete blocks and some blocks with memory are not allowed. These blocks will be added in a future release.

Sampling times from as low as 0,3 ms up to some minutes can be reached on a Pentium PC. Simulation results can be monitored during the realtime process using a ONLINE-GRAPHIC.

Application examples are:

- Evaluation of measurement values
- Hardware in the loop simulations
- Processcontrol
- Monitoring

Special simulation hardware and software is not necessary. The product supports access to standard data acquisition boards, as **Lab-PC+** from National Instruments, **DAS-1600** from Keithley Instruments and **PC30DS** from Meilhaus. Other boards are available on request. The product supports additonally boards for pulse generation and DA/AD-convertion.

With the ECHTZEIT-ERWEITERUNG we supply a very costeffective extension for realtime simulation. Current applications include testenvironments in industrial environments and at research labs and experiments for educational purposes.

The industry price (inclusive data aquisition board) is DM 4.950,00. Please ask for university prices.

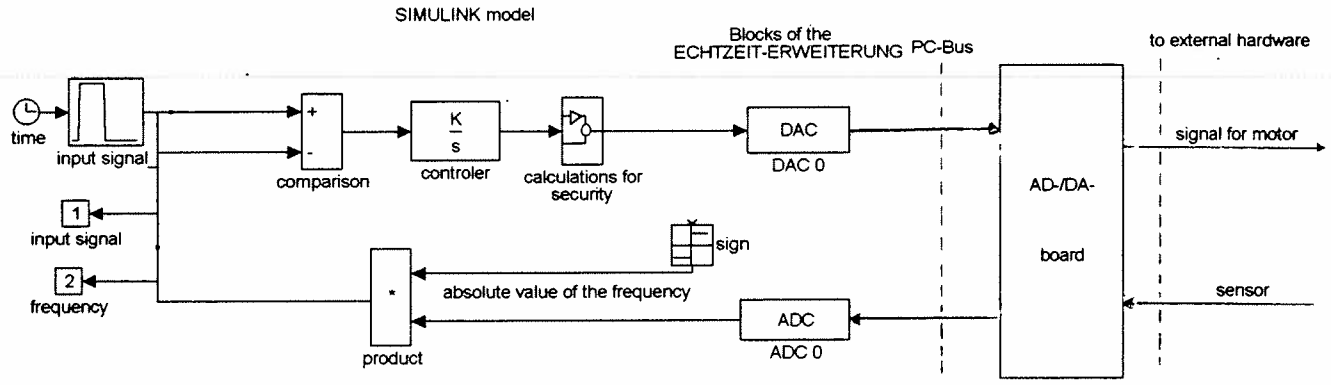
Example for a realtime simulation

The ECHTZEIT-ERWEITERUNG is called from the **MATLAB** command window as:

[t,y] = rtrun('MODELL',[T0 Tstop], Tsamp, X0);

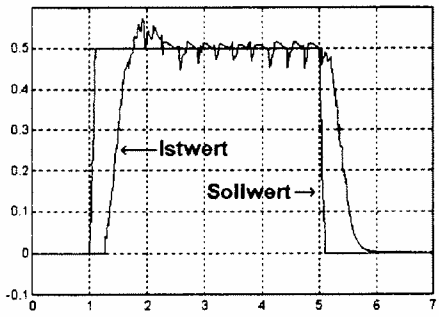
with starttime T0, stoptime Tstop, sample time of the realtime process Tsamp and initial conditions X0.

Realtime SIMULINK simulation model

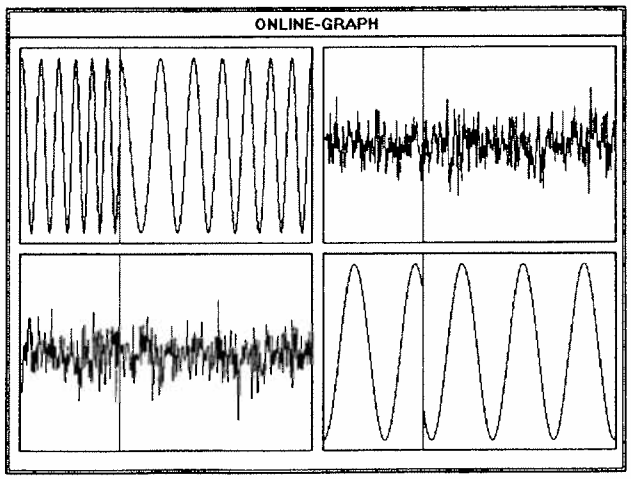


Realtime run

```
% Parameter of the I-controler
K = 3;
! Start realtime simulation
[t,y]=rtrun('drehzreg',7,0.01);
plot(t,y);  ! Plot results
axis([0 7 -0.1 0.6]);
grid;
```



ONLINE-Graphics



Linkage of a CAE-Simulation-Tool to a Realtime-Operating System

Klaus Beck
ARS Integrated Systems GmbH
Starnberger Strasse 22
82131 Gauting/München
Email: kb@ars-isi.de

Abstract

The use of the SystemBuild product for modelling a continuous engine coupled with an event based controller is presented in this paper. The example makes use of a recently developed interface between SystemBuild and the pSOSim product, a simulator for embedded tasks based on the pSOS+ real-time operating system. This integration allows high fidelity simulation of asynchronous and periodic controller tasks running on a real-time embedded kernel in an automotive environment.

1. Introduction

Popular computer aided control design products can effectively simulate continuous, discrete and hybrid models. However, real-time automotive controllers are not only periodic in time but are often asynchronous, i.e. event driven, having to respond to hardware generated interrupts. As a result, it has not been possible to use conventional control design software to perform high fidelity closed-loop simulations consisting of a continuous behavioural model of the plant combined with an accurate representationof a real-time asynchronous controller.

This paper demontrates the use of recent enhancements to the SystemBuild product that enables the closed-loop simulation of a continuous of discrete time behavioural model and a real-time event driven controller. The bevavioural model of the plant is numerically integrated within SystemBuild which controls the progression of the simulation time. The state-event cabability within SystemBuild is utilized to find the occurrence of sensor detected physical events which correspont to asynchronous interrupts within the control micro-processor. The event occurrences are utilized by a newly developed interface between the SystemBuild simulator and pSOSim to synchronize the execution of real-time control tasks to the simulation of a behavioural model. pSOSim is a simulator for embedded tasks based on the pSOS+ real-time operating system. pSOSim will also simulate the behaviour of pSOSelect, a highly scaleable version of pSOS+ that can require as little as 2K of ROM.

The example control problem used in this paper is a multi-tasking engine management controller that is both event-driven and time periodic. Modern engine control software is typically synchronized to the physical rotation of the crankshaft by means of sensors that generate interrupts in the engine controller. The tasks that are associated with the crankshaft sensors, are asynchronous in the time domain, but periodic in the crank angle domain. The control software also has time periodic tasks that calculate the engine speed which is used as a control variable.

2. Software Architecture

The simulation architecture, which is illustrated in Figure 1, consists of two processes running on the same UNIX workstation. The control loop is closed within the SystemBuild graphical modeling environment which completely encapsulates the simulation. The child pSOSim process is created automatically at the start of each simulation.

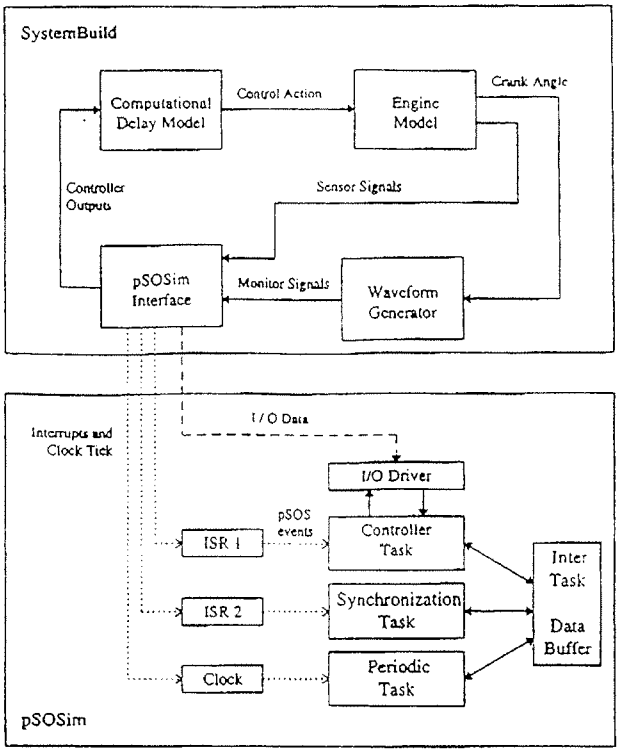


Figure 1, SystemBuild and pSOSim software architecture

The SystemBuild simulator performs the numerical integration of the system dynamic equations and controls the progression of time during the simulation. The four components of the SystemBuild model include the behavioural model of the engine, a waveform generator, the pSOSim interface and computational delay. The engine model is described later in this paper. The waveform generator uses the crank angle from the engine model to create an oscillating monitor signal for each rotational event. These monitor signals are input to the SystemBuild state-event detection algorithm where each zero crossing corresponds to the occurence of an event. The detected events can be time periodic, periodic with respect to a system state variable (such as the crank angle), or completely asynchronous. The magnitude and shape of the monitor signals are not important, provided that their zero-crossings occur at the proper time.
The sensor information from the engine model and the monitor signals are input to the pSOSim interface block. The interface block can have an arbitrary number of sensor inputs and is able to monitor multiple event signals. If the system has asynchronous events, the interface block must be defined within a continuous subsystem. In which case, each asynchronous event and the controller time clock will require a unique monitor signal. The monitor signal for the clock is generated from within the pSOSim interface block based on the specified clock tic interval. If there are no asynchronous events, then the interface block can be defined within a discrete subsystem where the sample rate corresponds to the controller clock tic interval.

Each time one of the monitor signals experiences a zero-crossing, SystemBuild supends the integration algorithm and releases control to the event processing software within the interface block. The SystemBuild simulator is blocked during event processing, after which the integration algorithm will compute a new consistent operating point and resume the simulation. In effect, the dynamic equations are numerically integrated piece-wise continuously between event.

The event processing software in the interface block communicates the sampled sensor signals and an event identifier to pSOSim through an inter-process communication (IPC) interface that uses a socket for data transmission and a UNIX signal for the handshake. The interface block will then wait for pSOSim to complete execution of the appropriate task in response to the event, and read the controller outputs back from pSOSim through the same IPC interface. When the pSOSim process receives the handshake signal from SystemBuild, it will automatically execute the appropriate

interrupt service routine (ISR) that corresponds to the event identifier. Each ISR can create a pSOSim+ event, causing data sampling and/ or task execution. Both the reading of the sensor signals and the writing of the controller outputs are done through a device driver that is written specifically to correspond to the communication protocol that is used between pSOSim and SystemBuild. The interface between the device driver and the pSOS+ tasks are written to emulate the actual device driver used in an embedded processor. As a result, the simulated ISRs and the tasks are the same as those that would be used within an embedded processor in a vehicle.

The computation in pSOSim as a result of an event are instantaneous with respect to the SystemBuild simulation. As a result, a computation delay must be modeled within SystemBuild between the pSOSim interface block and where engine model applies to controller outputs.

This simulation framework will not show the effects of preemption due to the occurence of events while the controller is executing a task. The exact timing characteristics of the controller software, including the potential for task preemption, are target specific and can be verified by running the controller software on a real-time processor.

The pSOS+ controller tasks can be user written, or automatically generated by the AutoCode software. AutoCode is a customizable code generator for SystemBuild models. AutoCode generated tasks that run on pSOS+ are realizable using a special set of template files.

3. Engine Model

The plant model used in this paper is a nonlinear 4-cylinder spark-ignition engine model [I]. Note that exhaust gas recirculation has been neglected.
The SystemBuild state event detection capability is used both in the engine model, and by the pSOSim interface.

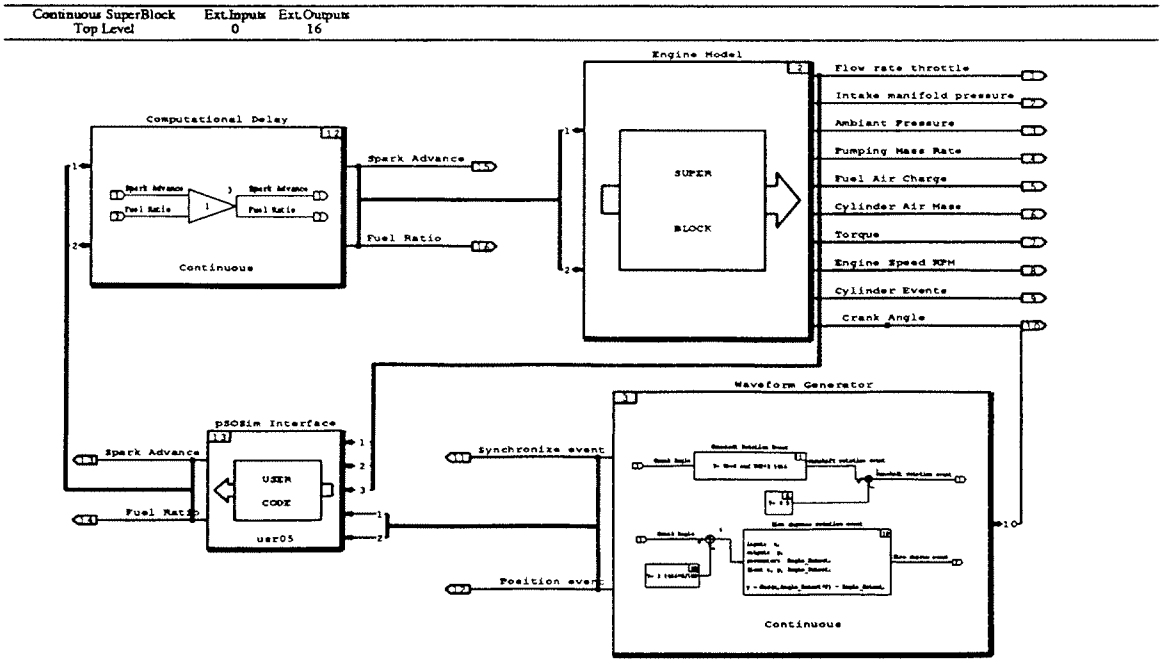


Figure 2, Top Level SuperBlock model

4. Real-time Controller

The real-time engine management controller is a multi-tasking controller that uses one periodic task and two asynchronous tasks. The periodic task is driven by a real-time clock in the embedded processor or by SystemBuild in this simulation. The two asnychronous tasks are driven by crankshaft mounted sensors that generate interrupts as the engine rotates. One of the sensors generates an interrupt every 5 degrees of crankshaft rotation and is called the position interrupt. The other sensor generates an interrupt every 360 degrees of engine rotation and is used synchronize the controller to

the crankshaft rotation. The three tasks cooperatively calculate two state variables, engine speed and engine position, and computes the control actions. Communication of calculated state variables between these tasks is accomplished through the use of a global inter-task data buffer.

The periodic task executes at 10 ms and performs the single function of calculating the engine speed, in revolutions per second (RPS). This is done by reading the position interrupt counter from the data buffer and deviding that value by the sample period. The calculated engine speed is then written back to the data buffer and the position interrupt counter is reset to zero.

The controller task is driven by the position interrupt and three functions. First it increments the position interrupt counter that is used by the periodic task. Second, it increments the position variable in the data buffer that represents the current physical position of the crankshaft. And lastly, this task calculates the appropriate control actions based on the computed state variables and the sensor values read from the device driver. The resulting control actions are finally written back to the device driver.

The synchronization task is driven by the less frequent interrupt that occurs every 360 degrees of crank rotation at top-dead-center for the first cylinder. This task resets the postion state variable which synchronizes the physical orientation of the crankshaft with the control software. During the startup phase, this task also enables the control task to safely take action by setting a flag in the data buffer once the control software computes the actual crankshaft orientation.

The control task employs a control algorithm that is implemented as table lookup structure that uses four inputs. These inputs the computed engine speed and the three sensor signals, intake mass flow rate, intake manifold pressure, and the ambient air pressure. The two computed control actions are the sparc advance and the fuel-air ratio. The engine model in SystemBuild employs the fuel-air ratio to manipulate a throttle-body fuel injector and the sparc advance to manage the engine performance.

This controller and engine model could be enhanced to support the simulation of various controllers and engines, including a sequential port fuel injection system where the injection timing of each individual port fuel injector is being sequenced with the timing of its respective cylinder. The number of sensor and actuator channels can be increased or decreased. The frequency at which the crankshaft position sensor generates events and the periodic task frequency are also customizable.

5. Implementation in a Real Vehicle

Implementation of a controller simulated in this environment requires the following steps:

1) Tie the ISR's controlling asynchronous tasks to externally generated interrupts.

2) Replace the ISR's controlling periodic tasks with events linked to timers. For example, the pSOS+ ev_every command can be used to generate an event every n pSOS+ timer ticks. For controllers running faster than the pSOS+ timer ticks, and ISR linked directly to a timer can be used, and the appropriate pSOS+ event can be created in the ISR.

3) Replace the I/O drivers with drivers for the physical I/O devices.

6. Conclusions

In this paper a coupling between two simulators has been presented and demonstrated. The simulators are SystemBuild, a general purpose simulator, and pSOSim, a simulator for applications running on the pSOS+ real-time operating system or pSOSelect. The combined environment allows accurate simulation of high fidelity plant models with periodic and event driven controllers. The actual embedded code for the controllers is used, which can be either hand written or automatically generated from SystemBuild.

ACSL For Real Time Simulation

Joseph S. Gauthier
MGA Software, Inc., 919-B Willowbrook Drive
Huntsville, Alabama 35802 U.S.A.
Tel: (205) 881-0947; Fax: (205) 883-5516

Introduction

The Advanced Continuous Simulation Language (ACSL) is a continuous system simulation language (CSSL) based upon the Society for Computer Simulation's CSSL standard. Based on Fortran and C, it permits the description of simulation models in terms of non-linear ordinary differential equations. Models written in ACSL are transportable among computer platforms ranging from PC compatibles to Unix workstations to mainframe supercomputers.

ACSL was first adapted for real time operation in 1975 for the U.S. Army Missile Command in an Analog-Digital hardware-in-the-loop environment. In this custom environment, ACSL resided on a Control Data mainframe host which would communicate with multiple analog consoles, digital minicomputers and actual missile guidance hardware in real time. This system supported radar, infra-red and electro-optical test environments for missile seeker and autopilot testing.

Since this first real time version of ACSL, there have been many other custom implementations of ACSL in real time world-wide. ACSL for real time (ACSL/rt) is available commercially on PC compatibles running Windows 3.1, Silicon Graphics workstations and Harris Night Hawk workstations.

There are many definitions of "real time". For ACSL, a real time simulation is a time critical simulation. If a real time ACSL model uses a ten millisecond integration step size, then it will take a single integration step every ten milliseconds of real time. The model must finish taking its integration step within ten milliseconds of real time or the simulation fails.

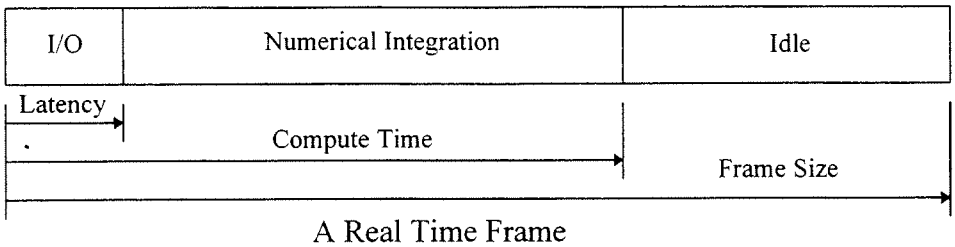
The special features of ACSL/rt required for real time operation are:

- Synchronize numerical integration steps with a real time clock
- Communicate with the outside world via real time analog and digital I/O channels
- Record time history data on disk for later analysis

These features are handled automatically by ACSL/rt so that models written in ACSL are platform independent.

The Real Time Frame

Real time simulation is concerned with hardware I/O rates. This update rate defines a time interval known as a *frame*. A simulation's integration step size is interpreted by ACSL/rt to be the size of the real time frame. On any particular computer system, ACSL/rt will see to it that the user's model is activated at each real time frame in order to take a single integration step. The frame size imposes an upper limit on the amount of work which can be done during an integration step.



A Real Time Frame

As shown, a frame is concerned first with performing I/O with the real time hardware. The time measured from the start of the frame to the end of hardware I/O is called the *latency* time. This is the time delay between the start of the frame and the point at which the model gets control to take an integration step. Latency also includes any overhead time required by the computer system to transfer control to the frame handler. The latency defines the smallest possible frame size which can be achieved by a computer system. The latency is determined by: the computer system's hardware architecture, software architecture, and the amount of time that the computer's CPU must be involved with moving data between the hardware I/O board and memory.

The *compute time* is the time measured from the start of the frame to the end of the model's integration step. It defines the smallest possible frame size which can be achieved by a particular simulation. The compute time is the sum of the latency and the time required to take an integration step. The factors affecting the integration step performance are: the computer's scalar floating point speed; the selection of the integration algorithm; and the total number of model equations. (The number of states in the model has minimal effect.)

ACSL/rt also supports multi-section simulations. Portions of the simulation can be assigned to different integration algorithms with different integration step sizes. In such systems, the section with the largest integration step defines the frame size.

Real Time Synchronization

ACSL/rt automatically handles the details of synchronizing with the real time clock. The simulation is only concerned with specifying the frame size by defining an integration step size. ACSL can use two different techniques to achieve synchronization: polling or interrupts. In the polling approach, ACSL continuously reads a clock driven counter until the frame interval time has elapsed. In the interrupt approach, the computer's hardware and software architecture are used to activate an interrupt handler which initiates frame processing.

The advantage of the polling approach is that latency is minimized by avoiding the overhead of interrupt processing and computer context switching. A disadvantage of polling is that the CPU is dedicated to the real time process. During the "idle time" described in the real time frame diagram, the CPU is spinning in a tight loop, locking out any other lower priority tasks.

The advantage of the interrupt approach is that the CPU can truly be idle during the "idle time" described in the real time frame diagram. This frees the CPU for other tasks. The disadvantage of interrupts is the potential for increased latency required for interrupt processing and context switching.

External interrupts can be handled with either the polling or interrupt approach. In the polling approach, the external signal is physically connected to a counter so that a change in counter value signals the start of a frame. In the interrupt approach, the external interrupt is assigned to the software interrupt handler which controls frame processing.

Real Time Hardware I/O

ACSL/rt simulations specify I/O operations to real time hardware boards by using a set of device independent operators:

- WDAC writes a scaled floating point value to a digital to analog channel
- WDIG writes an integer value to a digital channel
- RADC reads a scaled floating point value from an analog to digital channel
- RDIG reads an integer value from a digital channel

These operators handle the details of initializing the I/O board(s) and arranging for the exchange of data between the board(s) and computer memory. These generic operators are implemented for a variety of I/O devices. Specific devices are selected during ACSL/rt installation so that the ACSL models themselves remain transportable. The real time I/O operators provide default operations when used outside of real time, so that models can be checked out all digitally without requiring a special real time configuration.

Real Time Data Recording

The simulation's time history data needs to be written to disk for later analysis. The problem is that disk I/O is slow and not deterministic. In the PC implementation of ACSL/rt all real time processing, hardware I/O and data recording is handled by a single CPU. To minimize compute time, data is written to memory during real time and dumped to disk at the end of a run. Provision is made to record any number of data values for 10000 time points during a run on the PC.

Workstations configured for real time operation have multiple CPU's for true parallel multitasking. ACSL in real time on a workstation can dedicate a CPU to perform unlimited data recording

ACSL/rt for Windows on the PC

ACSL/rt on the PC runs a Watcom Fortran implementation of ACSL which operates as a 32 bit Windows/NT application. Microsoft's "win32s" software allows Windows to handle 32 bit NT applications. This configuration was selected for performance and for source code compatibility across DOS, Windows, Windows/NT and Windows/95 operating environments. The advantage of ACSL/rt on the PC is its price and ease of use. The primary disadvantage is its performance relative to workstation solutions.

I/O latencies as small as 50 microseconds have been measured under this PC configuration for short real time runs. It has been found that Windows adds almost a millisecond of overhead at irregular intervals when the same simulation is run for a longer duration. Windows was not designed to be a real time operating system with deterministic performance.

In the PC world, Keithley Metrabyte and National Instruments build real time I/O boards for the standard PC bus architecture. Synchronization on the PC version of ACSL/rt is done by polling the two cascaded 16 bit counters on a Keithley DAS-1600 board. Timer resolution is 100 nanoseconds.

ACSL/rt for Silicon Graphics

The Silicon Graphics (SGI) version of ACSL/rt makes use of SGI's "React" real time software package running on a multiprocessor architecture. SGI was selected for ACSL/rt because it is one of the major workstation manufacturers and because it has a vested interest in real time operations due to its market niche in real time video processing. The advantage of SGI is in dealing with a major hardware vendor, using Unix and the availability of a parallel processing solution.

SGI uses a multiprocessor approach to maximize deterministic performance in real time. The first processor (processor 0) runs the Irix (SGI's Unix) operating system and system daemons. The other processors can be dedicated to individual real time tasks. Since most real time I/O boards use the VME bus architecture and SGI uses a proprietary system bus, SGI computers configured for real time add a VME bus to the existing proprietary bus.

ACSL/rt uses SGI's React as an interrupt driven real time system. As a result, latency will be higher than in a polling implementation. SGI guarantees a latency of less than 200 microseconds, but this does not include hardware I/O time. SGI's CPU's are among the fastest in the workstation world, so what is lost in latency is made up during integration.

ACSL/rt for the Harris Night Hawk

The Harris Night Hawk is a computer system designed for real time simulation. Its CX/UX (Unix) operating system supports the Posix 1003.4 standard. This revision of the Posix standard provides for accessing high precision timers (one nanosecond resolution) to implement a polling solution to real time synchronization. Latency is about 20 microseconds. Like Silicon Graphics, Harris uses multiple CPU's to achieve deterministic real time performance. Unlike SGI, Harris uses a standard VME bus to facilitate the use of standard real time I/O boards.

The advantage of Harris is its demonstrated dedication to the real time simulation market. Its hardware and software solutions have been optimized for real time. In addition, Harris is an experienced real time systems integrator.

Other Platforms

ACSL/rt can easily be ported to other hardware/software platforms as the real time market develops. Potential future systems include Sun, HP, DEC Alpha and IBM workstations.

Summary

ACSL/rt extends the demonstrated success of ACSL's system simulation solution to the real time world. The details of solving the real time problem have been encapsulated into ACSL/rt so that simulations can easily move from the all digital world to the hardware in the loop world and from low end platforms to high performance platforms.

AUTOMATIC CODE GENERATION FOR MULTI-DSP NETWORKS ON THE BASIS OF SIMULINK BLOCK DIAGRAMS

Dr.-Ing. U. Kiffmeier
dSPACE GmbH, Technologiepark 25, D-33100 Paderborn, Germany
E-Mail: ukiffmeier@dspace.de

Abstract

For advanced controller prototyping and sophisticated real-time simulations the performance of a single processor may not be sufficient. A solution to these advanced performance requirements can be a multiprocessor implementation. This paper describes a tool, that allows graphical-oriented programming of multiprocessor systems for real-time applications. SIMULINK block diagrams are used to describe not only the model dynamics, but also the network structure of an application. The code generated by the Real-Time Workshop for each processor is automatically augmented by appropriate communication functions and embedded in a real-time simulation frame for multiprocessor systems. The close integration of the proposed tool into the SIMULINK environment allows easy handling of real-time multiprocessor applications.

1 Introduction

The real-time implementation of control systems by automatic code generation based on SIMULINK block diagrams has found its way into practice and is now widely used. The advantages of this approach are obvious:

- Easy graphical programming of system dynamics and shortened developement cycle.
- High portability of the application between different real-time platforms.
- Close integration into the MATLAB/SIMULINK® environment offering easy access to powerful toolboxes for analysis, synthesis and optimization of control systems.

The implementation of real-time code on single processors, e.g. digital signal processors (DSP's), is supported very comfortably by the Real-Time Workshop™ [Mat94] and the dSPACE Real-Time Interface to SIMULINK [dSPACE95]. If the performance of a single DSP is not sufficient, e.g. for a sophisticated Hardware-in-the-Loop simulation, an implementation on a network of multiple DSP's may be necessary. When facing a multiprocessor implementation, a number of problems can arise:

- The application must be distributed over the different processors of a network under the constraint that the CPU load of all processors should be nearly identical.
- The communication connections between the processors must be implemented. Deadlocks must be avoided and data transfer should be as fast as possible. In most control engineering applications a synchronization of the processors is necessary.

The programming of multiprocessor applications can be considerably simplified using a graphical-oriented approach. In this case, a SIMULINK block diagram is used to define the whole multiprocessor model including the communication connections. Based on such a block diagram description the *Multiprocessor Option for the dSPACE Real-Time Interface to SIMULINK* (RTI-MP) allows fully automatic code generation for multi-DSP networks with a single mouse-click. The implementation and background of RTI-MP will be described in the following.

2 SIMULINK Block Diagrams for Multiprocessor Systems

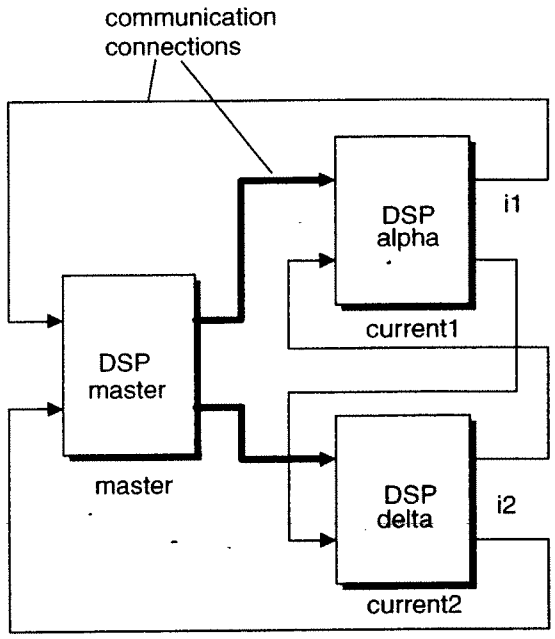


Fig 1: Network layer of a multiprocessor SIMULINK model

Fully automatic code generation is only possible, if all necessary information about a multiprocessor application is contained in the SIMULINK block diagram. This not only concerns the submodels to be implemented on each processor, but also the structure of the network and the communication connections between the DSP's. An obvious approach is to organize the block diagram in a hierarchical form as shown in Figs. 1 and 2. The top level of the model, the so called *network layer*, describes the number of processors involved, along with their connections. The submodels to be implemented on the different processors can be opened with a mouse-click on the corresponding DSP block on the network layer. This level of the block diagram, called *application layer*, can contain arbitrary SIMULINK blocks including hand-coded S-functions in 'C'. The signals to be exchanged with other processors are defined by special *ComPort icons*. For example, in Fig. 2 the signal *i1* is send from DSP „*alpha*", port 4 to DSP „*master*", port 2. The behaviour of the Simulink model is not affected by the ComPort Mux/Demux icons. This makes it possible to verify the dynamics of the whole multiprocessor model in an off-line simulation before implementing it on the real-time hardware.

The definition of the block diagram, i.e. the distribution of a simulation task over the different processors of a network, is the user's responsibility. In many cases the distribution of the model directly follows from the structure of the real system. For example, in an automotive application, the four wheels of a car could be simulated on four processors. The number of processors is not limited by software. Errors in the network structure, e.g. missing connections, are recognized and reported by RTI-MP automatically.

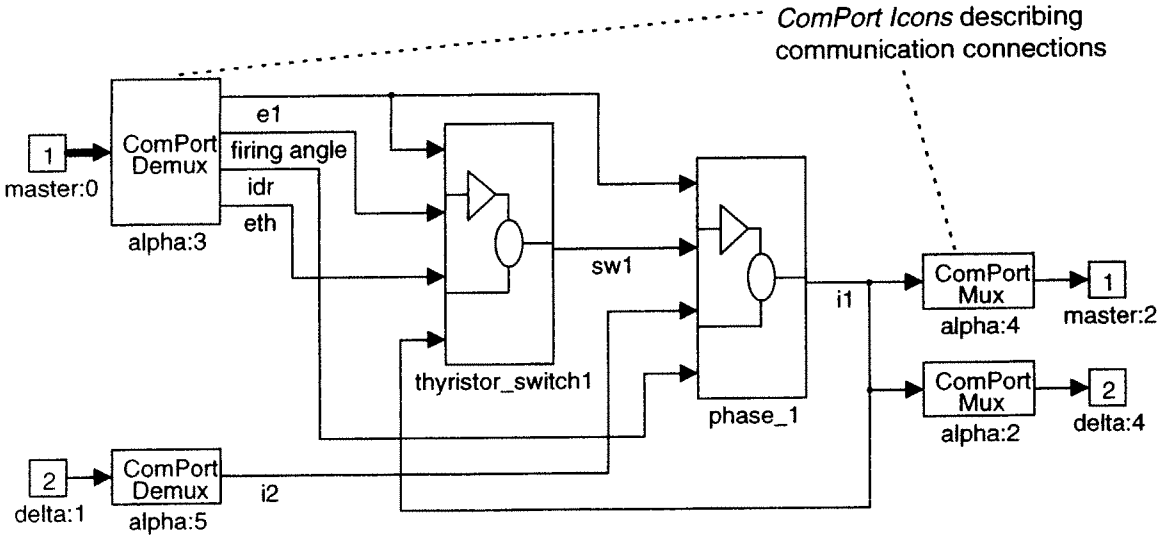


Fig 2: Application layer of a multiprocessor SIMULINK model

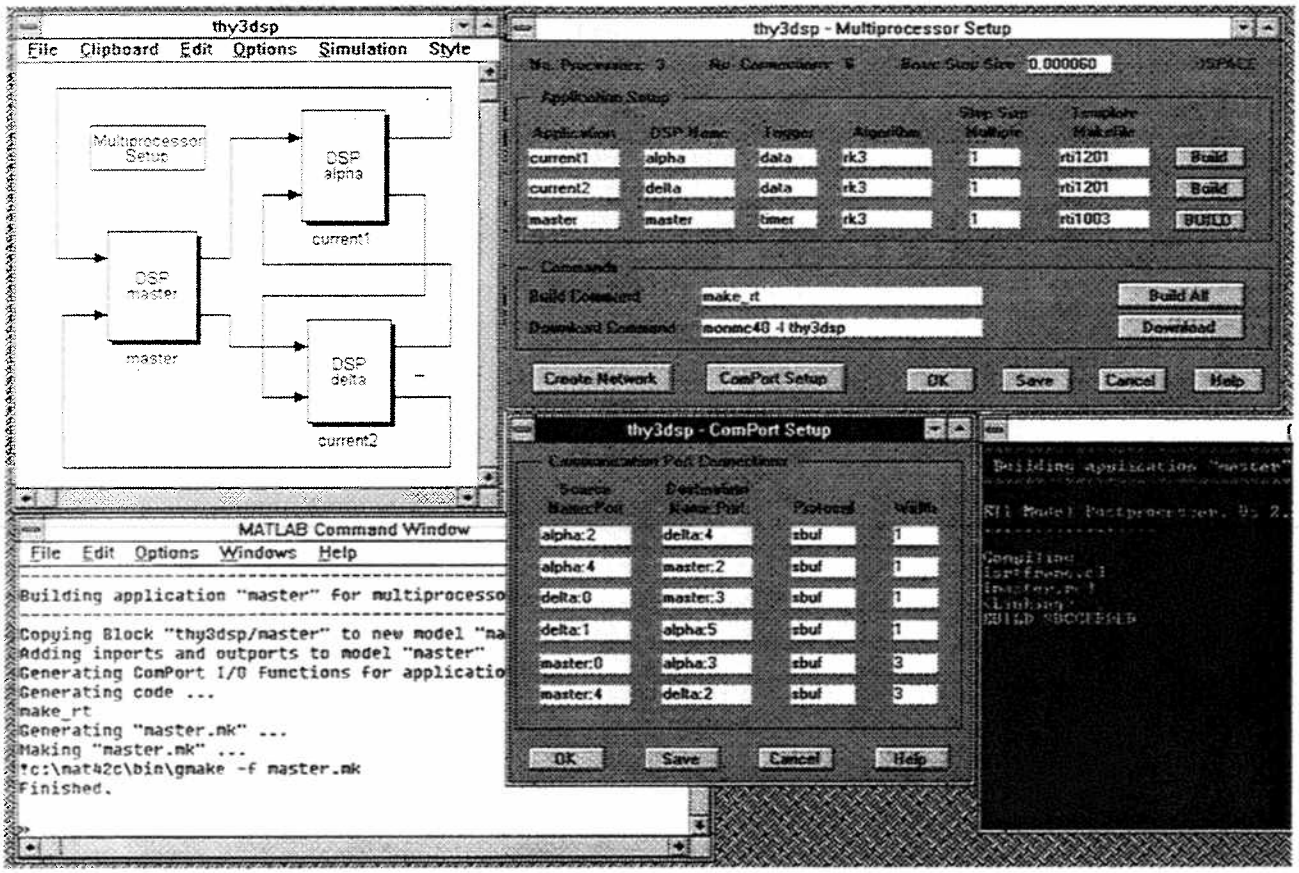


Fig 3: Automatic code generation with RTI-MP

The real-time simulation parameters are specified in a *Multiprocessor Setup* menu shown on the screen copy in Fig. 3. The following parameters can be assigned individually for each submodel of a multiprocessor application:

Algorithm The integration algorithm. Five different methods with fixed step size from *Euler* to *Runge-Kutta 5* are available.

Step Size Multiple Multiple of a basic real-time step size, used for integration of the model. The individual selection of the integration algorithm and step size allows the user to balance the load of the different DSP's in the network. For example, a small subsystem with fast dynamics can operate with a smaller step size than a large subsystem with slow dynamics.

Trigger This field defines whether the evaluation of a submodel is controlled by *timer* interrupts or by incoming *data*. In the first case, a new cycle of model evaluation is started at fixed time steps with a timer interrupt event. If a DSP is defined to be triggered by *data*, the corresponding submodel is evaluated as soon as a new vector of input data is received. This mode allows the fastest possible response on the inputs.

The parameters of the communication connections are specified in a second menu, the *ComPort Setup* menu also shown in Fig. 3. The generation of the communication functions and timing of the model evaluation is described in section 3.

With the **Build** and **Build All** buttons, automatic code generation can be started either for a single submodel or for the whole network. The Real-Time Workshop then generates a C file describing the dynamics of the corresponding submodel. This C code along with a set of communication functions generated by RTI-MP are embedded in a special real-time simulation frame for

multiprocessor applications. After successful compilation, the object files for each DSP in the network can be loaded to the multiprocessor hardware using the **Download** command. Before starting the application, the loader program verifies that all processors and the necessary connections are available on the hardware.

3 Communication in a Multi-DSP Network and Timing of the Model Evaluation

One of the most important factors for an effective multiprocessor system is a fast interprocessor communication. A processor very well suited for parallel processing applications is the TMS320 C40 DSP from Texas Instruments. Besides its impressive floating point performance of 50 Mflops, the C40 DSP offers 6 *communication ports* for high speed data transfer between the computing nodes. Each of the 6 communication ports achieves a transfer rate of 20 MBytes/sec. A DMA coprocessor enables data transfer parallel to the normal CPU operation. RTI-MP only uses communication mechanisms, that take advantage of the DMA coprocessor.

A very fast data transfer method is the *Virtual Shared Memory* (VSM) mechanism. In this case, the data is transfered continuously by DMA and written directly into the memory of the receiver DSP. While the DMA coprocessor is receiving data, the CPU executes its normal model evaluations at the same time. A synchronization with the sender is not performed, i.e. the receiver uses the available data as present at the time of reading.

In most control engineering applications it is desirable to have all input signals of a submodel result from the same simulation step of the sender; for example, if the submodel contains logical blocks. Since this is not guaranteed with the VSM, it is often better to use a *Swinging Buffer* (SBUF) mechanism for interprocessor communication. In this case, 3 buffers are allocated on the receiver DSP. While the DMA coprocessor is filling one of the buffers with new incoming data, the CPU reads its inputs from another. As soon as a new buffer is filled, the CPU can acquire it as new input, so that the data used by the CPU will always be consistent. Timing of the Model Evaluation

To achieve a computation flow free of deadlocks, the different submodels must be evaluated in a proper order. RTI-MP recognizes automatically whether or not a submodel has a direct feedthrough. According to this information, and the structure of the multi-DSP network, RTI-MP assigns the order of model evaluation for each DSP. To minimize the input/output delay of a submodel, the outputs are computed as early as possible and the inputs are read as late as possible. For systems without direct feedthrough, the outputs are computed before reading the inputs.

4 Conclusions

Automatic code generation for multiprocessor systems on the basis of SIMULINK block diagrams allows the user to access new classes of real-time performance. A key problem is the implementation of the communication between the processors and the timing of the model evaluation. Automatic generation of the communication functions with RTI-MP and simple graphical-oriented programming with Simulink simplifies the handling of multiprocessor systems enormously. Of course, it is the user's task to distribute a control application over the processors of a network.

References:

[Mat94] *Real-Time Workshop User's Guide*. The MathWorks Inc., 24 Prime Park Way, Natick, Mass., 1994.

[dSPACE95] *Real-Time Interface to SIMULINK User's Guide*. dSPACE GmbH, Technologiepark 25, 33100 Paderborn, Germany, 1995.

[Sch95] F. Schneider, F. Wienand and H. Rake: *Parallel Simulation of Complex Technical Processes*. EUROSIM Congress '95, Sept. 11.-15., Vienna, Austria.

scientific COMPUTERS

Technical Computing Environment MATLAB - New Developments -

Helmuth Stahl

Scientific Computers GmbH, Franzstr. 107, 52064 Aachen, Germany

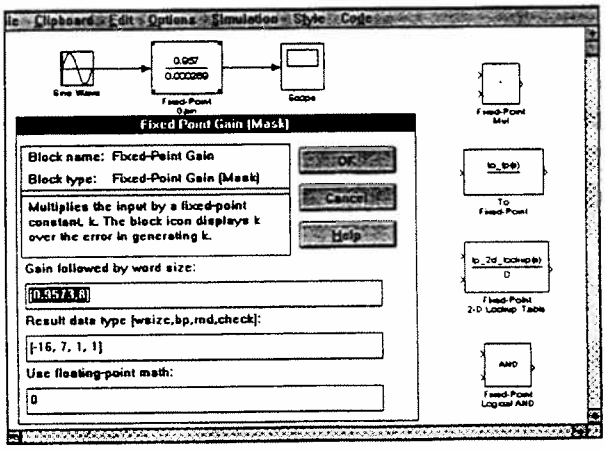
It is well known that ***Toolboxes*** extend the technical computing environment MATLAB for various application specific areas. Quite similar to this so-called ***Blocksets*** are available now to extend SIMULINK directly. These Blocksets constitute block libraries which may be used in SIMULINK simulation models and may be combined with other blocks as usual. It is important to know that the block functions are also supported by the Realtime Workshop.

The following pages shall give a general overview about new MATLAB related products which are just coming up or will come up in very near future.

● Fixed-Point Blockset - Block library for Integer Arithmetic in SIMULINK

The Fixed-Point Blockset offers SIMULINK V 1.3c users to simulate models or parts of them with fixed-point arithmetic. This new feature is based on a collection of more than twenty additional blocks like logical elements, blocks for the basic calculations and Lookup Tables. Users may perform calculations in an unsigned or the 2's complement format with 8, 16 or 32 bits. Besides that you may switch between floating and fixed-point format.

This allows to simulate effects that typically arise in control systems and digital filters applications which have been implemented on fixed-point hardware. When using the Realtime Workshop, C source code will also be generated for these fixed-point blocks.

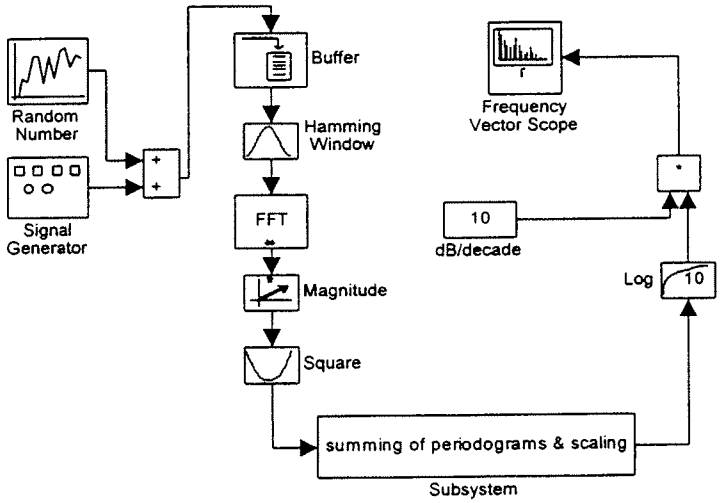


The figure illustrates some typical blocks of the Blockset and a dialog box for the fixed-point gain block. The dialog box offers the above mentioned ways to represent data types (*wsize=word size, bp=binary point location, etc.*) and to select floating point arithmetic. The block displays the gain quantization error when using an 8-bit word length.

● DSP Blockset - Block library for signal processing in SIMULINK

With more than 100 additional blocks the new DSP (Digital Signal Processing) Blockset extends SIMULINK in the signal processing areas. It targets engineers who need an easy-to-use software tool to develop and simulate DSP algorithms. This becomes increasingly important in areas like mobil communication techniques, medical or consumer electronics. Some of the major DSP Blockset diagnostic functions are:

- Basic operations in digital signal processing (e.g. FFT, correlation, etc.)
- Data buffering for parallel computing.
- Additional signal sources and sinks (e.g. FFT Scope)
- Filtering, filter design and windowing techniques (see Signal Processing Toolbox)
- Complex arithmetic (e.g. calculation of magnitude and phase) and vector-oriented math operations
- Compatible to Realtime Workshop, i.e. SIMULINK and the Realtime Workshop may be used to generate portable C code from a SIMULINK model for use on an external DSP board or processor.



DSP Blockset: SIMULINK block diagram to determine the power density spectrum according to the Welch method

The DSP Blockset is available on all MATLAB platforms. MATLAB 4.2c, SIMULINK 1.3c and the Signal Processing Toolbox 3.0b are required.

● LMI Control Toolbox - Linear Matrix Inequalities Control Toolbox

The LMI Control Toolbox is a new toolbox that extends the MATLAB product family in areas of control engineering and general math. LMIs are convex optimization problems that have to be solved e.g. in linear algebra, interpolation or control engineering and system identification.

The following equation depicts a typical example for an LMI problem:

$$A' * X + X * A + Q < 0$$

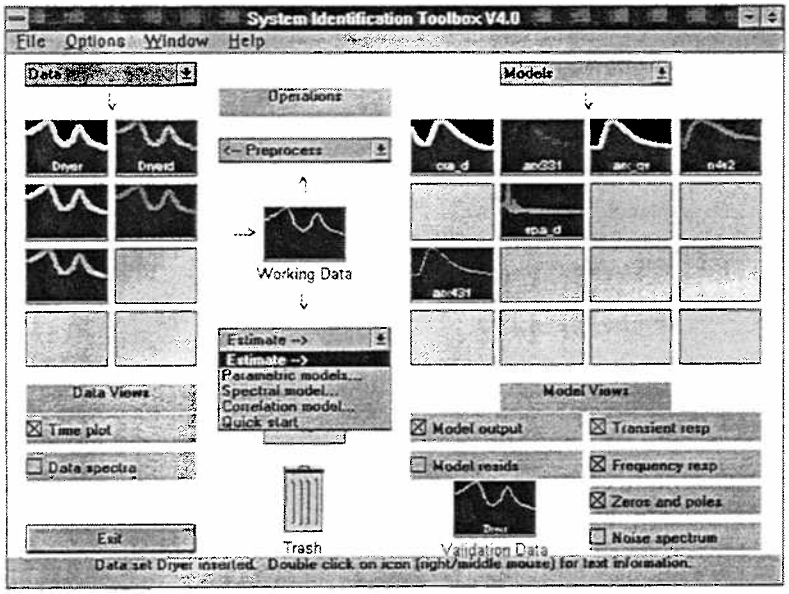
The strength of the LMI Control Toolbox is the availabilty of optimized algorithms and a powerful GUI environment.

scientific COMPUTERS

LMI will be available on all MATLAB 4.2c platforms. For control applications the Control System Toolbox is strongly recommended.

● System Identification Toolbox - Update: Version 4

The MathWorks responds to the request of many users and provides a graphical user interface (GUI) for the System Identification Toolbox. This new version includes a completely new GUI. Less experienced users and novices are guided throughout the system identification process (data preprocessing, modeling and model validation). Experienced users may compare different methods and actions transparently.



With the exception of one additional parameter esti-mation method for state space models the toolbox functionality did not change. As in the past all commands may also be entered via the MATLAB command line. The manual has been adapted with respect to the GUI.

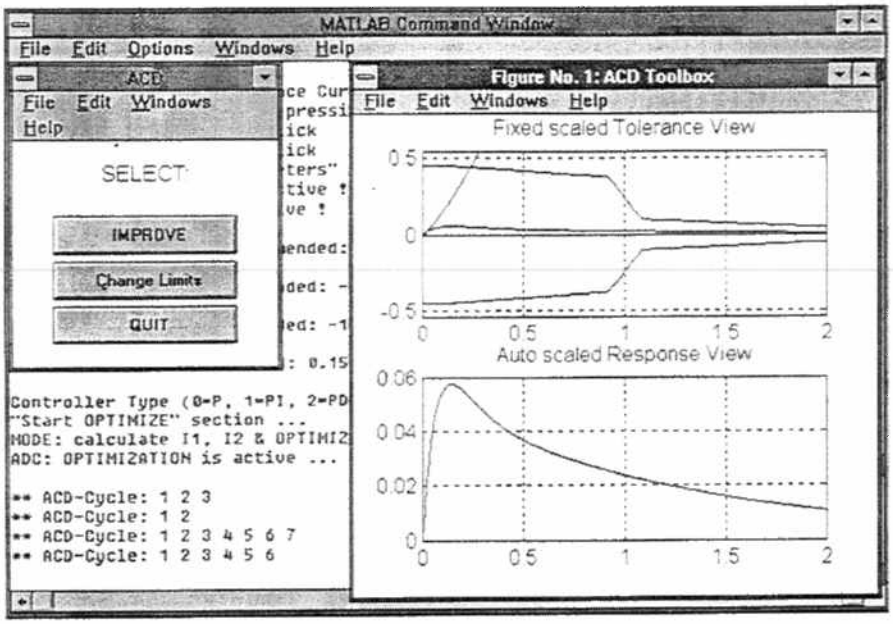
The System Identification Toolbox is available for all MATLAB 4.2c platforms.

● ACD Toolbox - Automatic Control Design Toolbox

This new toolbox provides powerful and very fast functions for a completely automated controller design which may also be carried out by non-experts in the control area. ACD especially provides all parameters for an ARW (Anti-Reset-Windup) controller structure to avoid undesired windup effects of compensators with an integral part. The major ACD characteristics are:

- results are controller transfer functions, also observer-based reduced-order state controllers or the parameters for the standard controller types: P, PI, PID-T_1, PD-T_1.
- works in a reliable way also in cases where the system to be controlled is unstable or contains pure time delay or system zeros are located in the right half of the s-plane.
- The Control System Toolbox is required.
- The desired responses and the corresponding tolerance regions may be defined graphically in the time domain. The ACD Toolbox may also be integrated into a user-written user interface.

- ACD is suitable to calculate good initial conditions for the Nonlinear Control Design Toolbox and the Quantitative Feedback Theory Toolbox.
- Closed-loop stability is always taken into account and guaranteed.



- Controller parameters for an ARW-structure are generated automatically, also for standard controllers.
- A reference filter is generated automatically. This means that optimized behaviours will be achieved both for disturbance and reference inputs.
- Robust controller design: several system models are allowed (parameter varied models or even different model structures).

● NAG Foundation Toolbox - Library of the Numeric Algorithms Group

The NAG Foundation Toolbox consists of a comprehensive set of more than 240 M-files with algorithms from optimization, statistics, partial and ordinary differential equations, integration methods, data fitting, etc.

The Toolbox is based on the NAG Foundation libraray which is well known for its comprehensive Fortran library containing more than 1100 routines for numerics and statistics. MATLAB users may access a set of these functions within the MATLAB environment. The function names of the corresponding M-files and the function call syntax are derived from the Fortran routines.

The NAG Foundation Toolbox is currently available under the MS-Windows-MATLAB.

● PCDAQ - PC Data Aquisition System for MATLAB

PCDAQ is a menu-driven PC software package for data aquisition and filtering for MATLAB. The aquisition is carried out in realtime while using Keithley Instruments' DAS1600 board.

Functionality: graphical representation of curves with printout capabilities on standard printers; measured data is stored in MATLAB format for post-processing within one of the various Toolboxes.

The simulation system ANA V2.0

J.W. Goldynia[1], J.M. Marinits[2]

ANA V2.0 is a CSSL based simulation system with a graphically driven frontend based on block scripts. It is able to handle state and time events in models at arbitrary precision and is available free of charge. The ANA product familiy is well established in education since 1986.

Introduction

The ANA V2.0 project has been initiated to support forthcoming control engineers with a powerful and yet flexible simulation system free of charge. The predecessor, the simulation program ANA 1.x, was invented in 1986 and is in wide spread use for control education at universities, technical colleges and courses. The secret of the success of ANA is the very simple user interface which allows newcomers to use the system from the scratch without any introduction.

The design goals of ANA V2.0 were to support the same userbase but to enhance the capabilities of the system tremendously and to define a solid base for future extensions. For that reason the capabilities of ANA V2.0 also satisfy the need of the professional user in industry and research.

The block oriented user interface of ANA 1.x is based on block diagrams. The contents of a block may include both differential equations and sequential algorithms to realize discrete or nonlinear behavior.

Since there is a need to introduce a full flagged CSSL within ANA V2.0 this new system is based upon a layer model. The frontend module ANAide covers all duties of input as well as presentation of results and composes an input script to ANAmdl - the CSSL compiler stage. The block library consists of ANAmdl templates which are assembled to a model.

Control engineers often deal with nonlinear and/or switching systems and therefore ANA V2.0 supports time events as well as state events. The possibility to determine the occurence of an event at an arbitrary accurency is just one of the advantages of ANA V2.0 towards ANA 1.x. An outstanding feature of this CSSL is the powerful Pascal like procedure processing which accompanies the state based ODE coding. This language has been successfully used to simulate complex hybrid models like power electronics circuits, helicopter flight dynamics and sewage works dynamics.

The ANA V2.0 CSSL-Compiler produces either PI code for immediate interpretation or C language code which needs to be compiled and linked with the runtime stage ANAsim. Integration algorithms for control engineering problems are adapted to handle events. Advanced explicit one step methods within the Runge-Kutta family of algorithms can be choosen amongst others to provide best results.

[1]Dipl.-Ing. Dr. Johannes W. Goldynia (goldynia@iert.tuwien.ac.at)
[2]Dipl.-Ing. Johann M. Marinits (marinits@iert.tuwien.ac.at)

Institut for Electrical Control Engineering, Vienna University of Technology,
Gusshausstr. 27-29/375, A-1040 Vienna, Phone ++43 1 58801 3906, Fax ++43 1 5058907

Some extensions, currently in process of development are a real time enhancement for hardware in the loop simulation, fuzzy logic extension and artificial neural networks by improving ANAmdl as well as analysis in the frequency domain like bode, locus and root locus plots.

The ANA V2.0 project is a multiplatform approach and currently available for MS Windows 3.1, MS Windows NT, MS Windows 95 and X/Motif without license fees and can be obtained via anonymous ftp from `iert.tuwien.ac.at` (look for ANA2).

Example 1

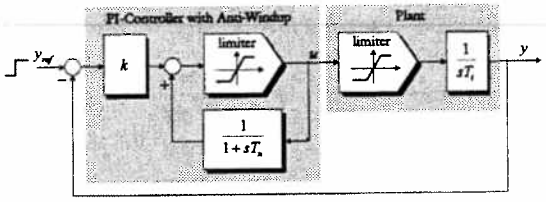


Figure 1: Control circuit with Anti-Windup measure

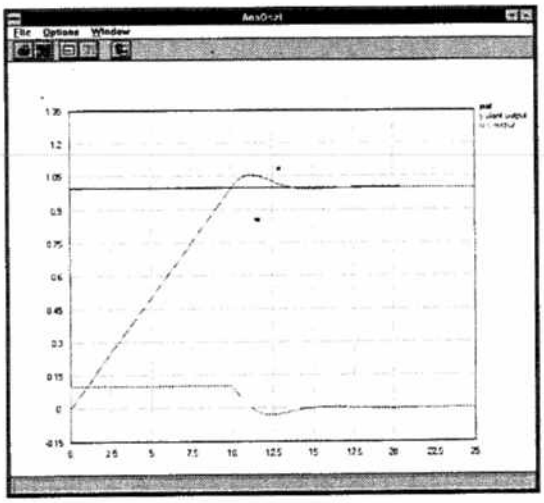


Figure 2: Step response simulated with ANA V2.0

Figure 1 shows a typical example of a simple control loop. The input of the model is straight forward since the graphical editor of ANA V2.0 is based on block diagrams. Figure 3 presents the graphical scheme including an additional block called `control`. Within this block the terminate condition (`te = 25`) for the simulation and the communication interval (`h = 0.25`) is set. All used blocks are part of standard libraries which can be extended or redefined by the user.

Figure 2 confirms that this structure of a PI-controller avoids any windup effect in the integral part of its structure. The plant output y is nearly a ramp function because the controller output is limited to the upper bound from the start.

The apperance of a block can be individually changed. This includes resizing and adding texts as well as drawings. Signal paths may be named. The user interface is intuitive. It can be driven by tool buttons, menus or mouse actions on the drawing pane. The bottom statusline informs the user about the current activity.

Example 2

The strength of ANA V2.0 is the CSSL called ANAmdl. This language is equipped with a powerful mechanism for handling time and state events. Each type of block element is defined by an ANAmdl description which acts as template if a block is used.

The statements in this template can be subdivided into information for the graphic interface (lines starting with `$`), declarations of inputs, outputs, parameters, states and variables and finally the equation body consisting of intialization, a simulation part and event controlled procedures (performed sequentially).

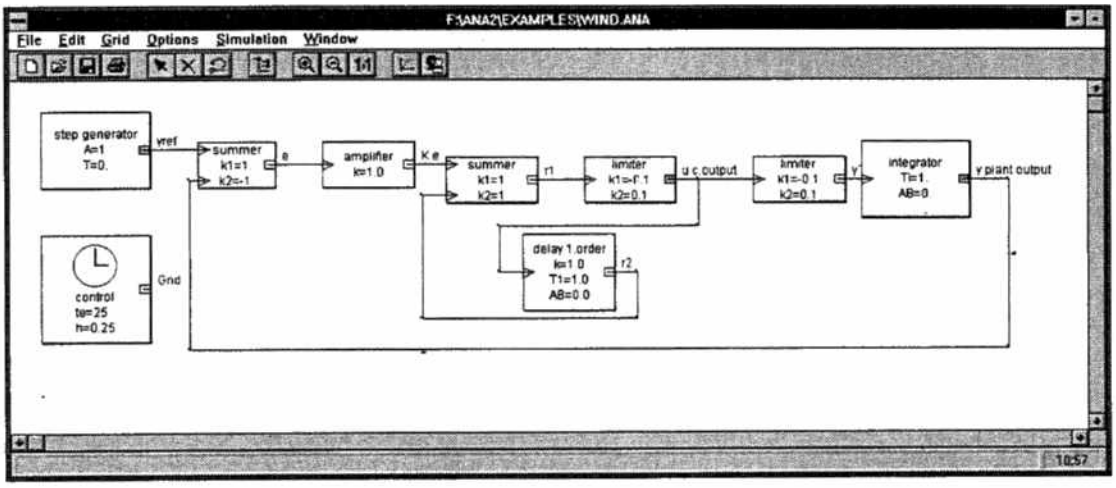


Figure 3: Control circuit drawn with ANA's graphic editor

A one dimensional model of a 'Jumping One-Wheel' is presented in order to illustrate the state event capability of ANA V2.0. Figures 4 and 5 give a formal description, figure 8 shows the corresponding implementation.

Lines 36 to 44 show the differential and algebraic equations. Lines 45 to 54 put the state event dependent equation of the force f into action. The variable `state` is declared as `DISCRETE` and is therefore able to memorize its value. In line 72 it's initial value is assigned. The `SWITCH` statement distinguishes `state == 0` (mass 2 standing on ground) and `state == 1` (mass 2 up in the air). In the case where mass 2 is standing on ground line 48 is carried out and the `ONRISE` condition on line 49 is constantly checked to trigger a call to the `PROCEDURE jump`.

Since ANAmdl allows arbitrary complicated `ONRISE` conditions, the evaluation of such a condition is controlled by specifying a boundary for the independent variable *time*, called *event precision*. ANA V2.0 guarantees that the associated procedure is called only once if the condition changes from *false* to *true*.

The procedure code of `jump` (line 59 to 63) changes the variables `state` and `fly` and saves an extraordinary communication record due to the statement `STORE ALL`. This ensures that the visualisation handles the discontinuities of the model properly.

Figure 6 and figure 7 show sample results. The variable `fly` was introduced to animate the state switching to the display.

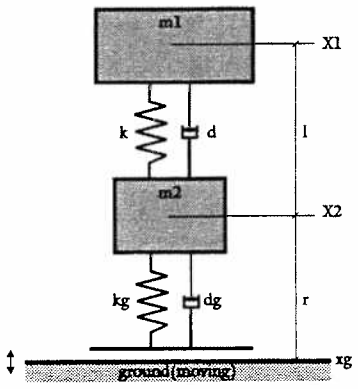


Figure 4: One dimensional model of a 'Jumping One-Wheel'

$$
\begin{aligned}
\frac{dx_1}{dt} &= v_1 \\
m_1 \frac{dv_1}{dt} &= -m_1 g - d(v_1 - v_2) - k(x_1 - x_2 - l) \\
\frac{dx_2}{dt} &= v_2 \\
m_2 \frac{dv_2}{dt} &= -m_2 g + d(v_1 - v_2) + k(x_1 - x_2 - l) - f \\
f &= \begin{cases} k_g(x_2 - x_g - r) + d_g(v_2 - v_g) & : x_2 - x_g < r \\ 0 & : \text{else} \end{cases}
\end{aligned}
$$

Figure 5: Simplified equations

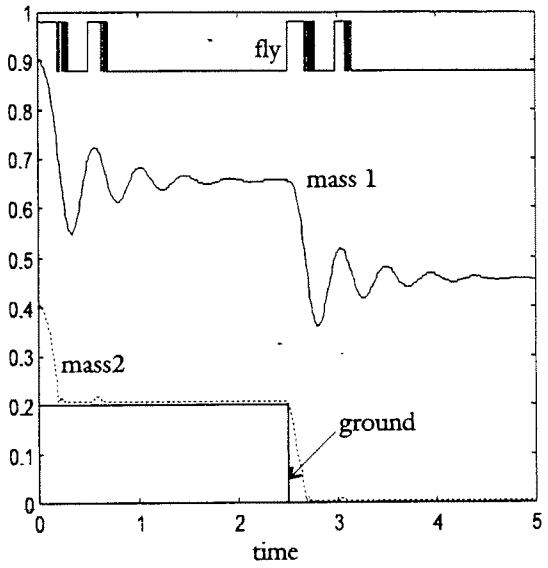


Figure 6: 'Jumping One-Wheel' going downstairs

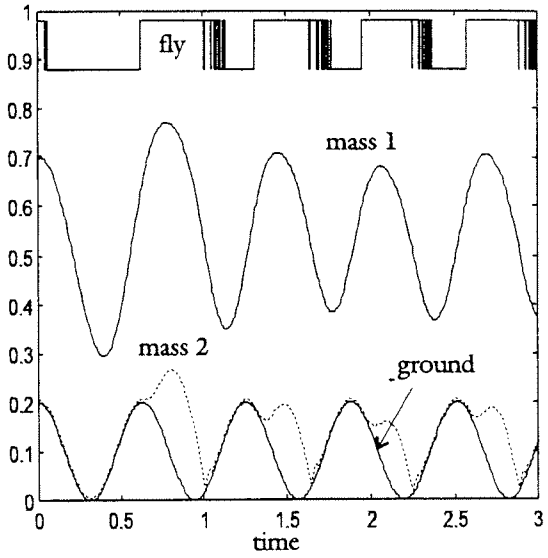


Figure 7: 'Jumping One-Wheel' driving on a 'sinusoidal gravel path'

```
1   BLOCK JFEDMAS;
2   $ BLOCKDIM -1 -1 6 16
3   $ TEXT "One Wheel 1"
4   $ TEXTALIG 1 1 1 1
5   $ ICONNAME "*** NO PIC ***"
6   $ ICONDIM 4 4
7   $ ICONALIG 1 1 1
8   INPUT
9      xg "[m] position of ground";
10     vg "[m/s] velocity of ground";
11  OUTPUT
12     X1 "[m] position mass 1";
13     X2 "[m] position mass 2";
14  PARAMETER
15     Ax1 = 0.705 "[m] X1(0)";
16     Ax2 = 0.205 "[m] X2(0)";
17     m1 = 50 "[kg] = upper mass 1";
18     m2 = 10 "[kg] lower mass 2";
19     k = 5e3 "[N/m] spring between";
20     d = 200 "[Ns/m] spring damping";
21     l = 0.5 "[m] spring length";
22     kg = 3000e3 "[N/m] spring to ground";
23     dg = 500 "[Ns/m] spring - ground damping";
24     r = 0.005 "[m] c.o.grav. mass 2 - ground";
25     g = 9.81 "[m/s^2] acc. due to gravity";
26  STATE
27     x1 "position mass 1";
28     v1 "velocity mass 1";
29     x2 "position mass 2";
30     v2 "velocity mass 2";
31  VAR
32     dist; a1; a2; a3; a11; a21; f;
33     state DISCRETE "ground - air";
34     fly DISCRETE;
35  SIM
36     x1 .= v1;
37     x2 .= v2;
38     a1 = -g -d/m1*(v1-v2);
39     a2 = -g +d/m2*(v1-v2);
40     dist = x1 - x2 - l;
41     a11 = k/m1*dist;
42     a21 = k/m2*dist;
43     v1 .= a1 -a11;
44     v2 .= a2 +a11 + f/m2;
45     SWITCH state
46     CASE 0:
47        // standing on the ground
48        f = kg*(r-x2+xg) + dg*(vg-v2);
49        ONRISE x2-xg > r DO jump;
50     CASE 1:
51        // up in the air
52        f = 0;
53        ONRISE x2-xg < r DO stand;
54     ENDSWITCH
55     X1 = x1;
56     X2 = x2;
57  ENDSIM
58
59  PROCEDURE jump;
60     state := 1;
61     fly := 0.98
62     STORE ALL;
63  ENDPROCEDURE
64
65  PROCEDURE stand;
66     state := 0;
67     fly := 0.88;
68     STORE ALL;
69  ENDPROCEDURE
70
71  INIT
72     state := 0;
73     fly := 0.88;
74     x1 := Ax1;
75     x2 := Ax2;
76  ENDINIT
77  ENDBLOCK JFEDMAS;
```

Figure 8: Blockscript

Intelligent Simulation Interface for LATISS Simulation System

Yuri Merkuryev, Galina Merkuryeva, Ainars Mazversitis, Agris Goldmanis
Dept. of Modelling and Simulation, Riga Technical University
1, Kalku Street, LV-1658 Riga, Latvia
E-mail: merkur@itl.rtu.lv

The paper discusses design of the Intelligent Simulation Interface (ISI), being used within a user-friendly intelligent discrete-event simulation system LATISS (LATvian Intelligent Simulation System). LATISS is a version of the generic interactive system for modelling and simulation GISMOS (Vangheluwe et.al., 1994). It is based on the same approach, when knowledge regarding a system under simulation is encapsulated in models. These models are organized in accordance with the MSL (Model Specification Language) formalisms. The main difference between these simulation systems is that in GISMOS all knowledge about the simulation process itself is also represented in the form of models. On the contrary, in LATISS only a part of knowledge about the simulation process is encapsulated in models (namely, knowledge about optimization algorithms to tune parameters of simulation models). The simulation process here is controlled by ISI, which supports intelligent abilities of the simulation system as well. Another essential part of LATISS is an object-oriented model base, where all models are kept. Simulation models are developed in the HGPSS simulation language, for both UNIX and LINUX platforms.

1. Abilities of simulation interface and its programming

The Intelligent Simulation Interface controls all processes in LATISS. Following are main functions, supported by ISI:

1. **The control function.** ISI supports control of the simulation process and also supports decision making, necessary to realize stages of the simulation process, e.g.

- **control of the simulation process**, aiming to achieve goals of simulation, specified by the user (e.g., analysis of a system under simulation, sensitivity analysis, optimization of system parameters). It is based on schemes of the simulation procedure, corresponding to various goals of simulation;
- **support of decision making,** accompanying realization of the simulation procedure. It is aimed to ensure a possibility to perform most of simulation stages automatically, thus not asking from the user for a deep knowledge in the field of the simulation theory. As a result, LATISS could be used by domain specialists, who are experts in their professional areas, but are not experienced in simulation studies. In order to support necessary statistical procedures, an interface with standard statistical software tools is foreseen. It is aimed to support such stages as analysis of input data (analysis of outliers, design of histograms, testing for fitness to various probability distributions), strategic design of simulation experiments (fractional factorial design, Plackett-Burman design), tactical design of simulation experiments (correlation analysis, evaluation of the necessary number of simulation runs), analysis of simulation results (design of confidence intervals, comparison of outputs from alternative designs). A possibility for the user to change the decisions, made by ISI, is provided as well.

2. **The advising function.** ISI supports the user in making decisions, concerning "manual" implementation of stages of the simulation procedure (for example, when implementing actions, which could not be performed automatically). It provides the user with necessary information, thus

advising him in making decisions. For instance, ISI provides the user with recommendations on use of various optimization algorithms in different situations.

3. **The explanation function.** The user can turn to ISI, asking it to explain him its decisions and recommendations. For instance, if ISI recommends to use some already existing models, which are kept in the model base, in order to model parts of the system to be simulated, the user can ask it to ground these recommendations.

4. **The learning function.** ISI is capable to learn from its own experience, generating new knowledge - corresponding facts and rules to be used for future decision makings. For instance, it collects information about effectiveness of optimization algorithms in various situations; when some of already observed situations occur, those algorithms, which have been found as the most efficient ones, are recommended.

All these functions are based on manipulations with knowledge, which include acquisition of knowledge, its normalization and application, and generation of new knowledge as well. Knowledge in ISI is organized in the following ways:

- **facts** which describe properties of objects, ISI is dealing with (e.g., properties of various algorithms to design simulation experiments: full factorial design, fractional factorial design and Plackett-Burman design);
- **rules** which describe rules, related to use of these objects (e.g., when to use each of the algorithms);
- **metarules** which are rules of a higher level, that can change existing rules or use them in order to develop new facts and rules;
- **comments** which are knowledge that can be treated by ISI and presented to the user in the form of comments.

Knowledge about MSL models is incorporated in LATISS also in the model base, in the form of its *Rule* elements. ISI manipulates with knowledge by means of **tools** which are procedures, that deal with facts and rules. Each tool consists of a PROLOG part that treats facts and rules, and of a C++ part that implements rules. Graphical ISI abilities are developed with C++ language using MOTIF library.

Currently, the research prototype of ISI is available. Its operation is illustrated below for the case study, when a manufacturing system with 4 working stations and a transport robot is simulated with a goal to find an optimal capacity for each working station.

2. Examples of ISI interactions

Building of a model is supported mainly by three windows illustrated in fig. 1. The first window is aimed to select the useful atomic models. A corresponding Class name, Application area and a Filter may be used to simplify the search procedure. Selected atomic models are presented with icons in the second window. It defines the model's structure and is used to form the coupled model's USE part. Working algorithm describes the sequence of atomic models or their interaction. This information is used to formalize the DYNAMIC part of the coupled model. Building of new atomic models and their icons is supported by ISI. Additional logical models are introduced to describe complex working algorithms. Animation of corresponding manufacturing process is available in the middle window. ISI realizes also the following intelligent functions concerning the model base:

- design of requests in accordance with a current stage of the simulation procedure and existing restrictions. As a result of that requests, corresponding models are found in the model base (e.g., models of separate parts of the system under simulation) and used to make the overall model of the system, and to run simulation;
- optimization of the model base (e.g., extracting unfitted models).

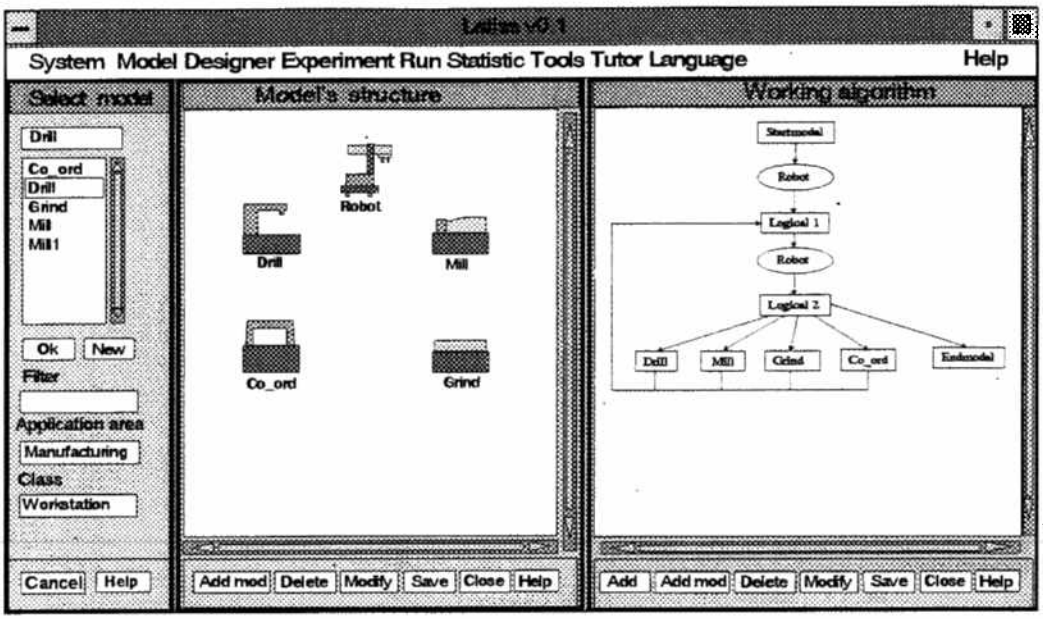


Fig. 1. Model building

ISI supports **specification of several simulation goals.** They are: G1- Process evaluation, G2- What If Analysis, G3- Sensitivity Analysis, G4- Bottleneck Analysis, G5- Comparison of alternatives, G6- Optimization of parameters, G7- Prediction, G8- Metamodelling, G9- Simulation Training. The goal of simulation may be specified by the user directly. Short description of each goal is provided. A special Goal Specification Inference procedure is developed as well. It is based on specification of each goal by a set of attributes. For example, the following G6 specification is used: a performance function exists and should be optimized, the corresponding model parameters should be founded. Goal specification by ISI means is illustrated in fig. 2,3.

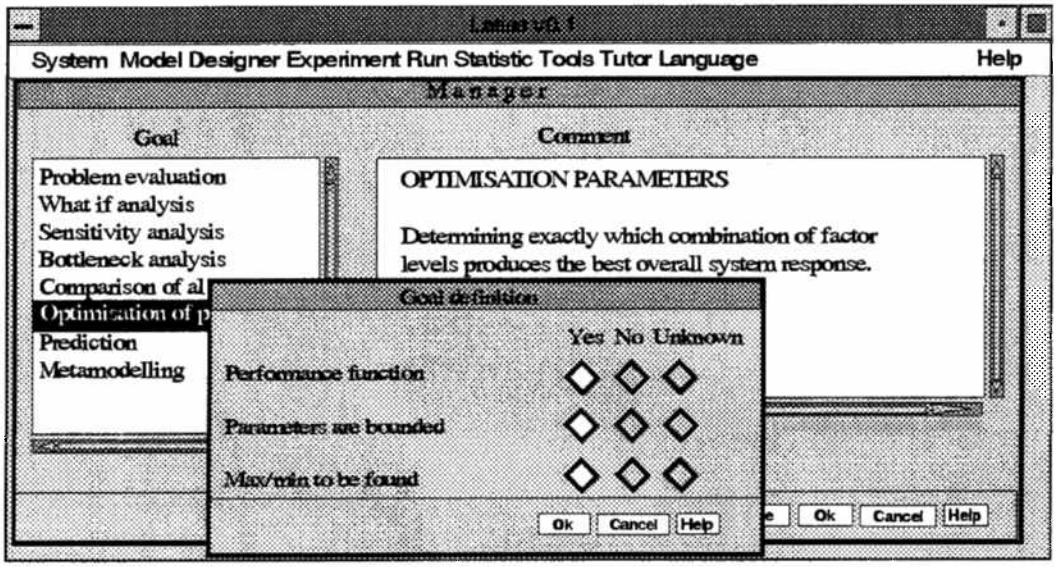


Fig. 2. Goal selection

Once the goal G6 is selected, the parameters of the model should be optimized. ISI incorporates knowledge and experience about 14 optimizers and their applications. The user initializes the characteristics of the model, e.g. number of parameters, their nature, etc. (see fig. 4). ISI makes selection based on knowledge inference, gets an available optimizer and presents explanations. The tool which realizes the corresponding optimizer is activated. Additional decision making procedure is provided if several optimizers are available.

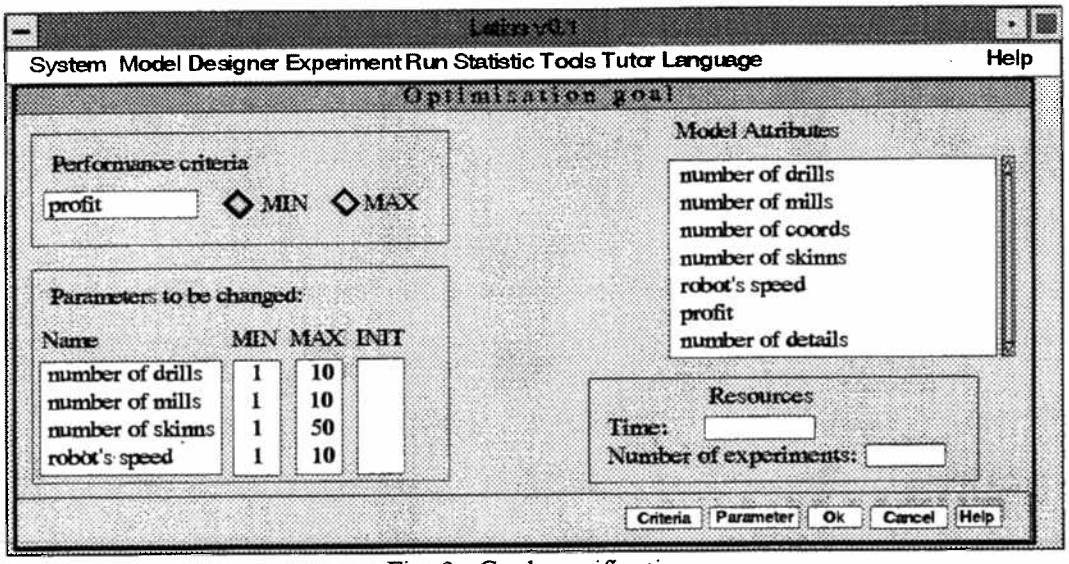


Fig. 3. Goal specification

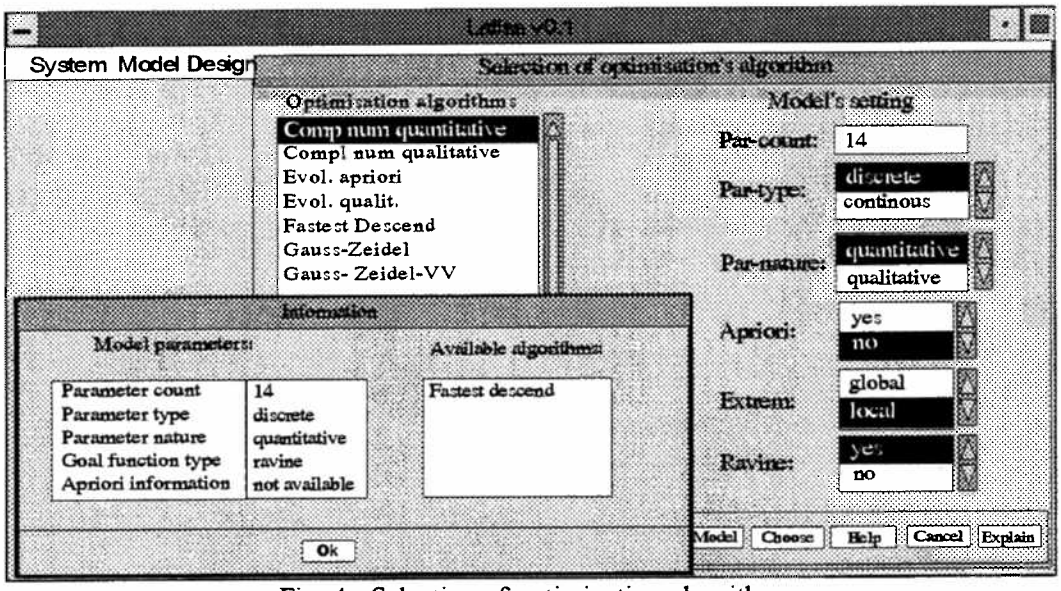


Fig. 4. Selection of optimization algorithm

The above described simulation interface incorporates knowledge and experiences. It is aimed to support intelligent abilities of the LATISS simulation system, e.g. goal searching, explaining and learning. By means of the ISI interface, LATISS operates in different ways adapted to the type of the user and the level of his experience. The non-experienced in simulation user will prefer to perform the most of simulation stages automatically. The most experienced one will prefer ''manual'' implementation of the simulation procedure and will ask for advises and recommendations in necessary cases. TUTOR tools to improve the own abilities of users of the LATISS simulation system are supposed to be developed as well.

H.Vangheluwe, G.Vansteenkiste, V.Visipkov, Y.Merkuryev, G.Merkuryeva and A.Teilans. (1994). Design of a user friendly modelling and simulation environment. Proc. of the 1994 European Simulation Multiconference. Ed. by Dr.A.Guasch and Dr.R.M.Huber.

ARCHITECTURE OF A SIMULATOR FOR RESEARCH IN ADVANCED TRAFFIC MANAGEMENT SYSTEM DESIGN

Richard M. Ingle

Georgia Tech Research Institute
Georgia Institute of Technology

ABSTRACT

Real-time traffic management is a key part of the overall vision of a future Intelligent Vehicle-Highway System (IVHS). The rapidly approaching IVHS era brings many technical challenges and questions related to the design, implementation, operation and maintenance of future Advanced Traffic Management Systems (ATMSs). This paper provides an overview of an Advanced Traffic Management System/Center Simulator developed at Georgia Tech and currently being used as a research tool to help answer questions that arise as IVHS technology makes its way into tomorrow's Advanced Traffic Management Systems. The architecture, technical challenges and fundamental implementation decisions are summarized. Brief descriptions of the traffic model, database, displays and associated simulated traffic video coverage are provided.

1 INTRODUCTION

The IVHS era brings high technology to the traffic management arena. For example, larger numbers of more capable sensors, automated control devices, automated monitoring devices, automated dispatching systems, advanced information systems, predictive traffic modeling systems, and various other support systems will incrementally find their way into Traffic Management Systems (Vostrez et al. 1992). The rapidly approaching IVHS era also brings many technical challenges and questions related to the design, implementation, operation and maintenance of future Advanced Traffic Management Systems. In order to start addressing many of the issues related to future IVHS-era ATMSs, simulation of the full ATMS environment is indicated. Simulation can help answer questions related to types of sensors needed, appropriate sensor coverage, design of automated and semi-automated support systems, appropriate degree of automation, and various human factors issues associated with vehicular, highway and traffic management center (TMC) systems. In order to meet this need, especially for the case of research into human factors issues associated with the design of future traffic management centers, we have developed a TMC simulator and are currently using it to conduct an intensive program of such research. As the research program continues, the capability of the TMC simulator is continuing to evolve.

This paper presents an overview of the hardware and software architecture of the TMC simulator, a brief discussion of some of the technical challenges addressed, a survey of the implementation decisions made on the basis of an extensive trade study, and a summary of initial performance observations.

2 TECHNICAL CHALLENGES

Some of the major technical challenges encountered in the development of the TMC Simulator included satisfaction of the requirements for rapid reconfigurability, real-time operation, and responsive CCTV simulation. Some of the details related to meeting these challenges are included in the architecture discussion that follows. Special attention is given to the CCTV system simulation, since this proved to be the most interesting and challenging problem solved.

3 ARCHITECTURE OF THE TMC SIMULATOR

The TMC simulator provides real-time, interactive operation, and is rapidly reconfigurable to allow for a wide variety of user interfaces and displays, ATMS reconfiguration, a variety of control methods, and varying degrees of automation. It provides simulated TMC inputs including traffic and roadway sensors, visual CCTV sensors, probe vehicles, cellular

telephone dialog, voice communication systems, and database services. It provides simulated TMC outputs associated with intersection control devices and algorithms, roadway access devices and control algorithms, variable message signs, highway advisory radio, commercial TV and radio, cable TV traffic channel, traffic bulletin board, and voice communication output. Operator support systems include adaptive traffic control, predictive traffic modeling, incident detection and location, response advisory, and information dissemination systems. The traffic model (AUTOS) was developed at Georgia Tech and is based on the Greenshield's speed-density flow model. The CCTV traffic video simulation (AUTOGRAPH) was also developed at Georgia Tech and is hosted on a Silicon Graphics Onyx / Reality Engine 2 system, making heavy use of the graphics-rendering hardware on this machine.

3.1 TMC SIMULATOR HARDWARE

After conducting detailed requirements and performance studies, the hardware architecture selected for hosting simulator included Silicon Graphics Indigo2 XZ machines for the four Operators' Workstations, the Experimenter's Workstation, the Traffic Model Server, and the Large Display Server. As discussed earlier, a Silicon Graphics Onyx / Reality Engine 2 was used as host machine for the simulated traffic video system. An additional bank of video monitors, whose configuration varies with experiments, uses Sony Trinitron PVM-1350 color video monitors. A Barco Retro Graphics 801 serves as Large Display and a plethora of video switching and converting devices are used in the implementation of a flexible video assignment and display system. The Operators' and Experimenter's Stations are rounded out with 486 PCs equipped with resistive membrane touchscreens which are configured as control panels and displays. These are generally associated with communications interfaces. An audio communications network with wireless headsets, Metamorphosis adjustable desks and Hag ergonomic task chairs complete the hardware used to implement the simulator.

3.2 TMC SIMULATOR SOFTWARE

Rapid software reconfigurability, including user interfaces, databases, displays, control devices, etc., was achieved by use of a unified TMC database and related control processes to drive the traffic model, the video simulator and all operator displays, and support system inputs and outputs provided the setting for achieving flexible software reconfigurability. The development of a script language, based on the public domain tool command language (Tcl) from the Department of Electrical Engineering and Computer Science at the University of California - Berkeley, for use in prescribing scenarios and configurations and the use of user interface builder tools, such as VAPS (a Virtual APplicationS development tool sold by Virtual Prototypes, Inc.), contributed heavily to flexible user interface reconfigurability.

An extensive trade study (Ingle et al. 1993) was done early in the simulator development program to locate and evaluate available commercial and government hardware and software for possible use in meeting various requirements of the simulator. None of the existing commercial or government traffic models met the interaction and real-time operation requirements, let alone ease of modification to allow for simulated support systems, etc. We opted, therefore, to start from an existing Georgia Tech traffic model called TERMINUS (Gilmore et al. 1992). Starting from this model, we created a much larger capacity, versatile model with hooks in place for the needed support systems. The resulting traffic model, called AUTOS, is a macroscopic flow model capable of running 5-10 times faster than realtime on a 5000-link network and 50 times faster than realtime on a 1700-link network. This performance permits us to use a copy of the model to achieve the simulated predictive traffic management support system. The AUTOS model is implemented using C++ object-oriented design. It is driven by TMCS database map data and traffic specifications. AUTOS (Gilmore et al. 1994) provides a macroscopic model of traffic flow using Greenshield's speed-density flow model:

$$u = u_f(1 - k/k_j)$$

$$q = uk,$$

where q is flow, u is speed, k is density, u_f is free flow speed and k_j is the jam density. AUTOS models traffic flow on both freeways and surface streets and allows for various types of intersection control (uncontrolled, signed, fixed signal, actuated signal, fixed meter, actuated meter), signal phases, turning percentages, delays and various parameters of link control (directionality, number of lanes, length, free-flow speed, road condition factor, initial load, load deltas for rush hour build-up/down, current and maximum vehicle counts, etc.). In addition, parking lots, high occupancy vehicle lanes, reversible lanes and turn lanes are modeled.

Perhaps the biggest technical challenge was achieving a responsive simulated traffic surveillance video capability. Since our interviews with traffic management experts all over the country consistently underscored the importance of traffic surveillance video in present and future TMCs, we elected to implement high fidelity traffic video simulation using computer animation. The traffic video simulation was implemented by taking 360° photographs at 38 key traffic locations in Atlanta. These photographs were digitized, existing traffic was removed, background and foreground images were separated, lane paths were defined, and animated traffic using texture-mapped graphical techniques were applied to produce animated traffic flowing between the background and foreground planes. We were able to achieve a very realistic animation, with densities, speeds and vehicle-type mix controllable by the traffic model and model control processes. Incident scenarios are easily choreographed and emergency vehicles (police, ambulance, tow trucks, etc.) arrive in accordance with dispatching actions. Where other visual confirmations are possible, such as variable message signs, these features are also depicted in the traffic video simulation. Smooth lane changes are also rendered. Some atmospherics are implemented, although this aspect has been at low priority to date. Many sites can be concurrently ready for display at any time and up to four sites can be concurrently rendered for viewing on any number and combination of monitors, video windows, or on the large screen display. The operators have full control of camera selection, 360° pan and zoom (1X to 6X). This traffic video simulation, called AUTOGRAPH, integrated with the traffic model has already attracted attention from traffic engineers, managers and researchers from many countries and has led to requests to extend the capability to include additional features, such as tunnels and bridges. Figure 1 provides a top-level depiction of the software architecture.

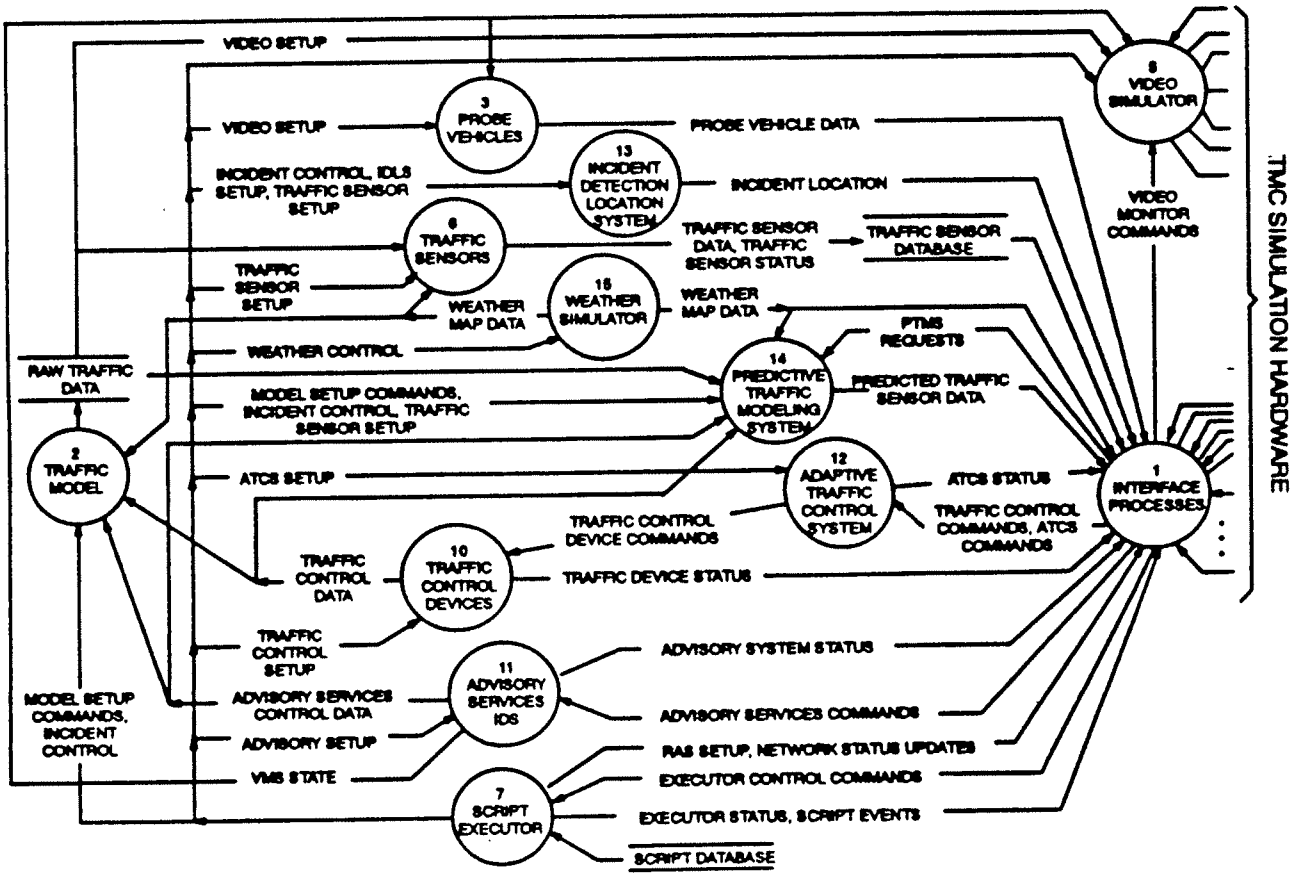


Figure 1: TMC Simulator Software Interfaces

4 SUMMARY AND RESULTS

A very capable ATMS Simulator has been implemented and is currently being used in a research program investigating human factors issues in the design of future TMCs. Simulation capabilities will continue to evolve over the next year and the simulator should be useful for a wide range of ATMS research and training purposes. The traffic model, AUTOS, is capable of running much faster than realtime for network sizes up to 5000 links. A sophisticated Traffic Video Simulator, AUTOGRAPH, is integrated with AUTOS and produces very realistic simulated traffic video animations. AUTOGRAPH taxes the current state of the art in graphical simulation to render four concurrent video simulations. It is anticipated that the next generation in Silicon Graphics compute power will allow this limit to expand.

ACKNOWLEDGMENTS

This work is sponsored by the Federal Highway Administration under Contract DTFH61-92-C-00094. Ms. Nazemeh Sobhi is the FHWA Contracting Officer's Technical Representative. Dr. Richard Ingle is director of the simulator design and development effort. Dr. Michael Kelly is principal investigator for the program of human factors research being performed using the traffic simulator as a tool. *Notice -- This document is disseminated under sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its content or use thereof. The United States Government does not endorse products or manufacturers. Trademarks or manufacturer's names appear herein only because they are considered essential to the object of the document. The contents of this report reflect the views of the author and project team who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official policy of the Department of Transportation*

REFERENCES

Gilmore, J. F., K. J. Elibiary, and H. Forbes. 1994. Intelligent Control in Traffic Management. To appear in *Proceedings of the International Conference on Vehicle Navigation and Information Systems*. August 31 - September 3, 1994 Nagoya, Japan.

Gilmore, J. F., K. J. Elibiary, and R. J. Peterson. 1992. A Neural Network Approach to Special Event Traffic Management In *Proceedings of the National Conference on Advanced Technologies in Public Transportation*. Transportation Research Board, San Francisco, California.

Ingle, R. M., S. M. Blankenship, D. J. Folds, J. M. Gerth, M. J. Kelly, D. R. Stocks, P. D. West, C. J. Whaley, and B. D. Williams. 1993. Traffic Management Center Simulator Trade Study. Technical report on contract DTF61-92-C-00094, Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, Georgia.

Ingle, R. M., N. Sobhi, and B. D. Williams. 1994. An Advanced Traffic Management System Simulator for Intelligent Vehicle-Highway Systems Research. In *Proceedings of the 1994 Winter Simulation Conference*. December 11-14, 1994. Orlando, Florida. pp. 1455-1460.

Vostrez, J., J. Arlook, E. S. Greene, D. C. Roberts, M. G. Sheldrick, and J. Sussman. 1992. *Strategic Plan for Intelligent Vehicle-Highway Systems in the United States*. Washington, D. C.: Intelligent Vehicle-Highway Society of America.

AUTHOR BIOGRAPHY

RICHARD M. INGLE is a Principal Research Scientist and head of the Systems Technology Branch of the Georgia Tech Research Institute's Electronic Systems Laboratory. He received a B.A. degree in mathematics from West Georgia College in 1967, and M.S. and Ph.D. degrees in applied mathematics from the Georgia Institute of Technology in 1969 and 1980 respectively. His research interests lie in computational aspects of modeling and simulation, and applied mathematical analysis. His career has intertwined teaching with applied research. He has also developed various other simulation and display systems in current use by the USAF and NASA.

MAPLIS - Simulation with Work Sheets in Hyper Space

Wilfried Tettweiler, Krailling

Summary: Recent releases of Lotus 1-2-3, QuattroPro for Windows and MS-Excel have shown considerable enhancements of these products, supporting three-dimensional spread sheets, "What-when"-analysis and spread sheet simulations etc.. New elements in the MAPLIS PC release reflect these developments by means of interfaces to access data of different file formats for popular PC software, supporting multi-dimensional views on the data contained, analysis, consolidation, aggregation (sums) and application of formulas to this data. Formulas and constant data from work sheets can be converted into MAPLIS-language models. Multi-dimensional computations can be vectorized to a high degree due to the tensor concept of MAPLIS. This permits high performance computing with a vector processor on an upgraded personal computer.

1 Introduction

Once more it is questionable if simulation with spread sheets means the birth of a good idea or the death of a bad one. Since Microsoft's Excel introducing "what-when"-analysis created innumerable spread sheet models, there is growing concern about the missing clock mechanism and the "intuitive" use of the Solver to manipulate categorical data. For example: interest rates (no bank pays 6.32789% interest on your money) or numbers of children (do you have 0.5 children?). The methodical search for a solution using MAPLIS language which is based on a tensor concept to store multidimensional objects in the model's data base aims at avoiding that sort of conflict resulting from mere computing with scalar data. The following printout shows the first part of the contents of a MAPLIS variable named EAGNWDC with six dimensions ZELLE, GEBALT, WGN, NATION, ALTER and GESCHL:

```
(ZELLE    )
(GEBALT   )
(WGN      )
(NATION   )
        --> ALTER
    |   +------------------+
    |   | EAGNWDC          |
    V   +------------------+
   GESCHL

(1    )
(B8889    )
(EINFM    )
(DEUTSCH)                    (AUSLAND)

        1         2                  1         2
       +---------+---------+        +---------+---------+
MAENNL |         1         2| MAENNL |         5         6|
WEIBL  |         3         4| WEIBL  |         7         8|
       +---------+---------+        +---------+---------+

(1    )
(B8889    )
(GESCH    )
(DEUTSCH)                    (AUSLAND)
        1         2                  1         2
       +---------+---------+        +---------+---------+
MAENNL |         9        10| MAENNL |        13        14|
WEIBL  |        11         0| WEIBL  |        15        16|
       +---------+---------+        +---------+---------+
```

Header of the Printout with names of the dimensions used

First two values for the dimension WGN

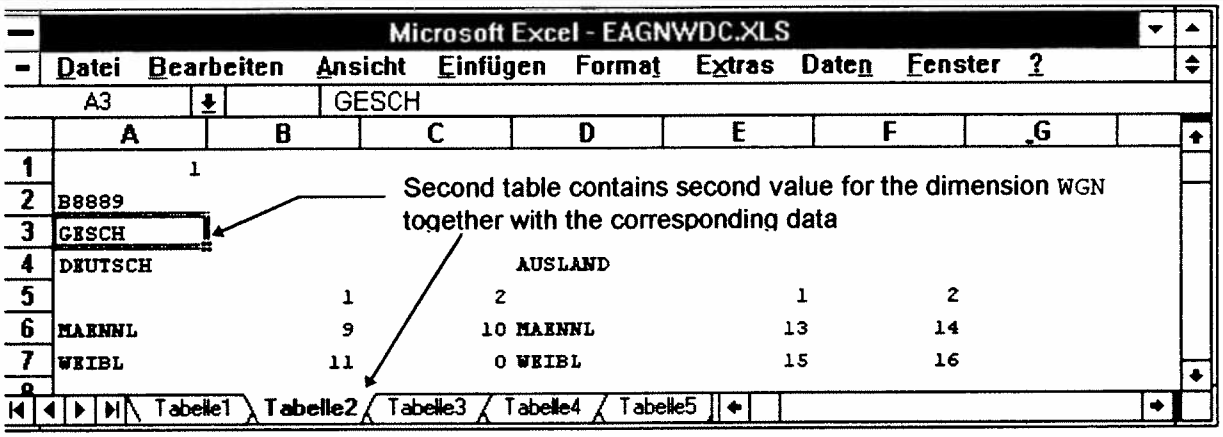
Furthermore MAPLIS gets welcome help from newly emerging trends in the development of spread sheet programs.

2 New characteristics of standard spread sheet programs

Recent releases of Lotus 1-2-3, QuattroPro 5.0 for Windows and MS-Excel have shown considerable enhancements of these products.

2.1 Supporting three-dimensional spread sheets

Seeing multi-dimensional tables being reflected as a sequence of two-dimensional tables is routine for users who are familiar with standard statistical packages like SPSS etc. Likely, one can prepare multi-dimensional data in appropriate sets of spread sheets. Any practical minded user can easily understand and handle those, because they meet his view of data categorially rather than continuously. The following example corresponds to the MAPLIS variable shown above, the upper left corner of each worksheet contains the respective value settings for the first three dimensions `ZELLE`, `GEBALT` and `WGN` (the actual value of `WGN` is `GESCH`):



Each cell `B6..C7` and `E6..F7` contain frequencies of units with idetical properties. Cells `A1..A7`, `D4..D7` and `B5..F5` define values for the corresponding properties.

Selective printout of a MAPLIS variable with any value combination (as wanted, not as predefined by the order or structure of the spreadsheet) serve to analyse and consolidate data. Aggregation level changements eliminate (or exchange) dimensions by computing row or column sums or vice versa add dimensions by distributing data according to given probabilities.

In addition a sequence of spread sheets is the most natural way to reflect also periodical alterations of data. The clock mechanism of MAPLIS is as period-oriented as DYNAMO, its predefined variable `TIME` allows for the saving of sets of variables according to their occurence in time in sets of spreadsheets. Support of three-dimensional spreadsheets in standard spreadsheet packages provides easy access to variable's contents being reported in the form of a time series.

2.2 "What-when"-analysis and spread sheet simulations

New functions as e.g. scenario-manager, if-then-else-analysis and break-even-analysis etc. have set standards for modelling not only in the area of financial calculation or business management. The problem connected is that the search for the optimum may bring forth only relative optima or even worse, rather unrealistic results (see introduction). MAPLIS-language performs strict compatibility checking to identify and avoid problems undetected in mere spreadsheet computing, e.g. multiplication of interest rates with number of houses.

3 New elements in the MAPLIS release for personal computers

MAPLIS can offer solutions that are adequate to the problems mentioned above.

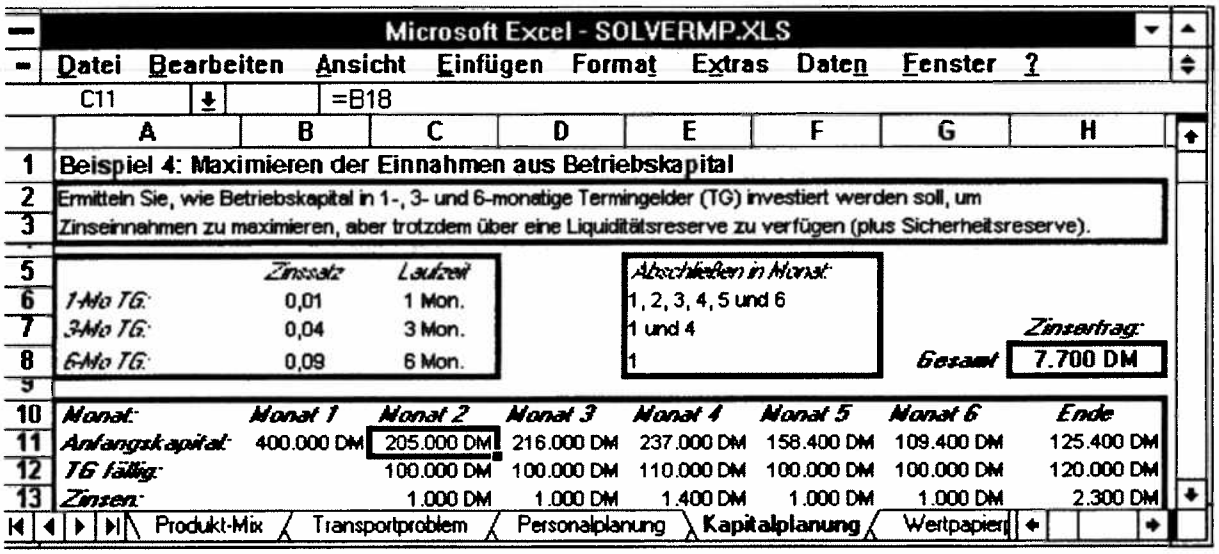
3.1 Interfaces within MAPLIS to process standard data formats

The MAPLIS command `INOUT_MEDIUM` now accepts de facto standards like ODBC and IDAPI to process data immediately from data bases and spread sheets. This allows for the collection of data for processing even from quite different sources. This data can then be examined and connected logically or arithmetically under any aspect within the multi-dimensional concept of MAPLIS.

3.2 Integrating spread sheet models into MAPLIS

The formulas and constant data from work sheets can be converted into MAPLIS-language models either directly or through filter programs.

The following Excel Solver spreadsheet



is translated into a model in MAPLIS language as follows:

```
add_model solvermp                          <-- typical definition of a scalar dimension
n_of_values EINS (1)
-- A1: 'Beispiel 4: Maximieren der Einnahmen aus Betriebskapital
-- A2: 'Ermitteln Sie, wie Betriebskapital in 1-, 3- und 6-monatige Terminge
-- A3: 'Zinseinnahmen zu maximieren, aber trotzdem über eine Liquiditätsrese
-- B5: ^Zinssatz
-- C5: ^Laufzeit
-- E5: 'Abschließen in Monat:
-- A6: +"1-Mo TG:"
-- B6: 0.01
...
-- G10: 'Monat 6
-- A11: 'Anfangskapital:
add_variable   B11 (EINS)
print_formats  B11 (16.9)
let            B11 = 400000
add_variable   C11 (EINS)
print_formats  C11 (16.9)
compute        C11 = +B18
...
end_model
```

Strings are translated to comments

Declaration and initialization of the variable corresponding to cell `B11`

Declaration and computation of the variable corresponding to cell `C11`

The same work sheet can contain the model as well as the data related to it. It is thus possible to perform a plausibility test on a model first by means of the spread sheet program based on scalar data. One can then continue processing multi-dimensional data with the corresponding MAPLIS-language based task after having changed the MAPLIS variable definitions to multiple dimensioning, e.g. . `add_variable EAGNWDC (ZELLE by GEBALT by WGN by NATION by ALTER by GESCHL by TIME)` according to the example above.

3.3 Benefits using a vector processor

Computations can be vectorized to a high degree due to the tensor concept of MAPLIS. This permits high performance computing with a vector processor on an upgraded personal computer.

4 State of development and preview

The MAPLIS-language was first published 1982 in german language in the proceedings of the "1. Symposium Simulationstechnik" (1st Symposium for Simulation Technique, Informatik-Fachberichte 56, Springer 1982) and one year later in english language in the proceedings of the "Mathematical Modelling in Science and Technology (Pergamon Press 1983). Various realization platforms since that time were main frames (IBM, Cyber and CRAY). A number of applications based on MAPLIS models are in the range from population development prognosis to investment planning.

A personal computer DOS version of MAPLIS has been in use since the beginning of 1994, a Windows user interface is in work. All parts of MAPLIS-computations, which can be vectorized or parallelized respectively, will soon be speeded up by a ZORAN vector processor and by means of INMOS transputers.

Wilfried Tettweiler, Brahmsstr. 12, D-82152 Krailling
Tel.: +49-89-8573777 Fax: 8572183
E-Mail: 100421.2304@CompuServe.com

VIBRATORY ANALYSIS OF THIN SHELL STRUCTURES IN MEDIUM AND HIGH FREQUENCY RANGE WITH THE ACTU SOFTWARE

PEYRAUT FRANÇOIS

SCIENCES INDUSTRIES CONSEILS
14, Avenue de Sceaux
78 000 VERSAILLES
FRANCE
Tel. : (1) 39 49 05 27
Fax : (1) 39 02 77 23

1 - INTRODUCTION

This paper presents the ACTU software which is devoted to structural vibrations and sound emission analysis outside thin shells structures. The sound can be generated by rotating machine's vibrations (pump, compressor, motor ...). These machines create periodic motions which propagate as sound waves in interior and exterior fluids and as vibratoring waves in the shells.

The quasi-analytic method used in ACTU allows low, medium and high frequency's computation. In addition, this method don't impose any restrictions on boundary conditions.

At the present time, ACTU works in vibratory field for cylindrical shells and axisymetrical loading. All the boundary conditions are allowed.

ACTU has been developed in fortran 77.

2 - THEORY

ACTU applies to thin shells. Therefore, we can use the Kirchhoff-Love's theory in which a little piece of transverse matter is considered as a rigid body. Thus we neglect normal striction and transverse shear strains ([4]).

With cylindrical shells and axisymetrical loading, the longitudinal displacement u, the azimuthal displacement v and the radial displacement w are solutions of the following partial derivatives equations :

$$\delta^2u/\delta z^2 + k^2u + \mu/R\ \delta w/\delta z = 0 \qquad (1)$$

$$(1-\mu)/2\ \delta^2v/\delta z^2 + k^2v = 0 \qquad (2)$$

$$\mu/R\ \delta w/\delta z + h^2/12\ \delta^4w/\delta z^4 + (k^2+1/R^2)w + \mu/R\ \delta u/\delta z = \alpha\ \delta p \qquad (3)$$

δp represents the pression applied to the shells and z the longitudinal variable along the cylinder's axis. R is the radius and h the thickness. We have :

$$k = \Omega[m_v(1-\mu^2)/E]^{1/2} \quad (4)$$

with k the wave number, Ω the circular frequency, m_v the density, μ the Poisson's coefficient and E the Young's modulus.

The coefficient α is defined by :

$$\alpha = (1-\mu^2)/(Eh) \quad (5)$$

The longitudinal and radial equations (1) and (3), dependant on each others by the terms $\delta w/\delta z$ and $\delta u/\delta z$, are solved by an analytical Fourier series's expansion and a plane wave's approach ([2] and [3]). In that case, the solutions of (1) and (3) are completely known except for the modal amplitudes. Consequently it's necessary to solve numerically the modal amplitude's linear system.

Moreover our method allows all boundary conditions in vibratory field.

3 - VALIDATION

ACTU has been correlated with a completely analytical method. This method assumes that longitudinal and radial vibrations are independant ([1]). On that hypothesis, the longitudinal resonant frequencies are given by :

$$f_k = k[E/m_v]^{1/2}/(2L) \quad (6)$$

where L is the cylinder's length.

Consider a cylinder with :

$$E=2.1E11\ N/m^2;\ m_v=7800\ kg/m^3;\ L=1\ m;\ R=0.25\ m;\ h=0.01\ m;\ \mu=0.285$$

By using equation (6) the third longitudinal resonant frequencies are :

$$f = 2594, 5188, 7782\ Hz \quad (7)$$

The comparison beetween (7) and ACTU's results is shown in table 1. The discrepansies don't exceed 3.9 %. Hence ACTU is valided in vibratory field.

Frequency (Hz)			
Analytic	2594	5188	7782
ACTU	2646	5363	8087
error	2 %	3.4 %	3.9 %

Table 1

In the same way, we have considered a cylinder with :

$$E=2.1E11\ N/m^2;\ m_v=7800\ kg/m^3;\ L=80\ m;\ R=0.25\ m;\ h=0.01\ m;\ \mu=0.285$$

By using again equation (6), we can easily find the fifth longitudinal resonant frequencies:

$$f = 32.4, 64.9, 97.3, 129.7, 162.1\ Hz$$

On figure 1, we have plotted the cylinder's dynamic response against frequency. It may be seen that analytical frequencies correspond exactly with ACTU computation.

A key has been put in ACTU for the amplitude's correlation. This key allows us to have input acceleration to zero. We can thus compare displacements and rotation computed by ACTU and statics calculation made with the finite element software I-DEAS. We have considered a cylinder with :

$$E=2.1E11\ N/m^2;\ m_v=7800\ kg/m^3;\ L=0.936\ m;\ R=0.25\ m;\ h=0.01\ m;\ \mu=0.285$$

We have assumed free rotation and radial displacement equal to zero at cylinder's ends and we have taken the following longitudinal displacement :

$$u = 0.1\ m \text{ at cylinder's top; } u = -0.1\ m \text{ at cylinder's bottom}$$

Observing figures 2, 3 and 4, it may be seen that displacements and rotation amplitudes are perfectly the same. Therefore, in addition of frequency's calculation, ACTU allows us to make statics computation. That's an important advantage for ACTU's users who want to test the structural resistance of a thin shell assembly.

4 - CONCLUSIONS AND FUTURE PROSPECTS

ACTU is helpful for silent installation's design (choices of material and geometry for instance). In consequence of its quasi-analytic approach, ACTU computes quickly, allowing engineers to work rapidly for a design project. Moreover, this approach is better than finite element for medium and high fréquencies and has no limitations for boundary conditions.

At the present time, a more complicated geometry than cylinder and a general loading are studied with an analytic approach to guarantee the ACTU's fast computation.

Finally, beyond the vibratory study, useful because a good acoustic behaviour needs a good vibratory approach, the acoustic emission's analysis is in progress.

REFERENCES

[1] "Modélisation des coques minces élastiques", Destuynder P., cours de l'Ecole Centrale, 1986/1987

[2] "Elastic Wave Transmission Through Plate/beam Junctions", Langley R.S. and Heron K.H., Journal of Sound and Vibrations, 143(2), p. 241-253, 1990

[3] "Comportement Vibratoire de Réseau de Plaques Assemblées", Rebillard E.et Guyader J.L., Journal de Physique IV, Colloque C5, supplément au Journal de Physique III, Volume 4, P. 121-124, 1994

[4] "Vibrations of Shells and Plates", Soedel W., Ed. Marcel Dekker (New-York), 1981

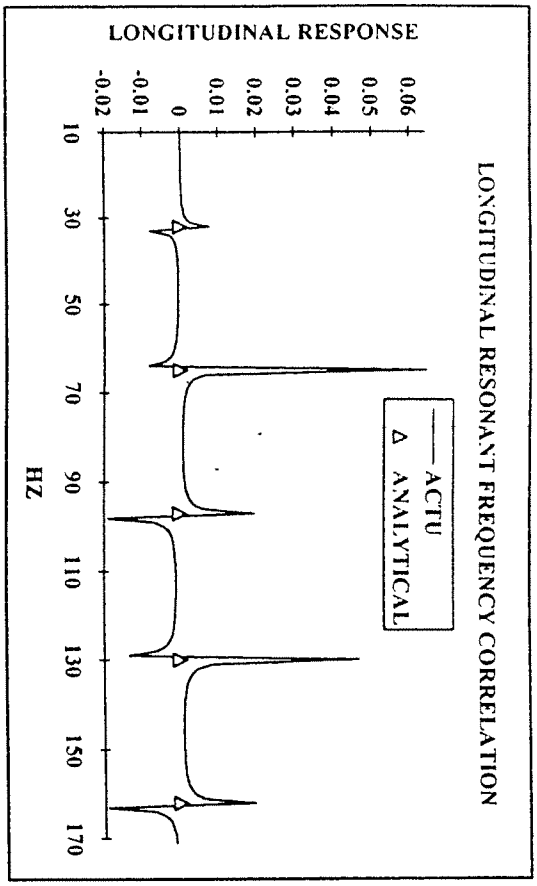


Figure 1

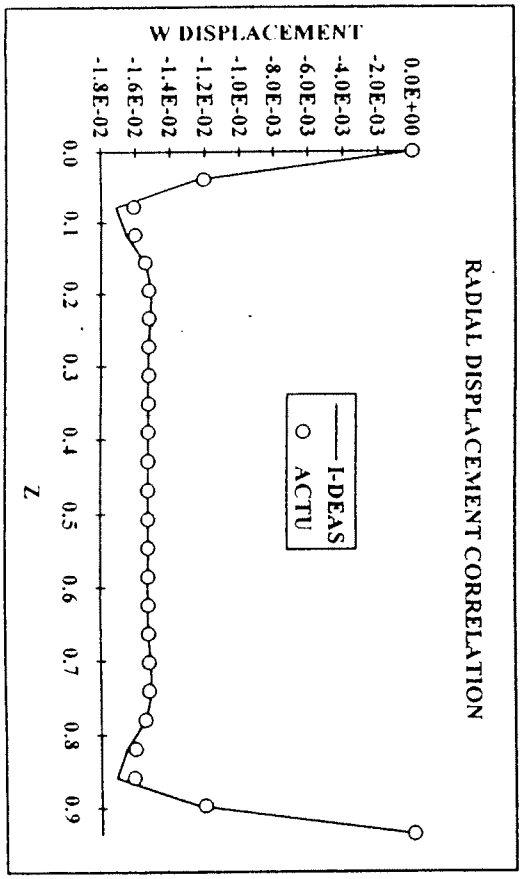


Figure 2

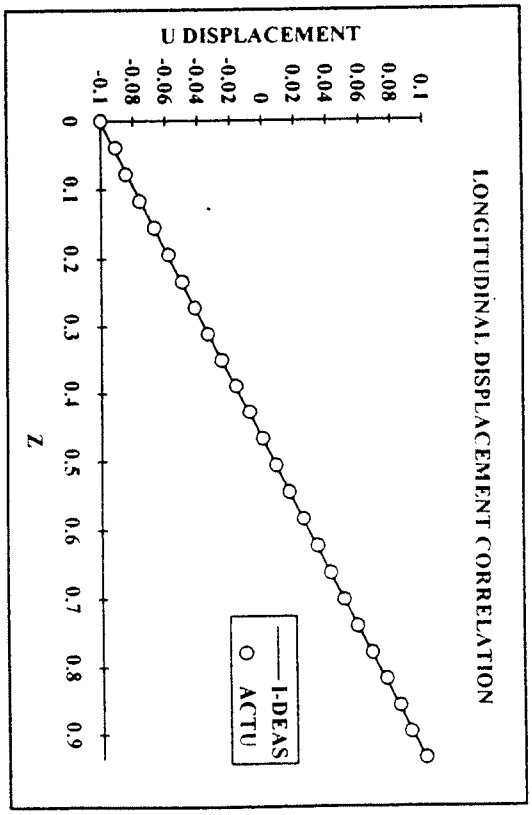


Figure 3

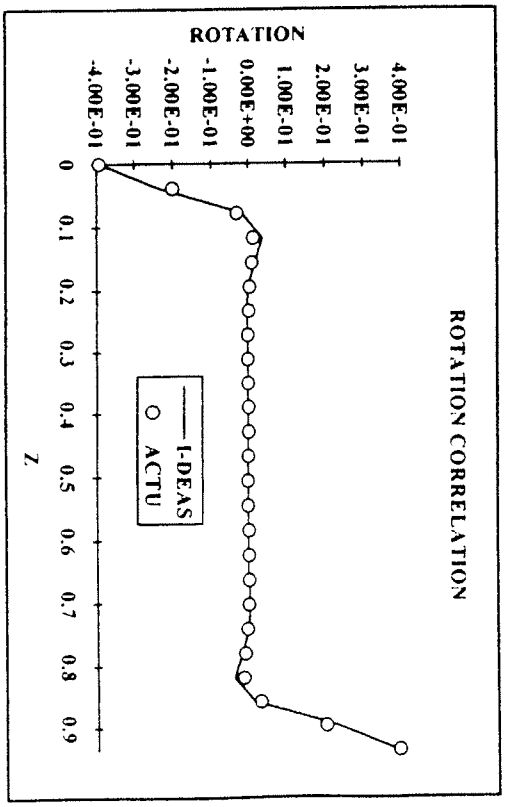


Figure 4

A Multi-Purpose Simulator for an Oil Production Plant

Ommund Øgård[1], Ivar Halvorsen[1], Kåre Telnes[2], and Dag Sjong[2].

ABSTRACT

This paper describes a multi-purpose dynamic simulator for the Heidrun oil production plant. The simulator integrates a commercial dynamic model with the actual process control system to enable dynamic simulation, control system verification and operator training within the same framework and based on the same models and configuration data. The control system configuration can be imported directly into the simulator. This will simplify simulator maintenance and ensure consistency between the simulator and the real plant. The entire system runs on standard Unix work stations. The simulator has so far been used for initial operator training, control system verification and controller tuning.

INTRODUCTION

The usage of dynamic simulation is rising in today's industry. The Norwegian offshore oil production industry uses dynamic simulation for different purposes. Typical examples are: design and verification of control structures, verification of the process design, testing of operational procedures, operator training and for verification of process modifications during the plant's life cycle.

There has so far been a clear distinction between simulators used for different engineering purposes and simulators used for operator training. "Engineering simulators" for the oil production process are based on detailed and accurate multi-component models. Engineering simulators are usually maintained and used during the plants life time. The OTISS simulator from the british company SAST dominates this market segment. Engineering simulators run on standard workstations and are moderately priced.

Training simulators, in contrast, are tailor made to match the training requirements and to mimic the operators' interaction with the control system. The models are traditionally different from those used in the "Engineering simulators", emphasising real time performance and robustness. Special HW might be required and the price is high. The deviating models in "Engineering" and training simulators increase the maintenance costs because two models must be updated when the plant undergoes modifications.

The rest of this paper will describe the HOPE (Heidrun OPerational Experience) simulator. The HOPE simulator integrates an OTISS simulator and Simrads's AIM-1000 process control system to enable operator training, control system verification and engineering studies on the same simulator. The HOPE development project consisted of two parts: the simulator and an operator support system for on-line execution of operational procedures. Woods, Backer, Wahl and Telnes (1994) describe this operator support system in detail.

PROJECT OBJECTIVES

The main objectives for the HOPE simulator are stated below.

1. To develop a multi purpose dynamic simulator for the Heidrun offshore oil production plant that enables both operator training and system verification.
2. The simulator should use the same configuration files as the control system and the Engineering simulator, in this case an AIM-1000 distributed process control system and an OTISS model.
3. The development costs should be lower than for a traditional training simulator.
4. The functionality should be the same as in other training simulators. Both instructor based and self training should be supported.
5. The simulator should run on standard work stations located in a normal office environment both on board and in the operation centre.

The scope of the HOPE simulator did only include the main process on the platform. This includes the systems for oil separation, gas production and produced water handling.

1. SINTEF Automatic Control, N-7034 Trondheim, Norway. (Ommund.Oegaard@regtek.sintef.no)
2. Statoil R&D Centre, N-7005 Trondheim, Norway. (ktel@fou.tr.statoil.no)

The requirements do not state that the training environment should be identical to the control room on the platform. More emphasis was put on making the simulator flexible and facilitate multiple usage. One rational for this is the fact that only experienced operators are recruited to the Heidrun platform. The focus of the training is therefore moved more on accustoming the operators to the process dynamics and the operational procedures, than on training elementary control room functions. A copy of the control room environment was therefore not important as long as the HCI was nearly equal to the HCI in the control room.

The basic training simulator functionality in the HOPE simulator include:

- An instructor Human Computer Interface (HCI) for control and operation of the simulator
- Selection and setting of initial conditions and scenarios
- Selection and setting of disturbances while the simulator is running
- Taking snapshots of the current state and saving initial conditions
- Go to a previous snapshot and replay from snapshots

TECHNICAL SOLUTIONS

The system architecture is illustrated in the data flow diagram in Figur 1. The re-used components are shaded. This includes nearly all of the AIM-1000 system, its internal communication protocol and the OTISS simulator. The most extensive SW components in the HOPE simulator are therefore re-used. Some enhancements had to be made in the basic AIM-1000 software to implement snapshots, replay and scenario functions. These modifications, however, do not interfere with the HOPE simulator's ability to import AIM-1000 configurations directly.

The Maritime Information Technology Standard (MiTS) protocol was used for all inter-process communication in the SW developed in the HOPE project. MiTS defines a application program interface that allows flexible network transparent inter-process communication. MiTS is based on TCP/IP and is implemented on different unix dialects, Windows and several real time operating systems. The HOPE simulator is thus designed to utilize several cooperating computers connected in a standard local area network. MiTS is described more in detail in Rødseth and Haaland (1993).

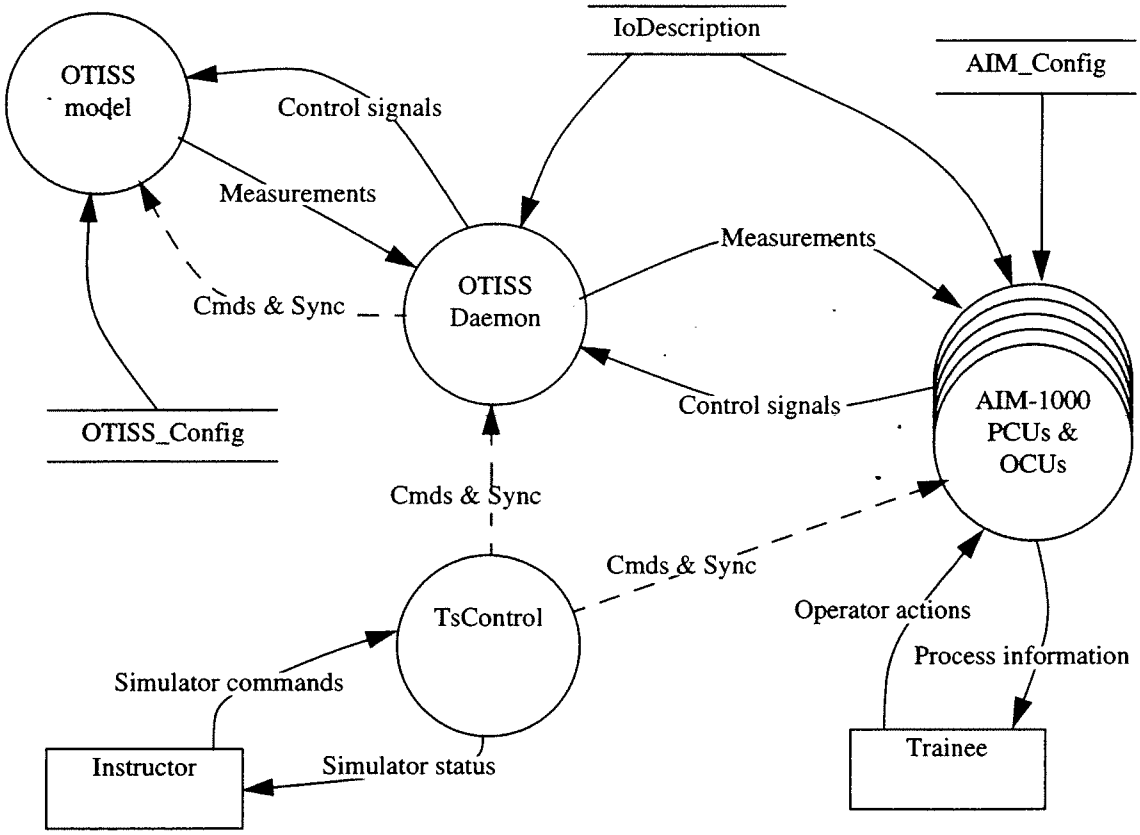
The following SW components in Figur 1 was developed in the HOPE project:

1. TsControl, the instructor HCI, which also synchronises the simulator to real time and coordinates the data exchange between AIM-1000 and OTISS.
2. The command protocol from TsControl to OTISS Daemon and AIM-1000, and the data exchange protocol between the OTISS Daemon and AIM-1000.
3. The OTISS daemon which implements the MiTS interface to SAST's proprietary internal protocol.
4. The IoDescription cross connection file defining the data transfer between OTISS and AIM-1000 by mapping OTISS and AIM-1000 tag names.
5. Some utility programs for setting up the IoDescription.

The key point that made the HOPE simulator feasible, was the ability to run the AIM-1000 system software nearly unmodified on Unix work stations. Each Process Control Unit (PCU) and Operator Control Station (OCU) executes as an independent Unix process. This allows flexible allocation of the PCUs and OCUs on the available computers.

The HOPE simulator runs on four standard HP9000/735 computers. The OTISS simulator and the OTISS Daemon run on one computer. TsControl and the 19 PCUs and 3 OCUs are distributed on the other three. These three are equipped with special AIM-1000 keyboards and are used as operator stations during training. This configuration runs in real time with a data exchange frequency at 1 Hz, which is the same as in the plant.

Fray (1995) distinguishes between emulated, stimulated and hybrid simulator architecture. The emulated models plant, control system and HCI. The stimulated architecture models the plant and uses control system HW for the rest, while the hybrid only uses control system HW for the HCI. In this sense the HOPE simulator is a completely software stimulated architecture where the AIM-1000 system's ability to run under Unix, saves the costs and complexity associated with stimulated solutions using control system HW.



Figur 1 The HOPE simulator architecture

The selected technical solution allows a new control system configuration to be imported directly from the process control system's data base to the HOPE simulator. Only minor modifications have to be carried out in the model to reflect a change in the control system configuration. If for example a new transmitter tag has been added to the control system, a corresponding transmitter module has to be configured into the model, and an extra line has to be inserted into the IoDescription file. This new entry in the IoDescription file is needed to link the measurement from the transmitter in the model to the input of the new transmitter module in control system.

The OTISS model may also run completely disconnected from the HOPE simulator using its own internally modelled control system. This is achieved by a set of SW switches that allow the user to select whether control signals are to be fetched from the OTISS-Daemon or from the internal controllers. OTISS is used as a stand alone process simulator if the internal controllers are enabled. Consequently, only one model has to be maintained for the Heidrun plant.

THE DEVELOPMENT PROJECT

The HOPE development project consisted of three main phases. First, software development. Secondly, simulator configuration and thirdly simulator verification. We used standard structured technics as described in Yourdon (1989) for system analysis and design. The programming was done in ANSI C. We used the POSIX API to access the operating system, OSF/Motif to build the HCI and MiTS for all inter-process communication. About 4.5 man year were spent in this phase. All the software was developed in this phase, but the test configuration did only include a small number of AIM-1000 and OTISS tags.

The goal for the configuration phase therefore was to establish the full simulator. The main activity in this phase was to establish and test all the cross connections between the full OTISS model and the AIM-1000 configuration and to set up static default values for AIM-1000 modules which did not have a counterpart in the OTISS model. This task involved extensive search for, and pairing of, tag numbers in the OTISS and AIM-1000 configurations. We used the script language Perl to automate as much as possible of these tasks.

The numbers of connection points and defaults set in the IoDescription file are 1150 and 1400, respectively. Circa one man year was spent on this task. The result was a simulator that ran satisfactorily around different steady state values.

The final simulator verification phase was required to make the training simulator fully operable, i.e. to allow the trainee to start up the platform from a shut down state and execute normal operational procedures. This phase included a detailed walk through of the control structures and logics required to start up and shut down the process.

This phase also served as a useful verification of both the control system and the model. Although the control system configuration proved to be of good quality, the needs for some modifications were reported. Shortcomings in the OTISS model were mainly caused by mismatches in the granularity between the model and the control system. This activity will probably be continuing until the model has been updated with real operational data and the AIM-1000 system on the platform has become stable. The platform will start oil production by the end of 1995.

RESULTS AND EXPERIENCES

The simulator has so far been used successfully for control system verification and initial operator training. It has also been used to fine tune the control loops by a semi automatic method. Inputs and outputs resulting from system perturbation are logged and used off-line to calculate controller parameters. The method is described in Schei (1994).

The most serious problems during the start up period of the HOPE simulator originate from missing details in the OTISS model. These details are not needed for the operational studies the model originally was designed for, but are important to provide the control system with the necessary inputs and to create realistic operator training. However, it has generally not been difficult to extend the model to overcome such problems, but the total time spent on this has caused some delay in the start-up of the initial training.

One lesson learned is therefore to ensure that the specification of the granularity in the model matches both training requirements and the interfaces to the control system. In addition means must be provided for adding simple interface modules to tailoring the model to the requirements of the control system. Such a flexibility is needed because the configuration of the simulator will start before the control system configuration is thoroughly tested. The the simulator development project must thus be prepared to deal with changes in the control system configuration. To ensure consistency between the control system on the plant and that in the simulator all modifications must be done in the model.

The HOPE simulator concept has been taken over by the AIM-1000 system vendor, Simrad Norge, and is likely be re-used in one of Statoil's next major field developments.

The HOPE simulator's accurate process model together with it's copy of the process control system SW and HCI open interesting possibilities for using the system as a test bed for operator support systems and new control strategies. New control strategies can be implemented directly in the control system and tested in a realistic environment. The MiTS protocol gives any other MiTS process access to the measurements from the OTISS model. New operator support tools might therefore be fully integrated and receive on line data.

REFERENCES

Fray, R. (1995). "Compact simulators for fossil-fueled power plants". IEEE Spectrum, Vol. 32 No 2.

Rødseth, Ø.J., Haaland E. (1993). ``MITS: An Open Standard for Integrated Ship Control". Proceedings of ICMES 93, Hamburg September 1993.

Schei, T.S. (1994). "Automatic Tuning of PID Controllers Based on Transfer Function Estimation". Automatica, Vol. 30, No 12, pp 1983-1989. Elsevier Science Ltd.

Woods, E.A., H. Backer, P.E. Wahl and K. Telnes, (1994). "The HOPE Procedure Handling System, "Pre prints of AIRTC'94, 605 - 609, València, Spain.

Yourdon, E. (1989). "Modern structured analysis". Prentice-Hall, Inc.

ANALYSIS OF MULTILAYER MEDIUM ACOUSTIC BEHAVIOUR WITH THE SIMAMCO SOFTWARE

PEYRAUT FRANÇOIS

SCIENCES INDUSTRIES CONSEILS
14, Avenue de Sceaux
78 000 VERSAILLES
FRANCE
Tel. : (1) 39 49 05 27
Fax : (1) 39 02 77 23

1 - INTRODUCTION

This paper deals with the SIMAMCO software which allows the engineers to predict acoustic behaviour of multilayer media. Because of their high heterogeneity, source of many impedance's breaking, these media constitute efficient sound insulation. They are used in various fields like automotive industry, aeronautics, rail transport, shipbuilding, the building trade ...

To compute the reflexion and transmission acoustic response of multilayers with an harmonic incident plane wave, we have used a plane wave approach ([1]). The amplitudes of local waves in each layer are computed by a transfer matrix method. This method is originally due to W.T.Thomson ([6]).

By using a semi-analytical method which combines a plane wave approach with a transfer matrix technique, SIMAMCO gives results very quickly. Consequently, this software is really suitable for design stage.

A possible anisotropic behaviour of layers and an eventual damped behaviour of viscoelastic type are taken into account.

We have written SIMAMCO in fortran 77.

2 - NOTATION

Bold quantities represents vectors. The other quantities are real or complex numbers following the context.

3 - THEORY

We analyse the problem within the frame of small perturbation's theory around a configuration at rest. The fluids upstream and downstream of the multilayer are assumed perfect and we neglect the voluminal forces of gravity.

We also assume that the incident wave is harmonic plane and propagates in free field. In other words, the only multilayer attack wave comes from the source and the only reverse wave is radiated by the multilayer.

We assume in the same way that the transmitted wave downstream of the multilayer is plane and propagates in free field.

Finally, to permit a quasi-analytic solving of the problem, two layer's hypotheses are made :

1) the layers are homogeneous
2) they are infinite in the perpendicular directions to depth axis

With infinite layer's hypotheses, it can be seen that the problem is independant of x variable. This spacial variable is perpendicular to the plane defined by the propagation direction y and the layers depth axis z. Taking into account this geometrical property, the Snell's law ([5]) leads to write the displacements **u** with the form :

$$\mathbf{u}(\mathbf{X},t) = \mathrm{Re}[\mathbf{U}(z)\exp(-i\Omega\sin(\Theta)y/c_0)\exp(i\Omega t)] \qquad (1)$$

t is the time, Ω the circular frequency, Θ the incident angle, c_0 the upstream sound velocity and i the complex number whose square is equal to -1. The Re notation means real part and dash above quantities the conjugate of this quantities.

By combining equation (1) with the laws of momentum's conservation and material's behaviour, we can demonstrate that the vector U is solution of ([4]) :

$$\Omega^2(m_v\mathbf{Id}-\sin^2(\Theta)/c_0{}^2\mathbf{A}^{22})\,\mathbf{U}(z) + i\Omega\sin(\Theta)/c_0(\mathbf{A}^{23}+\mathbf{A}^{32})\,d\mathbf{U}(z)/dz + \mathbf{A}^{33}\,d^2\mathbf{U}(z)/dz^2 = 0 \qquad (2)$$

which is an homogeneous second order differential system with no-constant coefficients.

m_v is the density, **Id** represents the identity matrix and $\mathbf{A}^{ik}$ $(2\leq i,k\leq 3)$ are 3x3 matrix dependant of circular frequency and layers mechanical properties. For each couple of (i,k), the coefficients of these matrix are defined by :

$$(\mathbf{A}^{ik}(\Omega))_{jh} = a_{ijkh}(\Omega)\ (1\leq j,h\leq 3) \qquad (3)$$

where $a_{ijkh}(\Omega)$ are the coefficients of the complex stiffness matrix.

With a first derivative unknown well ajusted to the problem, the second order differential system (2) is turned into a first order system. This system is solved in each layer. The solution is then known analytically except for the integration constants. These constants are determined by displacements and stresse's continuity at layers interfaces in conjunction with a transfer matrix method ([3]).

4 - VALIDATION

SIMAMCO has been valided by numerical and experimental correlations in frequency field, for normal incidence, and in oblique incidence field. Reference curves are issued from ([2]).

Figures 1 and 2 represent the transmission coefficient expressed with decibel as a function of frequency. These two figures concern a microvon panel in water for normal incidence. The microvon's mechanical and geometrical properties are :

- density = 700 kg/m^3
- longitudinal wave velocity = 200 m/s
- longitudinal loss angle tangent = 0.2
- transversal wave velocity = 50 m/s
- transversal loss angle tangent = 0.2

Reference results are numerical (curve named Plan Infini) and experimental (curves referenced by L=0.1, L=0.2 and L=0.3). In each case, we note a perfect correlation with SIMAMCO computation.

On figures 3 and 4, we have plotted the transmission coefficient expressed with decibel against incident angle for a 3 cm steel panel in water. The reference result is issued from a numerical calculation. We note again a perfect concordance with SIMAMCO computation.

5 - CONCLUSIONS AND FUTURE PROSPECTS

The acoustic software SIMAMCO is helpful for sound insulation design, especially for acoustic comfort problems (transport, building trade ...) and military ship's discretion (submarines).

We have successfully valided it in all tested configurations. Its reliability is consequently one of its greater advantage.

With its analytical approach of the propagation problem, SIMAMCO allows users to input quickly data (no tedious mesh generation). For the same reason, results are got very quickly.

SIMAMCO is conceived to take into account various materials (elastic or viscoelastic, isotropic or anisotropic) and stays open for new material's integration (porous or composite materials, elastomere ...).

Finally, SIMAMCO could be easily adjusted to deal with vibratory problems.

Because of its short response delay and its simplicity of using, SIMAMCO is really adjusted to multilayers acoustic design. SIMAMCO is complementary to experimental tests. It allows cost reduction by advising the engineer in material's choice.

REFERENCES

[1] "Wave Propagation in Elastics Solids", J.D. Achenbach, North-Holland, Amsterdam, 1973

[2] "Caractérisation de Matériaux Utilisés en Acoustique Sous-Marine à l'Aide de la Mesure en Cuve des Coefficients de Réflexion et de Transmission de Panneaux de Dimensions Finies", Christian Giangreco, Thèse de l'Université Technologique de Compiègne, décembre 1990

[3] "Calcul et Visualisation de l'Amortissement d'une Onde Harmonique dans un Multicouche Viscoélastique", François Peyraut, Thèse de l'Université Paris 6, décembre 1991

[4] "Amortissement Dans des Composites Viscoélastiques", Nadia Labed, Thèse de l'Université Paris 6, décembre 1992

[5] "Crystals Acoustics", M.J.P. Musgrave, Holden Day, San Francisco, 1970

[6] "Transmission of Elastic Waves Through a Stratified Solid Medium", W.T. Thomson, J. Appl. Phys. 21, 89, 1950

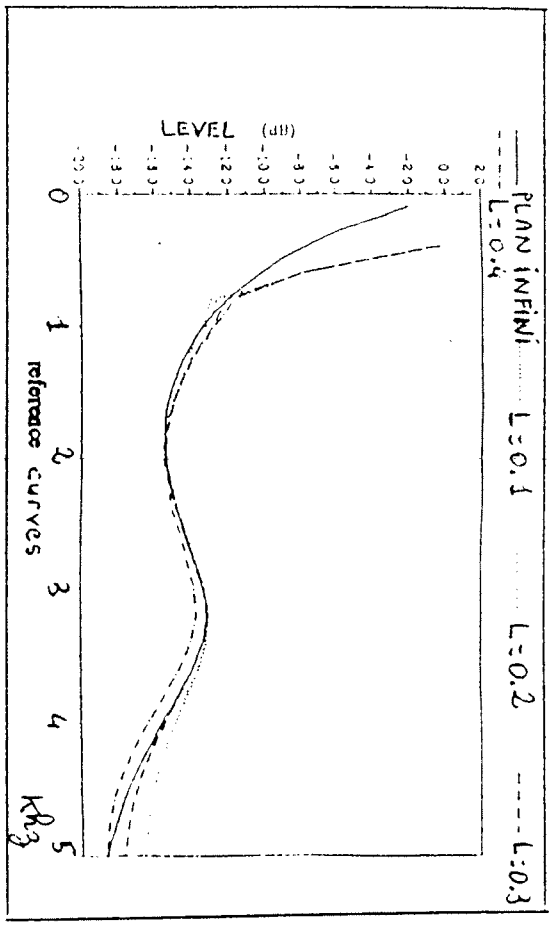


Figure 1

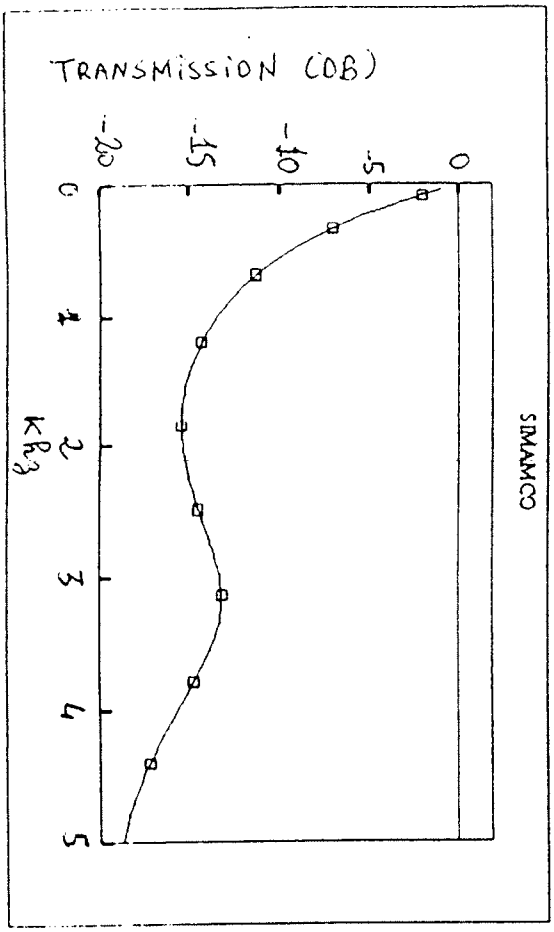


Figure 2

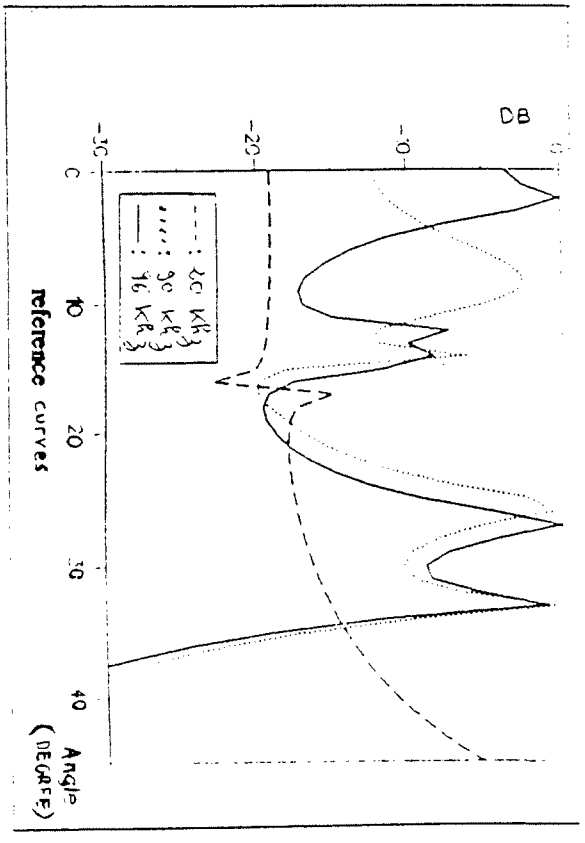


Figure 3

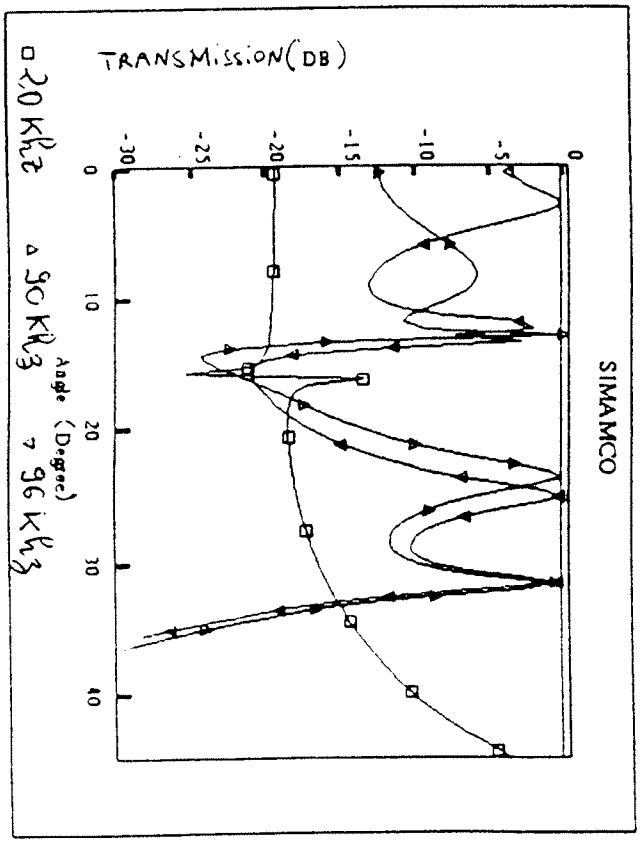


Figure 4

The module partDEQ - solving partial differential equations using SIMUL_R

R. Ruzicka
SIMUTECH
Hadikgasse 150, A-1140 Vienna

Abstract

This contribution will show all features of the simulation language SIMUL_R, which make it possible to solve systems of partial differential equations (PDEs) and display their solutions. The modelling features are contained in a module called *partDEQ*. Some examples will show how to use partDEQ to solve PDE problems.

Introduction

Most modern simulation languages for continuous systems' simulation contain features for solving ordinary differential equations (ODEs). But in many cases ODEs are not as accurate as is necessary to describe all dynamic features of a real system:

think of electronical circuits with long wires or e.g. the bus of a computer; these cannot be simply modelled by using for example a capacitor and a resistor to show the effects of voltage reflections;

or the absorbtion of pollutants by a filter: you need to model the process of absorbtion to be able to predict, when the filter will be filled;

or the closing of valves in waterworks: the waves of water pressure and velocity have to be modelled using PDEs to simulate extreme and dangerous states.

There are a lot of tools, each spezialized in one specific field of PDE simulation: finite element methods for mechanical purposes or special numerical libraries for special types of PDEs. Nevertheless those are difficult to use for general purposes and mostly have no user friendly interface for modelling and presentation of results.

How to add PDE features to a simulation system for ODEs?

A modelling utility for PDEs in an environment for ODEs, like SIMUL_R, should be

- fully integrated into the ODE system (PDEs and ODEs combinable)
- easy to use
- the result functions must be easy accessible
- the results should be drawn over the locality dimensions without user "chin ups"

SIMUL_R features for PDEs

SIMUL_R a priori offers some features, which help modelling PDEs:

- a very powerful macro and meta language
- sorting of equations
- easy modelling of implicite problems
- so-called *cross plots*

Meta and macro commands

SIMUL_R contains all macro features known from the programming langugage C, but extended by very important points:

+ recursive macros

+ varying number of parameters for the same macro
+ meta loops (for, while)

Equation Sorting

The equation sorting algorithm of SIMUL_R - sorting occurs within the DERIVATIVE section of a model for finding dependencies of variables -

+ automatically detects algebraic loops,
+ searchs for a minimal set of variables, necessary to be iterated for solving an implicite problem,
+ transforms the equations into a zero search problem (yet available algorithms, choosable by menu: damped Newton, DASSL),
+ delivers information messages, e.g. about the structure of the equation matrix

  banded matrices can be easily detected, necessary variable resorting can be performed.

+ SIMUL_R offers special banded matrices methods for the Newton and DASSL algorithms.

Implicite problems

can be modelled in SIMUL_R as

$0=g(x)$ zero point notation

or

$x=f(x)$ fix point notation

(as well for vectors and matrices). Therefore it is much easier to use special algorithms for PDEs, like Crank-Nicolson's method.

Cross plots

mean, that the time variations of a set of variables (e.g. variables representing the solution of a PDE over the locality dimension) can be drawn in the vertical y direction of a plot (with constant horizontal x position) and are automatically connected.

These features have been used to build up the partDEQ module with a library of macros, which translate a PDE notation into a set of equations and algorithms, which can be treated by usual SIMUL_R.

Modelling PDEs

PDEs of the form

$$\frac{d^k y}{dt^k} = f\left(t, y, \frac{dy}{dt}, \frac{d^2y}{dt^2}, .., \frac{d^{k-1}y}{dt^{k-1}}, x, \frac{dy}{dx}, \frac{d^2y}{dx^2}, .., \frac{d^m y}{dx^m}\right)$$

where

t	time (first independend variable)	*f*	an arbitrary expression
x	locality (second independend variable)	*k*	maximum derivative order over t
y	state variable	*m*	maximum derivative order over x

are solved and can be inserted as parameters of a partDEQ macro; additionally the initial conditions and the length of locality is specified.

The macros are also available for up to 3 locality variables (3 dimensions) with mixed local derivatives.

Special macros are used to specify boundary conditions (fixed conditions, or switchable conditions).

The results are sampled in user specified array variables, which can be simply plotted using cross plots.

Currently the two methods

- method of lines
- Crank Nicolson (implicite method)

are available, which only differ - at the user interface - in the name prefix of the macro.

Examples

Introductional example

The first example shows a simple model of sucking water into wood:

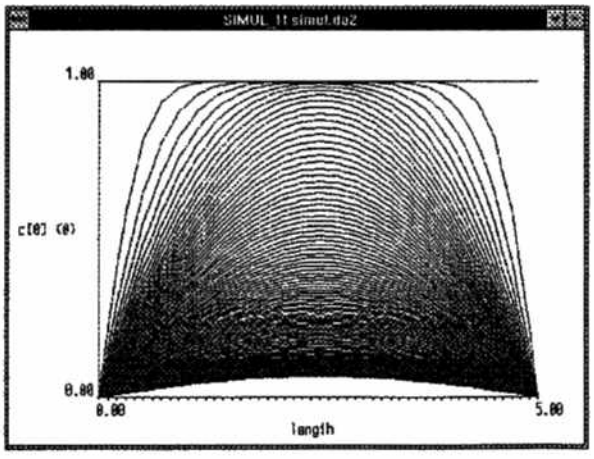
$$\frac{dy}{dt} = D * \frac{d^2y}{dx^2} \qquad \text{with} \qquad y(t0,x) =$$

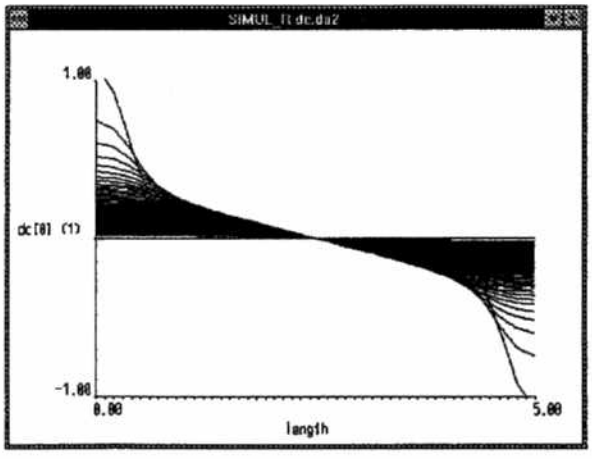
At time *T* a step from 1 to 0 is performed on the left and right boundaries. Here the first and second order locality derivatives are named *dyx* and *dyxx*, respectively.

The model:

```
#set n=50#                          " number of discretization points "
#include 'simcomac.def'
#include 'partdeq.def'              " include module partDEQ "

wood {

        CONSTANT tend=1000, len=1, D = 1e-5, T=0.2;
        float yl, yr, y[#n+1#], dyx[#n+1#], dyxx[#n+1#];
        int i;

        DYNAMIC {
                DERIVATIVE {
                        yl=1-#step(T);                      " left bound "
                        yr=1-#step(T);                      " right bound "
                        #partDEQlrc(1,D*dyxx[i],i,len,yl,yr,y,1,dyx,dyxx)
                }
                TERMINATE t>=tend;                          " termination condition "
        }
}
```

It is easy to draw the state variable results and the locality derivatives over *x*:





Net of water pipes

The second example shows pressure waves in a pipe net of waterworks: each pipe is described by the two coupled PDEs (for pressure *p* and velocity *v*)

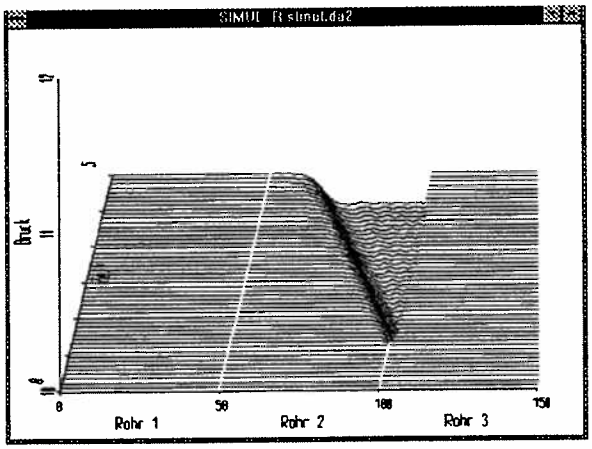
$$\frac{\partial p}{\partial t} = - \frac{c*c}{g} * \frac{\partial v}{\partial x} \quad \text{and} \quad \frac{\partial v}{\partial t} = - \frac{\partial p}{\partial x} * g - \frac{\lambda}{2*D} * v * |v| \quad ,$$

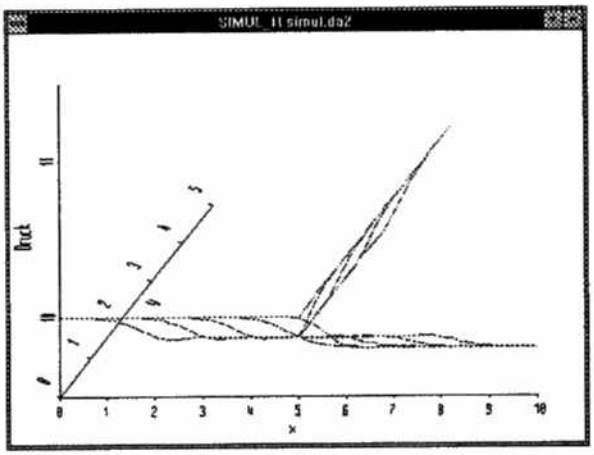
the net is described using bond graphs.

On the left hand side, there is a water container with constant pressure, on the right hand side and at the top there are two valves. A water consumer opens a valve: this leads to water reflections through the entire net.

A pipe is modelled using partDEQ macros for the PDEs and is formulated as a macro. The net is modelled by the bond graph tool BAPS using this macro. This allows for an easy changing of the net's topology.

The pictures show the pressure in the three pipes: the first wave after closing one valve and five states (at five different points of time over the real topology).





Future developments

In the near future there will be available methods for

- automatic adaption of the discretization points and
- new methods for computing the differential quotients.

Literature

- R.Ruzicka: *SIMUL_R - eine Simulationssprache mit speziellen Befehlern zur Modelldarstellung und -analyse*, Informatik Fachberichte 179, Proceedings of the 5th Symposium Simulationstechnik, Springer, Aachen, 1988
- R.Ruzicka: *Environments For SIMUL_R*, 3rd European Simulation Congress, Proceedings, Edinburgh, 1989
- J.Niwinski, R.Ruzicka: *Online Simulation With SIMUL_R*, SCSC 91, Baltimore Maryland, 1991
- R.Ruzicka: *SIMUL_R PARALLEL - Hardware-in-the-loop Simulation mit Transputern unter Windows*, Fortschritte in der Simulationstechnik, Band 6, Proceedings of the 8th Symposium Simulationstechnik, Vieweg, Berlin, 1993
- R.Ruzicka: *Optimierung technischer Systeme mittels Evolutionsstrategien - ein Standardverfahren in SIMUL_R*, Fortschritte in der Simulationstechnik, Band 9, Proceedings of the 9th Symposium Simulationstechnik, Vieweg, Stuttgart, 1994

EXTREMELY EXACT and FAST COMPUTATIONS

Jiří Kunovský, Karel Mikulášek

Department of Computer Science and Engineering, Technical University Brno
Božetěchova 2, CZ–612 66 Brno, The Czech Republic
Telephone: +42-5-7275234 (+42-5-41212219), Fax:+42-5-41211141
e-mail: kunovsky@dcse.fee.vutbr.cz

1 Abstract

The paper deals with the numerical solution of differential equations. The numerical integration employs the Modern Taylor Series Method. The useful properties of the Modern Taylor Series Method are demonstrated on two problems: in homogeneous linear differential equations and nonlinear differential equations with time dependent coefficients. An application of multiple word arithmetic to the solution of stiff systems is shown.

2 Introduction

The best-known and most accurate method of calculating a new value of a numerical solution of a differential equation is to construct the Taylor series in the form

$$y_{n+1} = y_n + h * f(t_n, y_n) + \frac{h^2}{2!} * f^{[1]}(t_n, y_n) + \ldots + \frac{h^p}{p!} * f^{[p-1]}(t_n, y_n), \tag{1}$$

where h is the integration step.

The Modern Taylor Series Method is used for the computation. The main idea behind the Modern Taylor Series Method is an automatic integration method order setting, i.e. using as many Taylor series terms for computing as needed to achieve the required accuracy.

Methods of different orders can be used in a computation. The 1st order method (ORD=1) means that when computing the new value y_{n+1} only the first Taylor series term is taken into account

$$y_{n+1} = y_n + h * f(t_n, y_n), \tag{2}$$

the 2nd order method (ORD=2) uses Taylor series terms up to the second power of the step h

$$y_{n+1} = y_n + h * f(t_n, y_n) + \frac{h^2}{2!} * f^{[1]}(t_n, y_n), \tag{3}$$

etc.

3 Homogenous Linear Differential Equations

Let us solve the differential equation

$$y' = y, \quad y(0) = 1. \tag{4}$$

A numerical solution of (4) by the Taylor series method (using (1)) is

$$y_{n+1} = y_n + h * y_n + \frac{h^2}{2!} * y_n + \ldots + \frac{h^p}{p!} * y_n + \ldots \tag{5}$$

$(y = y' = y'' = \ldots\ldots = y^{(p)})$,

or

$$y_{n+1} = y_n * (1 + h + \frac{h^2}{2!} + ... + \frac{h^p}{p!} + ...). \quad (6)$$

The numerical solution of (4) (using (5) or (6)) will depend on the number of Taylor series terms used.

Note : The analytical solution of (4) is

$$y = e^t. \quad (7)$$

Results illustrating the use of the Taylor series for applying a numerical integration method are shown in Tab.1. Tab.1 demonstrates the results of a numerical solution of the differential equation (4) after one computation step (with the integration step h=1s).

Reduced value y(1)	ORD	Time (ms)
2.	1	0.084
2.	2	0.140
2.	3	0.195
2.7	4	0.248
2.71	5	0.307
2.718	6	0.365
2.7182	7	0.422
2.7182	8	0.468
2.718281	9	0.531
2.7182818	10	0.589
2.71828182	11	0.649
2.718281828	12	0.693
2.7182818284	13	0.757
2.71828182845	14	0.828
2.71828182845	15	0.861
2.71828182845904	16	0.911
2.71828182845904	17	0.983
2.71828182845904	18	1.033

Tab.1

In each line of Tab.1 the "Reduced value y(1)" of the numerical solution of the differential equation (4) and the time evaluation of computation - for the method order ORD used are printed. In order to make the results of the numerical solution more clear and illustrative only the digits tallying the digits of the exact solution are shown (for the exact solution of the equation (4) for $t_1 = h = 1s$, in view of the equation (7), we have

$$y(1) = e^1 = 2.718281828459045235.....).$$

It follows from Tab.1 that the requirement of a higher method order is justifiable - with the same integration step h a higher method order (i. e. with more terms of the Taylor series) can yield a higher accuracy (it approximates better the exact solution).

3.1 Experimental Time Evaluations

Tab.1 brings time evaluation of the computation. For instance, using the 17th method order requires 0.983 ms.

If we wanted to reach the same accuracy (at point t=1s) by the 4th order Runge-Kutta method, we would have to use a substantially shorter integration step and the computation time would be 271.229 ms (Tab.2).

h(s)	Reduced value y(1)	Time (ms)
1	2.7	0.299
0.1	2.7182	2.691
0.01	2.718281828	27.500
0.001	2.71828182845904	271.229

Tab.2

Note: All time evaluations were obtained on the ACA 32000 computer (based on National Semiconductor 32000 processor).

3.2 Accuracy and Word Width

The increase in the accuracy of the result is not unlimited. In Tab.1 the accuracy stops increasing when the ORD reaches the value of 18. This is caused by an underflow during the computation of the higher order Taylor series terms. The addition of these terms changes neither the resulting value of y_{n+1} nor the absolute error - the absolute error has reached its saturated value ESAT (the value of ESAT depends on the word width of the arithmetic of the computer used).

The accuracy of the result can be influenced in a considerable way by the word width. For instance when the word width is 32 bits and the integration step h=1s, only the accuracy $1.863 * 10^{-9}$ of computation of the equation (4) can be reached. With a specially constructed 128-bit arithmetic a very high computation accuracy $4.408 * 10^{-39}$ for the equation (4) can be reached very fast even with the integration step h=1s.

4 Differential Equations with Time Dependent Coefficients

The high accuracy and the high speed of the Modern Taylor Series Method is demonstrated on the following system of equations

$$y' = aycost \quad y(0) = 1 \tag{8}$$
$$x' = -axcost \quad x(0) = 1 \tag{9}$$
$$z = xy \tag{10}$$

The system of equations (8), (9), (10) was deliberately designed for the variable z to characterize the accuracy of the computation.

The accuracy of the computation by the Modern Taylor Series Method is preserved even if the variables reach values of 10^{43} and 10^{-44} by order of magnitude. The numerical solution of the system (8),(9) reaches these values for a=100. The achieved high accuracy of the computation of the expression

$$z = x \cdot y = 1$$

is based on the fact that the new value of y_{n+1} is calculated in each step automatically until the adding of the next Taylor series terms has no effect.

Note: A special comparison has been completed. The value of z(100) (for a=100) has been calculated by the Modern Taylor Series Method and by the 4th order Runge-Kutta method. The result z(100)=1.000000000000000 was achieved after 136 seconds of computation using the Modern Taylor Series Method. In the same time the RK-4 method was able to yield only the value z(100)=1.0064.

5 Stiff Systems

Let us focus our attention on particular problems with the integration of stiff problems. It is well known that in stiff systems an extremely small integration step must be used. If, however, we combine Modern Taylor Series Method and multiple word arithmetic, the computation can be done with an extremely large integration step.

Let us consider a system of linear differential equations

```
y1'=y2,                  y1(0)= 1                          (11)

y2'=-a*y1-(a+1)*y2,      y2(0)=-1                          (12)
```

with the exact solution

$y_1 = e^{-t}, \qquad y_2 = -e^{-t}.$

The general solution of the system (11),(12) is in the form

$y_i = A_i e^{-t} + B_i e^{-a*t}$

$i = 1, 2 \qquad A_i, B_i \qquad$ are constants

and the eigenvalues of the matrix of the system are

$\lambda_1 = -1, \qquad \lambda_2 = -a.$

The results of the computation, for the integration step h=1, accuracy $EPS = 10^{-6}$ and for 30 terms of the Taylor series (ORD=30) are shown in Table 3. The table shows the effect of using the multiple word arithmetic. If the computation is done with the word length 8 bytes, the coefficient a can at most have the maximum value $a = 10^2$. When using the word length 512 bytes, the value of a can be $a = 10^{175}$.

The described positive effect of the use of multiple word arithmetic can also be used to solve intricate nonlinear stiff systems.

Word length (Byte)	a	Time of computation (s)
8	10^2	0.0012
16	10^5	0.0090
32	10^{10}	0.0500
64	10^{21}	0.1000
128	10^{43}	0.3800
256	10^{87}	1.2100
512	10^{175}	4.3400

Tab.3

6 Conclusion

We consider the following points to be a contribution to the methodology of the numerical solution of differential equations:

- Direct use of Taylor Series Method
- Use of variable number of terms of the Taylor series
- Use of multiple word arithmetic
- Applicability

INTERACTIVE SIMULATION OF A FUZZY-LOGIC-CONTROLLED NONLINEAR SERVOMECHANISM

Granino A. Korn

G.A. and T.M. Korn Industrial Consultants, RR1, Box 96C, Chelan, WA 98816

ABSTRACT

DESIRE/NEUNET software (DOS or UNIX), originally designed for combined simulation of neural networks and dynamic systems, is equally suitable for interactive simulation of fuzzy-logic control systems. A single page of code, much like ordinary mathematical notation, produces complete sets of nonuniformly spaced fuzzy-set membership functions as vector differences of simple hard-limiter neuron activations, implements rule table and defuzzification, and solves the nonlinear differential equations for a servomechanicsm. An interpreted experiment-protocol program sets and modifies fuzzy-set centers, servo parameters, and rule-table entries. When the interpreter calls for a simulation run, a fast runtime compiler compiles fuzzy-logic operations, rule-table references, and differential equations within 0.05 sec - an unnoticeable delay. The solution then starts immediately and produces color graphics. Interactive rule-table modifications easily permit experiments with different fuzzy-set partitions and nonlinear control schemes.

TRULY INTERACTIVE SIMULATION OF COMPLETE DYNAMIC SYSTEMS

A fuzzy-logic controller is a function generator which produces a desired control function from sensor inputs. Rule-table-based fuzzy-logic controllers let designers use intuition or prior knowledge by working with simple linguistic rules, such as

IF output error is positive AND output rate is positive THEN torque is negative

We will simulate the application of such a controller to an electromechanical servomechanism which is described by nonlinear differential equations.. Its servo motor drives its load so that the servo output angle **x** = *x(t)* follows an input waveform **u** = *u(t)* after an initial transient. **x** is read from an angle pickoff, and its time derivative **xdot** is read from a tachometer on the motor shaft. The controller produces the motor control voltage **V** as a function of the servo error **e = x - u** and the output rate **xdot**. **V** is then amplified and causes a motor torque **torque**, which is limited by motor-field saturation. For convenience, we scaled the effective moment of inertia of motor and gears to equal 1. The dynamics of motor, gears, and load are modelled by nonlinear differential equations.

We employ "direct-executing" simulation software (DESIRE/NEUNET) incorporating a runtime compiler which translates even large simulation programs so quickly that there is no noticeable translation delay. This technique permits truly interactive modelling. An interpreted *experiment protocol program* - much like an advanced basic program - sets up parameters, function tables, and arrays and then calls for a simulation run with a **drun** command. At this point, the simulation-run code (*DYNAMIC program segment*), automatically compiles into fast binary code, which then runs immediately.

DESIRE lets you enter and edit experiment protocol and scalar and/or vector differential equations, plus neural-network and fuzzy-logic models on the CRT screen. Reference 3 lists the complete program. The DYNAMIC program segment segment describing the servo and controller has only 13 program lines. The servo equations of motion are entered in quite ordinary, easily readable mathematical notation:

```
d/dt x = xdot ¦ d/dt xdot = torque - R * xdot
```

where **R** is a motor damping coefficient. The motor torque is modelled with

```
torque = - maxtrq * sat(g * V/maxtrq)
```

Motor field saturation limits the torque between - **maxtrq** and **maxtrq**. We used a hard limiter, but you can replace **sat()** with a soft limiter such as **tanh()**.

FUZZY-LOGIC CONTROLLER

The servo controller is a function generator producing **V** as a function of **e** and **xdot**. A simple linear controller might implement

```
V = - k * e - r * xdot
```

Instead, we experiment with a more powerful nonlinear two-input *fuzzy-logic controller*. The range of each of the controller input variables **e** and **xdot** is partitioned into 5 fuzzy sets, which could be labelled negative-large, negative-small, zero, positive-small, and positive-large. The rule table holds 25 entries corresponding to different combinations of **e** and **xdot** values.

GENERATING NONUNIFORMLY SPACED MEMBERSHIP FUNCTIONS

We need a one-dimensional array (vector) **mbre** of *Ne* triangular membership functions **mbre[j]** of **e**, and a vector **mbrxdot** of *Nxdot* membership functions **mbrxdot[k]** of **xdot.** For extra resolution near zero error and output rate, *we space our fuzzy sets more closely near zero.* The DESIRE experiment-protocol program reads nonuniformly spaced membership-peak coordinates **E[j]** and **Xdot[k]** from short **data** lists,

```
data - 2 * emax, - 0.5 * emax, 0, 0.5 * emax,
  2 * emax  ¦  read E
data - 2 * xdotmax, - 0.5 * xdotmax, 0,
  0.5 * xdotmax, 2 * xdotmax  ¦  read Xdot
```

To produce, say, **mbre**, the DYNAMIC program segment uses DESIRE **VECTOR** assignments [2,3] which first define hard-limiter functions **mbe** starting at **E[j]** and then subtract successive pairs of such limiter functions[4] to generate the nonuniformly spaced triangular functions **mbre[j]** shown in Fig. 1a.

The one-dimensional membership functions for **r** and **xdot** combine into joint membership functions **M12[j,k]**, with

```
MATRIX M12 = mbre * mbrxdot
```
(product/sum logic)

or

```
MATRIX M12 = mbre & mbrxdot
```
(min/max logic)

This produces the two-dimensional membership functions **M12** of **e** and **xdot** as elements **M12[j,k]** of a rectangular matrix **M12**. But the DESIRE declaration

```
ARRAY M12[Ne, Nxdot] = m12
```

also lets us access the joint membership functions **M12[j,k]** as elements **m12[i]** of a one-dimensional array (vector) **m12**, in the sequence

(negative-large **e**, negative-large **xdot**)
(negative-large **e**, negative-small **xdot**) ...
(negative-small **e**, negative-large **xdot**)
(negative-small **e**, negative-small **xdot**) ...
. . . . . .

The controller output **V** can then be obtained neatly as the inner product of the joint-membership-function vector **m12** and the rule-table vector **ruletabl**:

```
DOT V = m12 * ruletabl
```

The rule-table entries **ruletabl[i]** are programmed in another **data** list in the experiment-protocol program.

This convenient method of combining "singleton" rule-table entries and joint membership functions is *not* restricted to two-dimensional joint membership functions, since **m12** could be combined, in turn, with a third set of one-dimensional mebership functions by product or minimum fuzzy logic; the resulting two-dimensinal membership-function array **M123** is again made equivalent to a new one-dimensional array **m123**, and this process can be continued as needed.[3]

INTUITIVE DESIGN WITH LINGUISTIC RULES

The controller rules, presumably, ought to counter observed servo errors with torques of opposing sign. If this is done with a large gain parameter **k**, the output will overshoot and oscillate, so we damp the response with a torque component opposing the output rate **xdot**. It may or may not be useful to decrease these effects for small absolute errors and output rates. As a starting point, one can approximate a linear servo or a bang-bang controller.

Figure 1b shows the servo response with the rule table

```
k = 200  ¦  r = 40
data -8 * k-r, - 8 * k-r, -8 * k, -8 * k+r, -8 * k
data - k - r, - k - r, - k, - k + r, - k + r
data - r, - r, 0, r, r
data k - r, k - r, k, k + r, k + r
data 8 * k-r, 8 * k-r, 8 * k, 8 * k+r, 8 * k+r
```

This approximates the response of a linear servo, [1] but produces less than linear damping for large **¦xdot¦** values.

We can now experiment with new linguistic rules for improving the servo response. To lower the sinusoid-following error without causing too much of an initial transient, we substituted a narrower zero-zone fuzzy set for the servo error **e** via the new **data** list

```
data - 2 * emax, - 0.1 * emax, 0, 0.1 * emax,
     2 * emax  ¦  read E
```

and tried to kill servo damping in this low-error zone with the modified rule table

```
k = 150  ¦  r = 120
data - 8 * k - 2 * r, - 8 * k - r, - 8 * k, - 8 * k + r,
     - 8 * k + 2 * r
data - k - 2 * r, - k - r, - k, -k + r, - k + 2 * r
data - r, 0, 0, 0, r
data k - 2 * r, k - r, k, k + r, k + 2 * r
data 8 * k - 2 * r, 8 * k - r, 8 * k, 8 * k + r,
     8 * k + 2 * r
```

Figure 1c shows the resulting, entirely different response. The servo now follows the sine table with remarkable accuracy (note that we displayed *one thousand times* the servo error **e** !). The example illustrates the remarkable possibilities of experiments with membership functions and rule table.

USING A NOTEBOOK FILE

Computer simulation experiments require you to record and recall many successive parameter lists, rule-table entries, simulation time histories, and modified-program files. To keep track of such material, DESIRE/NEUNET maintains a ***notebook file*** which automatically records file operations and also lets you lets you record selected program or data lines, and your own comments, with command-mode or programmed **note** *line-number list* and **note** *'comment'* statements.

REFERENCES

1. Korn, G.A.: *Interactive Dynanic-system Simulation*, McGraw-Hill, New York, 1989.

2. -- : *Neural-network Experiments on Personal Computers and Workstations*, MIT Press, Cambridge, MA, 1991.

3. -- : *Neural Networks and Fuzzy-logic Control on Personal Computers and Workstations*, MIT Press, Cambridge, MA, 1995.

4. Geva, S., and J. Sitte: A Constructive Method for Multivariate Function Approximation, *IEEE Trans. on Neural Networks*, **3** : 622-624, 1992.

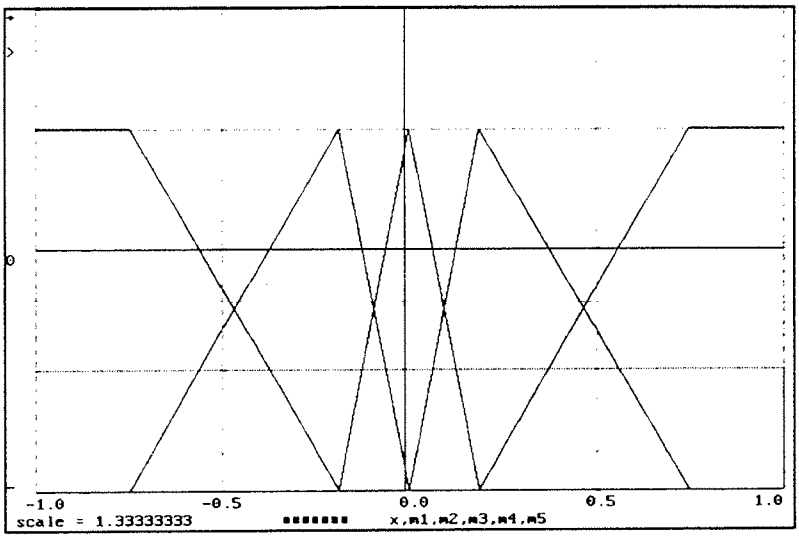


Figure 1a

These servo-error membership functions provide extra resolution near zero error.

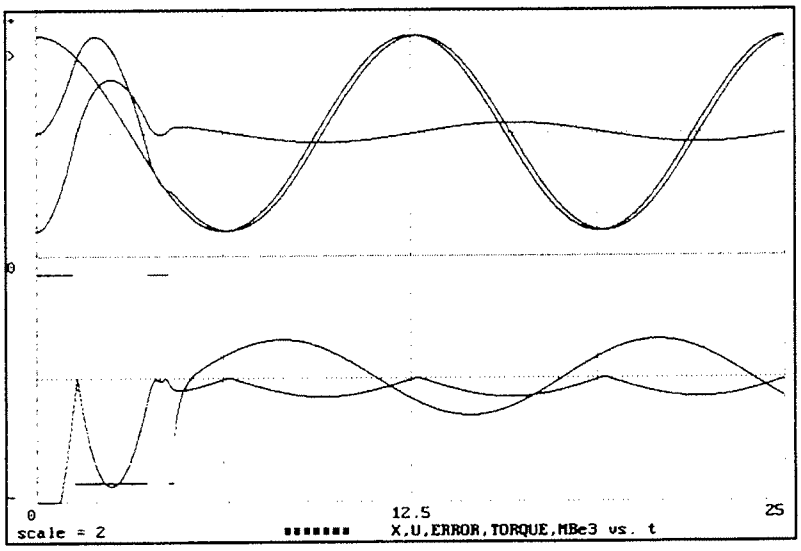


Figure 1b

Servo input, output, error, torque, and zero-error membership function vs. time. The original display was in color.

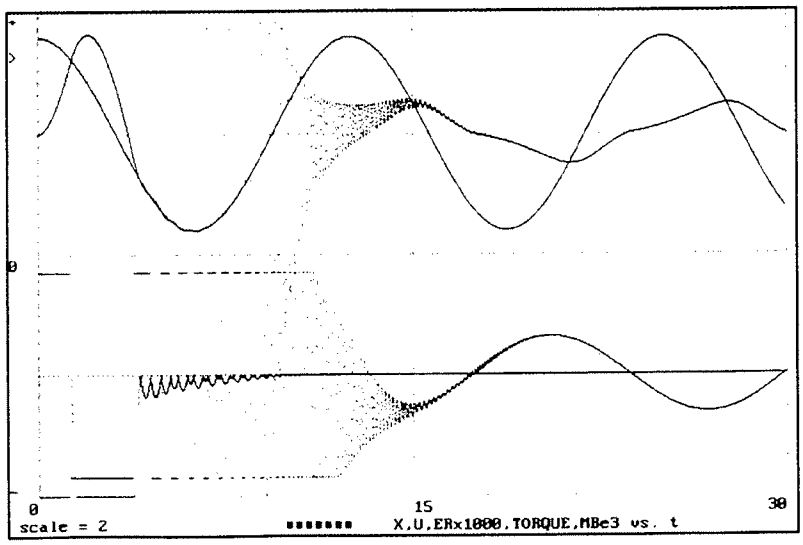


Figure 1c

Servo response with a radically non-linear rule table. The system hunts initially and then follows the sine wave extremely accurately.

Simulation-Based Analysis and Optimization of Complex Systems Using SIMPLORER®

Dipl.-Ing. B. Knorr, cand. ing C. Strunz
Simec GmbH & Co KG, Germany

The growing complexity of system designs as well as decreasing development times require an efficient analysis and optimization of technical systems during computer-aided system engineering. A lot of technical tasks can be reduced to a problem of parameter optimization by selecting relevant system parameters and criteria. The solution of such problems can be realized by various optimization techniques.

Modern mixed-mode/mixed-signal simulation systems, like SIMPLORER®, allow an easy, fast and efficient model creation even for extremely complex systems. The time necessary to examine the desired and the optimal system behavior is considerably high. Based on the automation of system analysis and optimization an optimization module is presented. This module is oriented on approaches which are typical in engineering.

The engineer can use computer-aided analysis and optimization tools in a wide area of technological processes starting from the adjustment of optimal closed-loop controller parameters up to limit analyses. To illustrate the procedure of optimization using SIMPLORER® the dynamic behavior of a battery-powered electrical drive system is examined.

System Development Using SIMPLORER®

The mixed-mode/mixed-signal simulation system SIMPLORER® allows the simulation of mixed technical systems, e.g. of electronic and power-electronic circuits also under consideration of control units. The simulation problem can be modelled in three different description languages. Electric/electronic circuits can be described by lumped components, closed-loop control systems can be represented with block diagrams (signal flow graphs) and control schemes can be modelled using state graphs. Consequently, it is possible to simulate heterogenous systems, e.g., mechanical, pure electronic, and control engineering problems within one system.

SIMPLORER® was recently extended by an optimization module. This module can be used in all cases, where complex interactions and various restrictions make it difficult to find the optimum solution for the development of new products. Great emphasis is put on the analysis and optimization under consideration of the required function on one hand and on the other on the given technological data. So the engineer has the chance to evaluate alternative product designs under technological and economical aspects before the real process will be modified.

Optimization of a Battery-Powered Electrical Drive System

The dynamic behavior of a battery-powered d.c. drive is examined according to the influence of varying system parameters. From the simulation results the optimum solution is extracted.

1. Simulation Model

The system consists of a d.c. motor which is powered by a battery. The electrical circuit of the d.c. motor (armature resistance, armature inductivity, counter e.m.f.) and the transistor pulse chopper are generated by the component library of the SIMPLORER® network modul. The mechanical part of the d.c. motor and the speed regulator are modelled on block diagrams (signal flow graphs) from the SIMPLORER® block diagram modul.

For an illustration of the entire system see Figure 1.

2. Optimization

Various analyses of the battery-powered d.c. drive model are used to optimize the dynamic behavior and to show parameter limits of the system. The aim of system optimization is to ensure the best quality factor by variation of given system parameters. The quality factor of a system stands for the deviation from accepted system behavior.

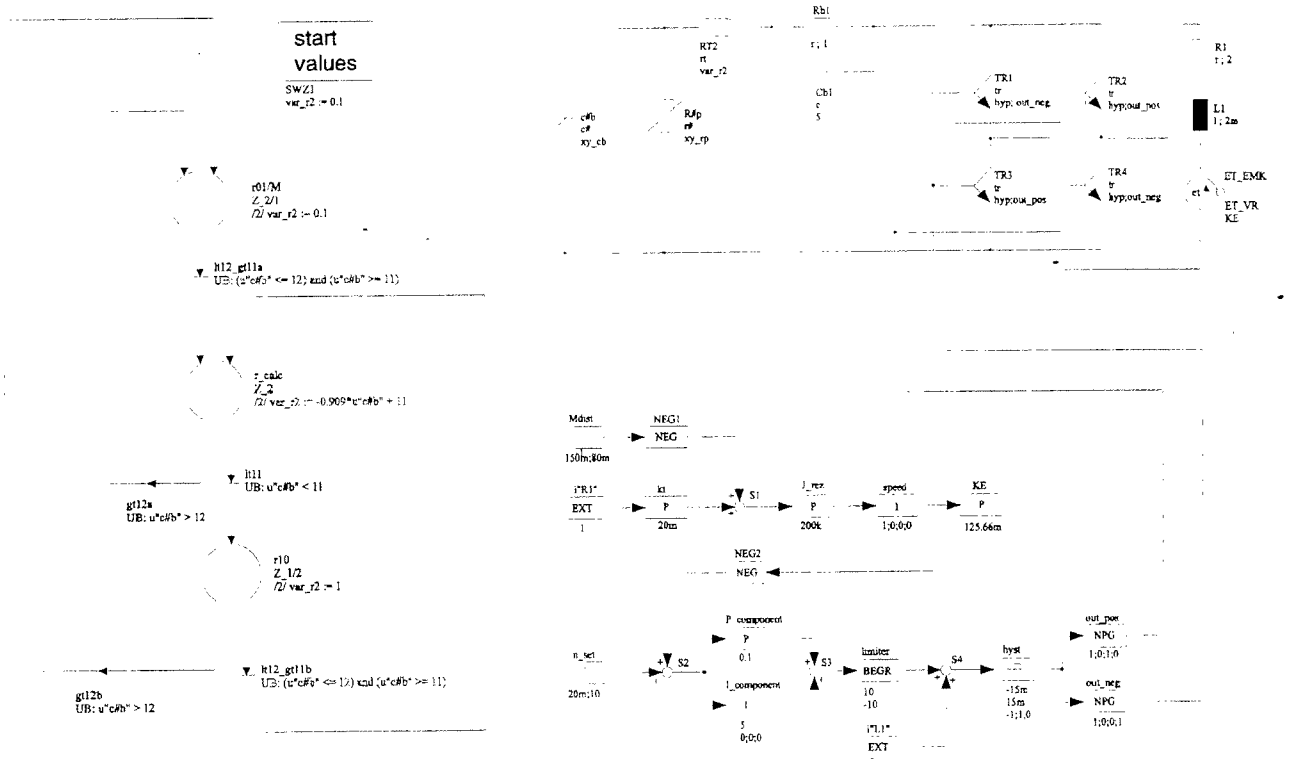


Figure 1: Simulation Model of the Battery-Powered Drive System

2.1. Introduction to the SIMPLORER® Optimizer

To realize an automatic system design the new SIMPLORER® optimization module is exploited. It supports the following kinds of analyses:

- Trend Analysis
- Monte-Carlo-Analyis
- Worst-Case-Analysis

Furthermore, the optimization module provides techniques of intelligent algorithms. A quality criterion (the target function) has to be maximized or minimized as a result of optimization.

The character of an optimization task is represented by its analysis. The analyses differ in the kind of parameter variation. The Trend Analysis is a more straight approach in opposition to the Monte-Carlo-Analysis which assumes a random parameter variation. In a Worst-Case-Analysis parameter limits of a system are interesting. Intelligent algortihms generate the new parameter set for the optimizer based on the results of the previous simulation run.

The use of an analysis combination as well as the use of a single analysis makes it possible to solve an optimization task. Every analysis has its own characteristics and can so be a helpful tool for computer-aided system engineering.

2.2. Usage of the Characteristic Value Library for the Specification of Target Functions

The quality of a system can be described by a number of various characteristic values. Even during an optimization these values have a great importance.

To automize the system design it is necessary to calculate the selected characteristic values during the simulation and not after the simulation. Petri Nets are the basis therefore, which can efficiently be described using the SIMPLORER® state graph module.

Supported by a library with prepared calculation algorithms of characteristic values it is easy to include optimization criteria in the actual simulation task. The particular macro which calculates the target function has to be specified by its parameters and can be evaluated in the same way.

2.3. Optimization of the Dynamic Behavior by Adjustment of Optimum Controller Parameters

2.3.1. Specification of the Optimization Task

The transfer function of the control process is fixed. A PI-Controller is part of the control process. The parameters of its transfer function can freely be selected. The controller realizes a correction of the stationary and dynamic behavior which results in a change of the system quality factor. The adjustment of the PI-Controller's parameters leads to an optimum dynamic behavior of the drive system especially if the reference value and the disturbance value will be modificated.

The d.c. drive has to be examined in no-load operation, in motor operation and in braking operation. The initial parameter set of the considered PI-Controller is P-Part=0.1 and I-Part=5. The simulation model has the system behavior shown in Figure 2 below, where the reference and actual value of the rotation is visible. In modifying the counter torque deviations between these variables are conspicuous. The controller's parameter set has to be determined, so that deviations from the rotation reference will be corrected fast and free of overshoot.

The calculation algorithms of this target functions have to be included via the Characteristic Value Library. The target functions of interest are listed in chart 1:

Transient Part	Characteristic Values
no-load operation t=20ms to t=80ms	maximum overshoot time to maximum
motor operation t=80ms to t=180ms	minimimum rotation control times
braking operation t=180ms to t=400ms	maximum rotation control times
overall transient	integral squared error

Chart 1: Target functions

2.3.2. Realization of the Optimization Task

An efficient way to realize an optimization of multiple target functions (so-called polyoptimization) is the use of the Monte-Carlo-Analysis.

By specification of desired limits or ranges for the target functions it is possible to select the simulated parameter sets during optimization. Thus, the optimizer generates a data set that meets all given specifications. The available storage methods are either all solutions, all permissible solutions or the Pareto Set.

In the case of the battery-powered d.c. drive a Monte-Carlo-Analysis with 100 simulation runs has been realized. The P-Part was evenly distributed between 0.1 and 7 and the I-Part was evenly distributed between 5 and 15. All solutions has been stored in a special file to keep the complex interdependencies also after the optimization.

2.3.3. Evaluation of Optimization Results Using Day-Optim

With the help of the postprocessor DAY, embedded in the simulation system SIMPLORER®, the optimization results can be filtered and sorted. Combined with the optional statistic evaluation part of DAY the simulated data sets can be pureposefully post-analysed. During the decision procedure the user has almost no restrictions. So he can use his own experience and special knowledge of the system.

Optimization can be applied in a number of ways. A variety of features that insures flexibility gives the user a complete control over the resolution of the results. Possible methods to find the optimum parameter set are:

- optimization of a single target function by sorting (rise or fall)
- optimization of multiple target functions by:
  - ▸ setting bounderies (filter)
  - ▸ generating the Pareto Set

and sensible combinations of these methods can rapidly increase the efficiency of the decision procedure.

Optimizaion by generation of the Pareto Set:
The comprehensive estimation of multiple criteria often causes the improvement of one criterion by simultaneous deterioration of another criterion. In this case the result of optimization can be a compromise, the so-called Pareto Set.

For the example of the Monte-Carlo-Analysis of the electrical d.c. drive the Pareto Set was found with the following adjustments:
(1) min. rotation in motor operation ➡ Maximum
(2) rise time in motor operation ➡ Minimum
(3) settling time in motor operation ➡ Minimum
(4) integral squared error ➡ Minimum
and Day-Optim has generated these solutions:

Data Set	P-Part	I-Part	Integral Squ. Error
23 -	6.8206	13.29	1.8356
48	6.2893	13.2	1.858
60	6.7171	11.13	1.8585
73	1.0867	14.43	4.2728
87	0.9418	13.66	4.7895
88	0.4036	11.74	9.4902
99	0.3484	11.14	11.024

Chart 2: Pareto Set

The decision on an optimum parameter set was caused by a minimum integral squared error (set 23). The transient behavior of the system before and after the optimization is shown in the Figures 2 and 3 below.

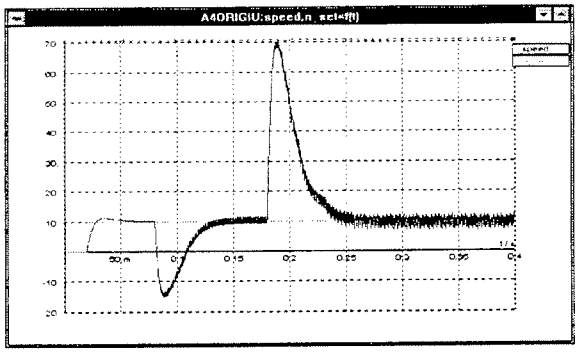


Figure 2: Transient Behavior of Rotation before Optimization

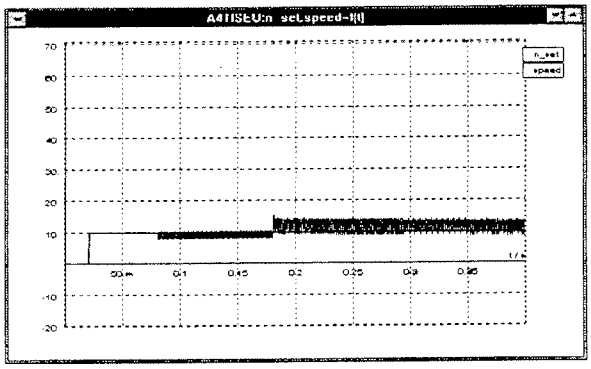


Figure 3: Transient Behavior of Rotation after Optimization

2.4. Analysis of Trends by Varying Counter Torque

A Trend Analysis was made to show limits of system loadability. The optimum controller parameters (P-Part=6.8206 and I-Part=13.29) has been inserted in the simulation model. The counter torque was evenly increased in the range of 5mNm to 80mNm.

Target functions to be obtained were (motor operation):

- maximum and minimum rotation
- time peroid of acceptable battery power and
- the integral squared error .

Results of the Trend Analysis with increasing counter torque are presented in Figure 4, 5 and 6.

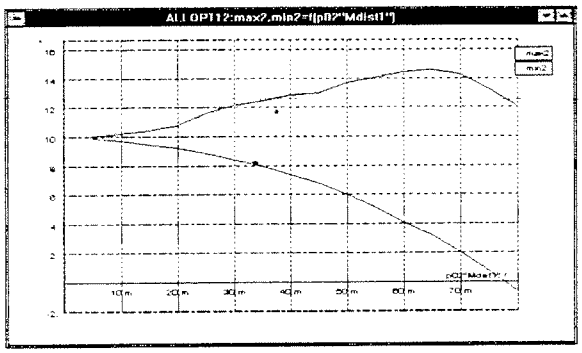


Figure 4: Maximum and Minimum of Rotation

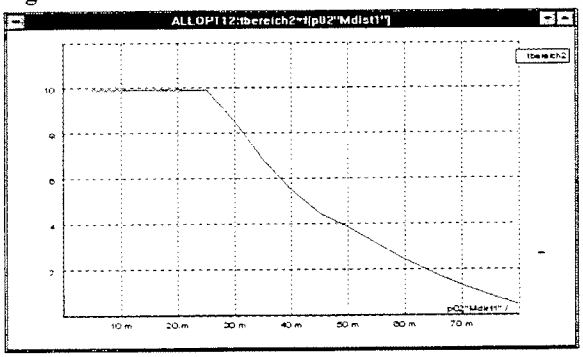


Figure 5: Time Period of Acceptable Battery Power

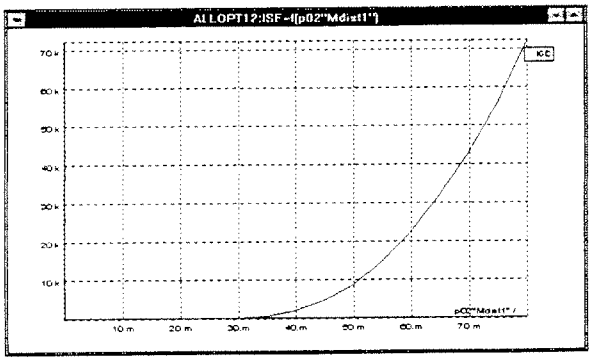


Figure 6: Integral Squared Error

3. Conclusion

By means of optimization of a d.c. drive various optimization techniques has been introducted. The peculiarity of this analysis is the intelligence of evaluation. The engineer is now able interactively to make a decisive, process-oriented classification based on simulated data sets.

New Developments in the Parallel Simulation System mosis

G.Schuster, F.Breitenecker
ARGE Simulation News, Technical University of Vienna, Wiedner Hauptstr. 8-10, A-1040 Wien, Austria
E-Mail: guenter@osiris.tuwien.ac.at

Abstract

This contribution describes the newest developments in the modular CSSL simulation system mosis that were implemented within the last year. After a short description of this simulation language the new features will be described in detail. Those developments include:

- graphical user interfaces under Microsoft Windows and the XWindow system
- handling of implicit models
- object-oriented definition of simulation models
- other new developments under development including graphical modeller

1. Description of mosis

mosis (modular simulation system) is a CSSL-type compiling simulation language on a "C"-basis with special features for modular development and parallelization of simulation models. The first version of mosis was implemented as a part of the dissertation of G.Schuster which was meant as a practical implementation of the *Model Interconnection Concept*. This is a concept for describing bigger simulation models from several independent models that are connected via special data links; those models can even be simulated on different processors in a multiprocessor network (*parallelization*). During this work, many features of other simulation languages were included in this system, so with the first release in 1994 it had most features of commercial systems and by now it is a full all-purpose high-performance simulation language which can be used on parallel systems (including workstation clusters) and singleprocessor systems as well.
mosis is distributed as *freeware*, i.e. it may be used and copied by anyone with no restrictions (although copyrighted software), provided the package or parts of it (or executable programs created from it) are not sold for commercial profit (except a copying fee). The system (complete packages, source code and documentation) can be downloaded from the TU Vienna simulation server <ftp://simserv.tuwien.ac.at>. Special services for commercial customers are offered by the software company "Advanced Technical Software GmbH" which is located at Vienna, Austria.

1.1 The Model Interconnection Concept

The *Model Interconnection Concept* (MIC) is the theoretical background of the parallelization strategy, modular development and external system connection in mosis. According to the definition in this concept, a model can be one of the following:

- an algorithmic description of a dynamic system (also with discrete model parts)
- a test model (a model which is only used for validation of another model)
- a predefined function or a constant or
- even an interface to an external - "real" - system or another simulation package.

A complete simulation model can be defined by connecting several independent models. For this reason, the parts of the whole model can have several input and output signals for data exchange with other simulation models. Parts of the complete simulation can be devices existing in real ("Hardware in the Loop"), interaction with humans ("Man in the Loop") or can be performed by other specialized simulation systems (Simulator Coupling). The parts of a connected model can be simulated on a distributed processor network which gives a simple and efficient strategy for parallelization.

1.2 Existing Features

With the first version of mosis (which was released shortly after the completion of the dissertation of G.Schuster at the end of 1994), although an experimental simulation system, mosis already had most features of modern commercial CSSL languages with a remarkably higher simulation speed (even much faster on one single processor than other compiling simulation languages). The main features of mosis 1.0 are:

- High level model description language on a "C"-basis, including a preprocessor (superset of "C"-preprocessor), very fast compiler. The language follows the CSSL standard (sections) with the

possibility to insert any "C"-statements within the code; user-functions can be called from within the model. Names and functions follow the standard ANSI-C name conventions.
- It includes a user-friendly make utility that performs all steps (translation, compilation and linkage) to translate the models and user-functions to the run time system.
- Several models can be defined and linked simultaneously to the run time system.
- Dynamic model instancing at run time level; several instances of one model are possible (e.g. on different processors in a network).
- High-level interpreter language (similar to "C") for simple functions, loops, graphics at run time level; hardware independent: all commands are exactly the same either used on a multiprocessor UNIX-network or on a simple PC.
- All simulation are done in the background; during a run it is always possible to enter user commands (start, stop, display, ...)
- All simulations are done in double precision (IEEE 64 Bit)
- Easy interfacing with other systems: Connection to other simulators or to external systems (hardware in the loop, man in the loop) by accessing interface routines from the simulation program.
- Many experimentation commands for optimization, graphical output etc.; easy way to implement new commands in "C".

The following hardware platforms have already been supported in version 1.0:

- MS-DOS under Borland C/C++ 3.1
- MS-DOS 32Bit (386+) under Watcom C/C++ 10.0
- UNIX systems under PVM (parallel virtual machine) as a communication system.

In the following sections the new features of the recent release will be described:

2. Graphical User Interfaces

In the previous implementations mosis had quite a simple user interfaces: The DOS versions (Borland, Watcom/32 Bit) had a command line input (with history lists etc.) and several graph screens that could be viewed by pressing a special function key (or by an interpreter command), while the UNIX version only had a very basic possibility to edit the interpreter commands and an interface to the "gnuplot" program to draw curves on a graphical interface (e.g X Window). By the new version of mosis, versions for two of the most commonly used graphical user interfaces (GUIs) have been implemented: For the PCs a version for Microsoft Windows (32 bit) is now supplied; for the UNIX world the XWindow implementation of the mosis simulation system has been completed. Both versions are quite similar to use (except the visual appearance dependent from the windowing system) and offer the following features:

- A command window consisting of a menu bar, a output area (where all output from the simulation system is written to) and an input line. The text in the input line may be edited (plus cut/copy/paste); a history list is also supplied. The text in the output area can also be transferred to other application by "Copy"; a certain number of lines off top may be recorded (user-selectable).
- Many commonly used commands can be entered by menu commands with the mouse.
- up to 32 simultaneous graph windows; they can be resized, moved and iconized.

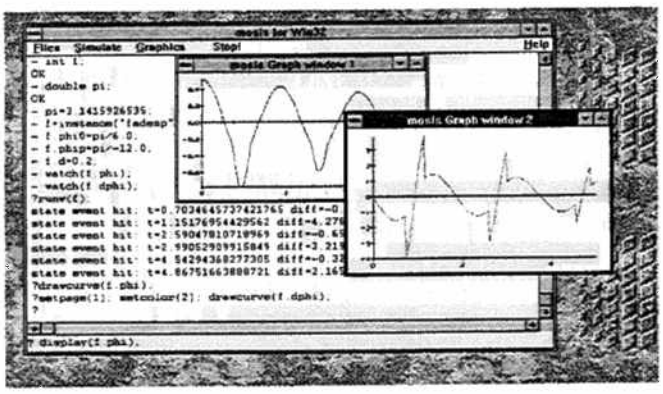


Fig. 1 shows a simple screenshot of the MS-Windows version with two open graph windows. For future compatibility and higher efficiency, the *Win32s* system has been used. With this and other special techniques, (in contrary to other simulation systems) the Windows version is not at all slower than the DOS-version (which itself runs extremely fast).

In the MS-Windows version also a graphical make utility is provided which corresponds to an IDE (integrated development environment) in other programming languages. By using this, the files that should be translated to the final simulation program can be entered in special fields, they can be edited by pressing a single button or removed from the make list. Moreover, the make process can be performed automatically and the resulting file can be executed by a single button press.

3. Implicit Model Definition

This new feature of mosis refers to the fact that not all problems in simulation technique can be easily described from ordinary differential equations (ODEs). In the recent years, much emphasis has been laid on the research of differential algebraic equations (DAEs). In the ODE approach, a dynamical change in a state is described as

$$\dot{x} = f(t,x)$$

while in the DAE approach the dynamics are described as

$$G(t,x,\dot{x}) = 0$$

(f, G are functions). Each ODE can also described as a DAE ($f(t,x) - \dot{x} = 0$), but not vice versa. There are several algorithms that can handle this type of problem (by a solution finder and iterations), usually in connection with other features like stepsize control, automatic handling of stiff systems etc. The probably most famous algorithm of this type is DASSL by L.Petzold which has also been integrated in the new mosis simulation system. In order to be able to use this algorithm, new syntactical structures had to be implemented in the mosis model description language:
A new type of variables - the algebraic variables - have been added to the existing variable types; they can be distinguished from the "normal" state variables. Those variables are declared with the algebraic keyword, e.g.:

```
algebraic x,y;
state z;
```

which determines that the variables x and y are part of the vector of algebraic variables, while z is inserted in the state vector. The DAEs are defined in a new special block - "`implicit`" - which exists beside the ODE block "`derivative`". Within the implicit block, the DAEs are defined in the form of

```
0 = G(t,x,x`);
```

(where G is replaced by the appropriate function of t, x - or another algebraic variable - and its first derivative x` (mosis-syntax for $\dot{x}$)). If an equation can be written as an ODE but it should be nevertheless calculated toghether with the other DAEs, it can be written in the known mosis syntax for ordinary differential equations (of first or higher order):

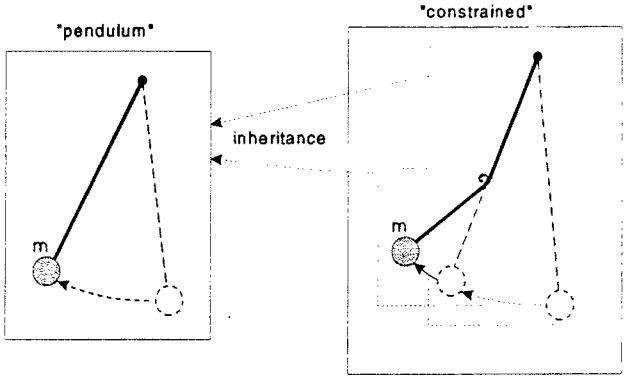
```
x` = f(t,x);
```

where f has to be replaced by the actual function describing the dynamic behaviour.

With this construction, it is possible to have two sections that describe the dynamics of a system that are calculated by two different integrators - one with DASSL and the other one with another integration algorithm, e.g. a Runge-Kutta one - which could increase the simulation speed by only having to calculate the implicit parts by the implicit algorithm and the other parts by a much faster one.

4. Object-Oriented Modelling

The new version of mosis incorporates an object oriented modelling approach which gives the user much higher flexibility and maintainability in writing simulation models and a clear model structure. The main key of this feature is the possibility to inherit methods of other simulation models; even of several different models if e.g. the described system consists of several parts that have been already described as mosis-models. By use of the `inherit` statement it is possible to use all methods of the given model name. A simple example for this would be a constrained pendulum that may hit a pin positioned at a certain angle. For this model the description of a general pendulum (without the constraints) can be used by inheriting the previously defined methods: The constrained pendulum can be described as a general pendulum that switches to another pendulum with a smaller length when the pin is hit. When the smaller pendulum swings back and reaches the switching angle, it turns again back to the first one. For higher performance, those two general models (with different lengths) can occupy the same address space (as only one model is active at a time) by use of the



shared keyword. Switching between those models (in an appropriate discrete section) can be done by issuing the activate command. The main structure of a mosis/2 model would be:

```
model constrained()
{
  inherit shared pendel LongPend, ShortPend;
  // use methods of "model pendel"

  LongPend:dynamic     // append to dynamic section in "LongPend"
  { sevent(LongPend.phi-Dphi,_NEG_,SwitchShort); }
  ShortPend:dynamic
  { sevent(ShortPend.phi-Dphi,_POS_,SwitchLong); }
  discrete SwitchLong  { activate LongPend; }
  discrete SwitchShort { activate ShortPend; }
}
```

With the "LongPend:dynamic" statement it is possible to append certain statements to the dynamic section of the pendulum model called *LongPend*. In the corresponding *discrete*-blocks it can be switched between the short and the long pendulum model (as the two models are used alternatively). It would be also possible to prevent the system to execute a section of the inherited model, to replace it or to add special features (shown above).

5. Other Implementations

Besides the described extensions and new implementations, other programs and modules have either been written or are under development. Those are:

- a graphical modeller tool (multi-hierarchy) under MS-Windows (Win32s) that produces mosis-code, but also simulation code for other common simulation systems.
- graphical environment for the run time system (model connections, instancing, ...)
- a bond graph modelling tool
- special graphical output modules: 3-dimensional graphics, animation etc.
- a real-time system with interface routines to several A/D-D/A boards
- interface-connections basing on the MIC to other simulation systems via PVM
- other algorithms for integration (DAEs,. ODEs), frequency analysis, optimization, visualisation etc.

Some of them will probably be presented at the congress. Several other enhancements and new features will be developed in the future.
Information about the mosis-system (and the complete package) can be obtained at the congress (exhibition area) or at the following internet-addresses:

<http://eurosim.tuwien.ac.at> World Wide Web-Suite of ARGESIM/EUROSIM
<ftp://simserv.tuwien.ac.at> Simulation software server at the TU Vienna
E-Mail: guenter@osiris.tuwien.ac.at

References

Cellier E.F.: Continuous System Modeling, Springer, New York 1991
Schuster G.: Definition and Implementation of a Model Interconnection Concept in Continuous Simulation, Dissertation, TU Wien, 1994
Breitenecker F.: Comparison of Simulation Software: Constrained Pendulum, Simulation News Europe 7/March 1993

Environment for the simulation and design of control systems

Borut Zupančič, Marko Klopčič

Faculty of Electrical and Computer Engineering
University of Ljubljana, Tržaška 25
61000 Ljubljana, SLOVENIA
Email: borut.zupancic@fer.uni-lj.si

KEY WORDS: digital simulation, simulation language, control, real time systems, fuzzy control

1. Introduction

The domain of modelling, simulation, analysis and design of control systems is known as a discipline which beside the signal processing area perhaps most drastically influenced the development of simulation hardware and software in the past. Simulation language SIMCOS, which was in its original concept general purpose CSSL type language, was expanded by numerous features, that make it extremely applicable in control design and simulation area. Although the user interface, the graphics and some other things can not be compared with some commercial simulation tools, there are some implemented ideas, which indicate important advantages in the field of control systems' design and simulation. The fuzzy controller design and simulation environment as the last extension shows the applicability also for the most advanced control approaches.

2. Basic principles of simulation language SIMCOS

The simulation language SIMCOS [1, 2, 3, 4] is a CSSL - type equation oriented language that was developed at the Faculty of Electrical and Computer Engineering in Ljubljana. It works as a compiler. The model, which is coded in CSSL syntax or is described by a graphical block oriented simulation scheme, is processed by the compiler into FORTRAN modules and a model data base. The FORTRAN modules are further processed by a FORTRAN compiler and subsequently linked with appropriate libraries into an executable simulation program. The supervisor program automatically handles all the above procedures and is able, together with a highly interactive user interface, not only to simulate the model but also to perform simple experiments (e.g. change of model constants, output specifications, function generators' breakpoints) and complex ones (e.g. parameter studies, optimization, linearization, ...).

All basic possibilities for a model description are available: The built in nonlinearities and signals, multidimensional function generators, the procedural block, basic and advanced univariable and multivariable dynamical precompiled submodels (transfer functions, state space, lead-lag, ...) in continuous and discrete forms as well as different controllers with integral wind up protection are implemented. Using a special pre-processor a hierarchical modelling is also available. Well tested and previously developed programs for particular components can be reused as submodels in higher hierarchical levels. Because of sorting algorithm an user need not to take care about the order of a model statements. Many integration algorithms (single step, multistep, low order, high order, extrapolation, stiff, ...) cover all commonly appeared numerical problems giving the simulation tool an appropriate numerical robustness.

For modelling, simulation and design of control systems the following implemented features are particularly important:

- extension of continuous simulation concept to the discrete one by so called generalized simulation operator,
- extension of the language with harware-in-the loop and man-in-the loop simulation possibilities implementing real time possibilities,
- extension of the simulation towards a complex experimentation system,
- possibilities for modular and hierarchical modelling of control systems,
- extension with design and simulation possibilities of fuzzy control systems.

3. Generalized simulation operator

Sometimes it is very efficient to simulate or to experiment with a model in discrete or in difference equation form. Such models can be easily derived from continuous linear models using well-known discretization transformations. Especially in experiments that demand many simulation runs (e.g. optimization) a lot of time can be saved using the mentioned approach.

The genaralization of the simulation was achieved by so called generalized simulation operator i.e. the integrator as the basic operator of continuous simulation was expanded to the generalized simulation operator which can be by the proper parametrization either the integrator or the discrete delay as the basic operator of discrete (difference equation) simulation.

4. Real time possibilities

Conventional computer systems are very limited for real time applications because of insufficient hardware capabilities, redundant and too complex software equipment, inefficient programming and poor input-output facilities. As SIMCOS is general purpose simulation language which works on conventional hardware (PC) with standard operating system (MS-DOS) and low cost process interface (PCI 20000), the simulation speed is of course limited. Using SIMCOS in hybrid configuration some other limitations occur so the speed is even more limited. But the use in slower simulations (or experimentations) is quite satisfactory.

The real time simulation was enabled by the following modifications and extensions:

- the generalized simulation algorithm was synchronized with real time,
- simple CPU low cost simulation algorithms were included (Euler integration algorithm, discrete simulation),
- the communication with real signals was enabled.

Program modules for data acquisition and transmission are specific for particular hardware equipment (PCI 20000). Because the language is opened to the user, it is not difficult to implement a specific hardware.

5. Experimental environment

It is well known, that simulation run is only one useful experiment in different studies. Modern tools include also more complex experiments as optimization, linearization, parameter study etc. Some experimental implementation concepts are known from the literature but compiler orientation of SIMCOS does not give the possibility to fully implement them. For this case we introduce three experimental sections (INITIAL, TERMINAL and DYNAMIC). Each model is compiled for in advance selected experiment. An experiment is treated as an execution of a particular method (simulation, optimization, linearization, parameter study, user experiment) on a model. Each higher level experiment allows also the execution of the pure simulation.

Optimization is a very useful experiment in many applications. It is particularly effective for the design of PID control schemes. An user should define a criterion function, constraints and parameters to be optimized. Linearization as another implemented experiment is also very important for control systems' design. It is used to obtain the linearized model (in state space form) from a non-linear one. It is well known, that most of design methods are limited to linear models. Parameter study is an experiment in which the influence of a parameter to simulation results can be efficiently studied. All these experiments are realized with pre-programmed INITIAL and TERMINAL sections. User experiment enables a user to create his own experiment.

6. Possibilities for modular and hierarchical modelling of control systems

Beside elementary modelling constracts, which are available in SIMCOS library, higher level modelling constructs. which can be grouped into two different classes, are available:

- precompiled objects and
- hierarchical sub models

Precompiled objects are used as standard (frequently) used blocks, written in Fortran language. The user can use a template to program his own blocks. To realise the concept of precompiled objects an efficient function analyser and a self configuration procedure of a simulation model during run time were implemented. Function analyser recognises each model function by the aid of an installation precompilled objects file. This file has an information, whether the precompiled object is statical or dynamical, continuous or discrete, univariable or multivariable, with or without delay attribute. Each function, which is called with inputs and parameters and returns outputs, is specifically pre-processed enabling appropriate sorting algorithm and efficient and correct calculations in run time. Self configuration enables that all states, derivatives and predictions are appropriately organised into common vectors of states, derivatives and predictions. On this way advanced univariable and multivariable dynamical control systems' constructs (transfer functions, state space, lead-lag, PID controller with integral wind up protection, fuzzy controller,...) in continuous and discrete forms are implemented.

Using a special pre-processor a hierarchical modelling is also available. Well tested and previously developed programs for particular components can be reused as sub models in higher hierarchical levels. The hierarchical pre-processor is in some way similar to MACRO concept in simulation languages. The internal variables of sub models are exchanged with working names so that each hierarchical model can be reused several times. Because of sorting algorithm the user needs not to take care about the order of model statements.

7. Extension with design and simulation possibilities of fuzzy control systems.

The extension with fuzzy design and simulation was implemented in two steps. As the optimization as the most important approach in conjunction with conventional PID schemes was not suitable for the design of fuzzy controllers, we decided to design a special graphically oriented design tool. Then the fuzzy controller as a precompiled block was included in the simulation library on the way, that it can automatically handle the results of the design phase. To decrease the development time, an existing fuzzy package, which contains also designing tools written in C language

was adopted. The adaptation was accomplished by an interface between fuzzy controller algorithm written in C too and simulation language SIMCOS.

Fuzzy controller designing tool Fuzzy controller design is more complex than classical PID controller design, because there are many parameters, including shape and position of segments. Because of the nature of the fuzzy controller settings, graphical representation is very helpful and simplifies design. The designing tool provides facilities for fuzzy controller design in graphical environment. User can choose among three standard shapes of segments and user defined segment type. There are four fuzzy set operations available. Rules can be defined by table or explicitly as `IF...THEN...` statements. The number of input and output variables and segments is limited by available memory. Designed fuzzy controller settings are written to a file in two available formats - ASCII and binary. Program described in this paper is written in C++. Objets were used to create the user interface.

Fuzzy controller implementation in SIMCOS Fuzzy controller implemented in SIMCOS can have two input and two output variables. The range of input variables is from -1 to +1, while output variables are given in the range from -1000 to +1000. The user has to make appropriate scaling before and after calling the fuzzy controller. If user does not define fuzzy controller properly and there exist such combinations of input variables that output fuzzy set is empty, this error is detected at runtime (during simulation). To provide user the information about the error an error flag is set and returned to SIMCOS.

Structure of fuzzy controller data The fuzzy controller settings are written to a file in binary form prepared for direct use by the fuzzy controller program. This file contains the following data:

- number of input variables,
- number of output variables,
- number of membership functions for each variable,
- number of rules,
- membership functions defined as integer arrays,
- centres of gravity and areas of output membership functions and
- rules.

Implementation of the fuzzy controller algorithm The fuzzy controller algorithm consists of three basic steps: fuzzification, calculation of output variables and defuzzification. To achieve fast fuzzification, the fuzzy membership functions are written to file as arrays of membership functions. The fuzzification is therefore performed simply by picking membership values from an array. In the next step the fuzzy decision logic is executed. The result of each rule is the centre of gravity for appropriate output segment and its area. Defuzzification is the last step in fuzzy algorithm. In this program the centre of gravity of all output segments is calculated. Its position represents the output value of the fuzzy controller.

8. Example: Design and simulation of fuzzy control of a laboratory pneumatic plant.

The model which describes the laboratory pneumatic plant is given by the transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \qquad \omega = 9.325 \qquad \zeta = 2.238$$

Y(s) ... pressure in the vessel *[mA]*
U(s) ... control signal to the pneumatic valve *[mA]*

Using the fuzzy controller designing tool and simulation approach the fuzzy controller was set experimentally. The truth table is shown in Table 1.

Table 1. Truth table (E ... control error, dE ... error derivative, mid ... middle, ze ... zero)

dE \ E	-big	-mid	ze	+mid	+big
-big	-big	-big	-mid	ze	+mid
-mid	-big	-big	-mid	+mid	+big
ze	-big	-mid	ze	+mid	+big
+mid	-big	-mid	+mid	+big	+big
+big	-mid	ze	+mid	+big	+big

Fig. 1,2 and 3 show the membership functions for control error and its derivative (E and dE) and for output controller variable U. The appropriate SIMCOS program is shown in Fig. 4.

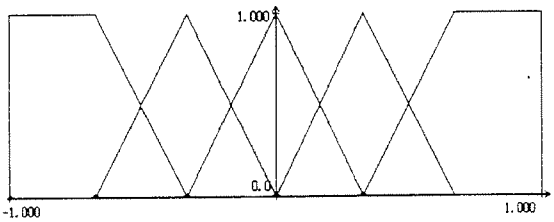


Fig. 1. Membership functions for input variable E

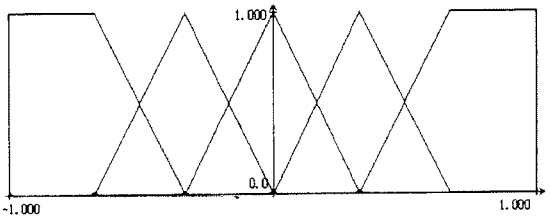


Fig.2. Membership functions for input variable dE

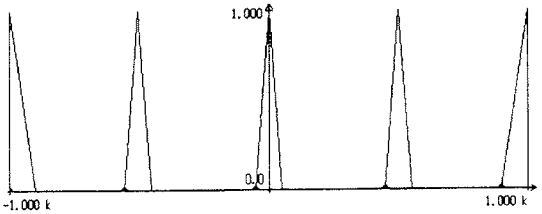


Fig. 3. Membership functions for output variable U

```
PROGRAM FUZZY
ARRAY E(2), U(2)

" reference signal
REF = STEP(T, 2.)

" error - the first input to the fuzzy controller
E(1) = REF - Y
" error derivative - the second input to the fuzzy controller
CONSTANT Tf = 0.01, Kd = 0.1
E(2)=DIFF(E(1),Kd,Tf)

" fuzzy controller - scaling of input and output variables is required
PROCEDURAL (U(1), U(2) = E(1), E(2))
    E(2) = E(2) * 5.0
    call FUZZY(U, E, FERR)
    U(1) = U(1) * 10.
END
UI = INTEG(U(1), 0.)

" second order pneumatic system
CONSTANT omega=9.325, zeta=2.238
Y=SECOR(UI, zeta,omega)

" simulation run duration and error flag
CONSTANT TFIN=5, FERR = 0
CINTERVAL CI = 0.01
TERMT T.GT.TFIN
PREPAR      Y, E(1), E(2), REF, U(1), UI, FERR
OUTPUT 20, Y, E(1), E(2), REF, U(1), UI, FERR
END
```

Fig.4. The fuzzy model of control system in SIMCOS is the following

Fig. 5 depicts the reference (REF), the controlled variable (Y) and the controller output variable UI.

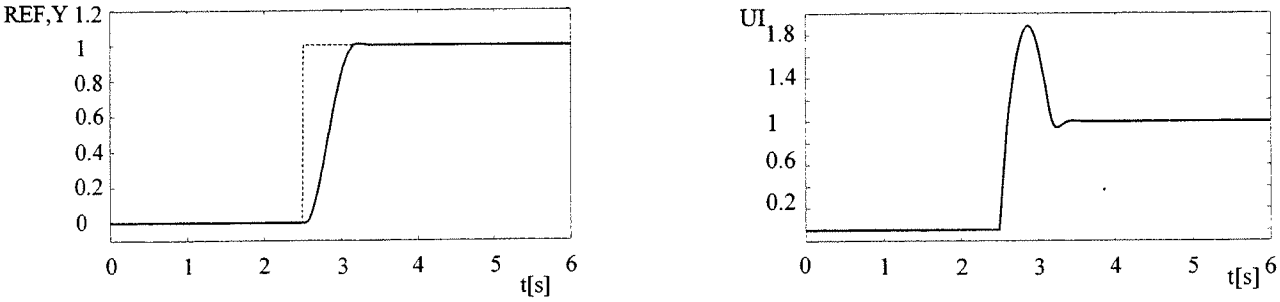


Fig. 5. Reference (REF), the controlled variable (Y) and the controller output variable (UI).

9. Conclusions

All parts of described environment were realized and tested in off line simulation studies as well as in hardware-in-the-loop experiments using real time facility. Although SIMCOS is in its nature general purpose simulation tool, the described extensions make it much more powerful and useful for the design and simulation of control systems. Fuzzy implementation as the last included feature shows, that the simulation concept is wide and open enough also for the inclusion of most advantages artificial intelligence based methods. SIMCOS is successfully used also in education process in conjunction with modelling, simulation, continuous and discrete control.

10. References

[1] Matko, D., B. Zupančič, R. Karba: *Simulation and Modelling of Continuous Systems - A Case Study Approach*, Prentice Hall Int., London, UK, 1992.

[2] Jekl, M., B. Zupančič, R. Karba: Hybrid simulation system based on digital simulation language SIMCOS, *Proceedings of the 8th symposium Simulationstechnik*, Berlin, Germany, 1993, pp. 171-74

[3] Zupančič, B., D. Matko, R. Karba, M. Šega: SIMCOS - digital simulation language with hybrid capabilities, *Proceedings of the 4th symposium Simulationstechnik*, Zürich, Switzerland, 1987, pp 205-12.

[4] Zupančič, B., D. Matko, R. Karba, S. Divjak, M. Jekl: The Concept of a new Hybrid Experimenting System, *Proceedings of the 13th IMACS World Congress on Computation and Applied Mathematics*, Dublin, Ireland, 1991, pp. 1318-19.

Distributed Object-Oriented Simulation Environment: An implementation of Time Warp using PVM

Roberto Beraldi and Libero Nigro

Dipartimento di Elettronica, Informatica e Sistemistica
Universita' della Calabria - I-87036 Rende (CS) - Italy
beraldi@telecom.deis.unical.it, l.nigro@unical.it

Abstract

This paper introduces DOSE, a development framework which facilitates the construction of a simulation model on the basis of library light-weight objects and a customizable scheduling control structure. The simulation model can be targeted for execution either on a uniprocessor machine or on a networked computing environment abstracted by PVM. As a case study, the performance of a Time Warp implementation is given.

1. Introduction

The work described in this paper aims at experimenting with parallel or distributed discrete-event simulation[1], which has the potential of allowing the exploitation of the computing resources of modern multiprocessors as well as networked computer systems. The objective is to provide an effective support to the analysis of complex and computational-intensive applications like broadband communication systems[5,6,7], wireless cellular radio networks[8] and so forth. Mathematical models used to asses the performance of such systems could require excessive simplification in the hypotheses. Simulation models are thus used to this end. The current state of ongoing work is represented by DOSE - Distributed Object-oriented Simulation Environment, a set of software tools which ensure a smooth transition from sequential to distributed simulation. A DOSE simulation model is achieved by selecting, instantiating, configuring and linking objects belonging to a reusable library specialised in a given class of application problems. The simulation model can be targeted for execution on a single processor or it can be decomposed into a collection of interacting subsystems to be allocated onto a networked system. DOSE is characterised by its openness and extensibility. New library objects, relevant to specific application areas, can be made ready to use to DOSE users. The runtime support of DOSE relies on an asynchronous light-weight architecture, DART[2,3], which has been developed for real-time applications[4]. DART is based on active objects and customizable scheduling through programming. It can easily be integrated with different communication systems and protocols. DART is truly object-oriented and permits reusable and extendible libraries of active objects to be established via inheritance. DART is available in C++ and Oberon-2. DOSE prototype implementations are in current use on concentrated DOS/Win or UNIX platforms for the simulation of communication systems[5], and in particular ATM (Asynchronous Transfer Mode) based Broadband ISDN systems. In order to meet the high speedups required for the simulation of more complex systems, e.g., mobile wireless radio networks[9], a distributed realisation of DOSE has been implemented on a standard network of UNIX workstations, possibly heterogeneous, abstracted by PVM[10] (Parallel Virtual Machine). PVM is responsible for configuration in-the-large management and inter-subsystems communication and synchronisation. Distributed simulation control centres on the Time Warp mechanisms[11], especially developed to work with DART and PVM. The implementation language is C++. The paper is organised in two parts. First the DOSE environment is summarised, by focusing on the configuration issues and the DART-based simulation level. Then the paper discusses the performance of the realised Time Warp implementation. Finally, some directions of future work are given.

2. An overview of DOSE

DOSE consists of a set of tools which help the end-user in making and executing an actual simulation model on the basis of reusable and extensible *library objects*. As shown in Figure 1, DOSE tools split logically into two classes: those responsible for configuration management (*configurer* and *analyser*) and those specifically dealing with the actualisation of a configuration (*spawner* and *actualiser*) on a networked environment handled by PVM[11].

2.1. Configuration level

A *configurer tool* is provided which supports the operations of an application-expert, e.g., through a friendly graphical user interface, for selecting, instantiating, initialising, connecting and allocating the *basic simulation objects* (**bso**s) relevant to a chosen problem domain. The configurer tool works on an *abstract representation* of a simulation model, where the basic building blocks are objects equipped with *input* and *output typed ports*. A port type essentially captures a set of messages which can be transmitted or received through the port. The type of an input port coincides with the type of the object to which the port belongs. The type of an output port is defined as the type of an object an input port of which can be linked to the output port. The number of output ports (*out-degree*) of an object is dependent on the "require part" of the object, i.e., the number and kind of communications required by the object behavior (see also later in this paper). The number of input ports (*in-degree*) is amenable to topology requirements. It can be defined at the object creation time. To facilitate the configuration of complex systems, the configurer enables *composite objects*, named *subsystems*, to be established. A subsystem is the unit of programming in-the-large and it is associated to a physical processor. A whole configuration is stored on a *configuration file*, which can also be directly generated by a standard editor. The configuration file is structured in three sections: a *create*, a *connect* and an *allocate* section. The create section lists all the bso instances involved in a simulation model. Object instantiation is expressed by the CREATE command:

CREATE OBJ <obj_id> **OF TYPE** <obj_type> **WITH**
<in-degree,value><attrib,value>... <attrib,value>

which states that a new object instance of obj_type has to be created. The new instance will be referred to by the unique identifier obj_id. Moreover, the in-degree and the internal attributes of the new object will be initialised according to the specified attribute-value list. The connect section of the configuration file captures the topology issues of the simulation model, and consists of a list of CONNECT commands, each being structured as in the following:

CONNECT OUTPORT <port_id> **OF** <obj_id> **TO**
INPORT <port_id> **OF** <obj_id>

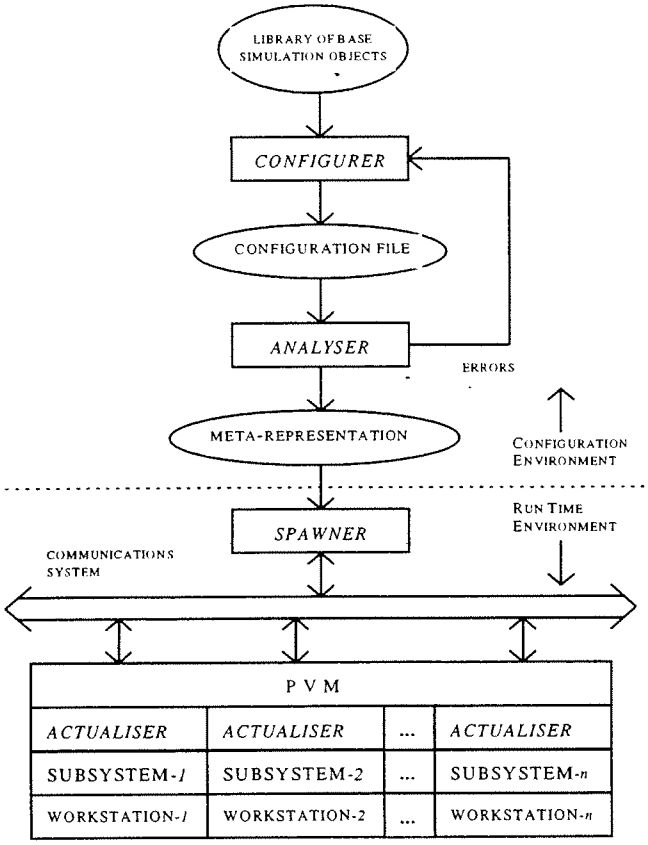


Figure 1: Dose framework.

Finally, the allocate section of the configuration file specifies the number of subsystems into which the simulation model is split, the bso instances allocated to each subsystem and the mapping of sybsystems to processors. An example of an ALLOCATE command follows:

ALLOCATE OBJECTS 0-15;17;20-25 **TO** <subsystem_id>
ON <host_name>

Object instances are named according to their unique identifiers. A single ALLOCATE command is sufficient to define the object organisation of a single subsystem. Furthermore, the following version of the ALLOCATE command:

ALLOCATE ALL TO <subsystem_id> **ON** <host_name>

can be used when the simulation model is targeted for execution on an uniprocessor machine. The *analyser* tool is responsible for checking the correctness (i.e., completeness and type-safe connections of the topology) of the configuration contained in a configuration file. Toward this, a *meta-representation* of the simulation model is built consisting of *object descriptors* where, for instance, the initialisation data are organised into suitable data structures. Object descriptors have been designed so as to be more easily translatable into the corresponding runtime representations within the different address spaces of a networked environment.

2.2. Simulation level

A *Spawner* tool is responsible for actualising a meta-representation of a simulation and of initiating the execution of the resultant distributed simulation system. The actual steps for carrying into execution a distributed simulation are summarised in the following. First all the subsystems are activated on the associated workstations of a networked system, then they are fed with the relevant object descriptors. Within each separated subsystem, an *actualiser* tool is in charge of converting object descriptors into equivalent runtime objects and actual connections. Finally, the spawner sends a message to the various subsystems for starting the simulation work. The runtime representation of a DOSE simulation model relies on an asynchronous, distributed, object-oriented architecture, DART[2,3], which has been developed for real-time applications[4]. DART is a minimal model whose design mirrors the *spirit of Oberon*[12]: "make it as simple as possible". More specifically, DART centres on the concept of light-weight active objects, which communicate one with another by message passing. The dynamic behaviour (*lifecycle*) of an object is modelled as a finite state machine which evolves through a succession of states[13]. Transition from one state to another is triggered by the arrival of an expected event (message). To each state is associated an action which is executed each time the state is entered. Action execution cannot be suspended nor pre-empted. *Action is the unit of scheduling*. Synchronisation relies upon communication and object lifecycle. A fundamental object of a DART subsystem is a programmable *scheduler* (minimal executive) which transparently collects all the message transmissions among the objects and dispatches them according to a given control strategy[14,15]. To exemplify, a basic DART scheduler rests on an *event-queue* for message buffering and an *event-loop* at each iteration of which a message is extracted from the event-queue and dispatched to its destination object. A scheduler can be customised to reason upon a *real* or *virtual clock*. As a consequence it can deliver a discrete-event simulation framework in a straightforward way. Time intervals are programmed by setting *timers*. A timer holds a *fire time* and a *timeout message* which is sent to its destination object at the timer expire time. All the timers waiting to fire, are stored into a ranked *timer-queue*, with the most imminent timer to fire at the beginning. When no more "instantaneous" messages exist on the event-queue to be dispatched, the basic simulation scheduler forces firing of the most imminent timer and adjusts accordingly the virtual clock. Flexibility of scheduler replacement is a key factor to effectively support a uniprocessor or networked execution of a DOSE simulation model. In the former case it is sufficient to adopt the basic simulation scheduler, where in the latter a distributed, e.g. based on Time Warp[11] or on a conservative strategy[1], is mandatory. DART has been implemented in C++ and Oberon-2. Active objects are equipped with event handlers[3] which allow for type-safe event communications. Objects can be organised, through inheritance, into hierarchies of reusable components. DART addresses *interoperability* by allowing different subsystems to be developed using possibly different programming languages.

3. Performance of a Time Warp implementation

Time Warp[11] (TW) is an optimistic asynchronous strategy supporting general purpose parallel discrete-event simulations (PDESs). Significant successes have been achieved from its use across a wide-range of applications[1]. Our case study concerns an implementation of TW in DOSE using a standard, networked environment managed by PVM. Key features of the realised prototype are: (a) it is a minimal and efficient kernel, whose behavior can be tuned to the application at hand; (b) it has been achieved according to the DOSE/DART approach in standard C++, hence is portable. A whole simulation model is split into a collection of subsystems (*logical processes*). Concurrency is ensured by mapping the subsystems on PVM tasks, by allocating tasks one per processor (*true parallelism*) and by executing object actions concurrently within subsystems (*apparent parallelism*). Each subsystem essentially carries a sequential simulation. Its progress in the simulated time is represented by its Local Virtual Time (LVT). Current time for the whole simulation is contained in the Global Virtual Time (GVT)

variable. Inter-subsystem messages are marked with two timestamps: the *sending time ts* (equal to the LVT of the sender sybsystem) and the *receiving time tr*, i.e. the time at which the message-event is to be handled by the receiver subsystem. Besides the event-queue and the timer-queue, the TW scheduler uses also an *input-queue* and an *output-queue*. The input-queue stores copies of received external messages, ranked by their *tr*. A receiver subsystem translates an external message into an equivalent timer whose fire time coincides with the message *tr*. The output-queue serves to annotate *undo messages* (see below) directed to subsystems to which external messages have been sent. LVTs are allowed to take different values as a consequence of load conditions and/or processor speed. Therefore it is possible to receive an external message whose *tr* is strictly below LVT (a *straggler*) giving rise to the basic TW *clock synchronisation problem*. TW needs frequent state saving in order to cope with *causality errors*, i.e., processing stragglers. A state version at a simulated time T consists essentially of copies of the object statues and of the timer-queue, performed just before any action at time T is executed.

3.1. Virtual clock synchronisation

On the arrival of a *straggler* with receiving time T, a subsystem must *roll back* to time T cancelling, on itself and on partner subsystems, all the effects of previous erroneous computation carried at times T'>T. Depending on the availability of state versions, roll back can cause re-installation of a state to a simulated time T''<=T. After that, forward computation is repeated until straggler time (*coasting phase*) then continuing by processing straggler and finally going further into future. To undo effects on other subsystems, standard TW uses *anti-messages*. An anti-message m^- either annihilates the corresponding positive-message m^+ stored in the input-queue of a subsystem S, or can trigger a roll back in S in the case m^+ has been already processed. Due the overhead normally accompanying PVM messages (e.g., pack/unpack operations for transmitting/receiving message arguments on the network of heterogeneous UNIX workstations), and considering the unpredictable latency of message transmissions on a shared LAN which can facilitate the spreading of erroneous computation[16] TW-DOSE uses "aggressive cancellation" (all the anti-messages are immediately sent for any previously sent positive-message with *ts*>T) but minimising the number of anti-messages. A single **undo** message, carrying an identification of the sending subsystem S and a timestamp T, is transmitted to every partner subsystem P for undoing effects caused by computation of all the messages sent by S to P with *ts*>T.

3.2. GVT update

A critical issue of every TW implementation is GVT management. The value of GVT is, at any real time *t**, the minimum among the LVTs and the timestamps of events that have not yet been processed at *t**, including the undo messages, of all the subsystems. Obviously, no event with a timestamp smaller than GVT will be ever rolled back, so storage associated with state versions less than GVT can be reclaimed for reuse (*fossile collection*). A frequent GVT update is clearly desirable both to conserve memory and for an early detection of simulation end. However, GVT-update operations are time-consuming and then can degrade performance due to the overhead required by the involved synchronisation. GVT is periodically updated by a subsystem acting as the *GVT manager*. GVT manager normally receives a request for a GVT-update by a subsystem which reaches a critical threshold in memory usage, measured in terms of the number of state versions and of the input-queue length. After that, the GVT-manager sends a control message to all the schedulers advising about the GVT-update. Every scheduler then stops dispatching message-events and enters a *barrier* synchronisation. All in-transit messages arriving during barrier, are actually received by the relevant subsystem at the end of the barrier and are scheduled on the timer-queue. An undo message is left pending until termination of GVT-update. The GVT-update protocol continues with each TW-scheduler proposing a value for the GVT to the GVT manager, which selects the global minimum among the received proposals and, finally, broadcasts it to all the subsystems. On receiving the new GVT value, every subsystem first reclaims fossile collection and then resumes its normal dispatching activity. The adopted strategy for GVT updates proves effective in coping also with expensive PVM memory requirements.

3.3. Simulation experiments

Performance measures of the realised DOSE TW prototype have been derived using a simple closed queueing network benchmark (CQNB), consisting of N fully connected switches. Each switch contains Q FIFO queues connected in tandem. Every queue models a non pre-emptive service station. A job arriving at a switch is served sequentially by the Q servers and is thereafter routed to one of N neighbouring switches (including itself) with equal probability. The service time of a job at a server is exponentially distributed, with minimum service time greater than zero. Initially, every switch is assigned J jobs. The parallel computing architecture used for the experiments, consists of four dedicated workstations (a Sun SPARC 1, a Sun SPARC IPC, a Sun SPARC 5 and a HP9000 Apollo) connected by a 10 Mb/s Ethernet LAN shared with other users. Experiments were conducted at times when the network was "light loaded" by other users in order to minimise their effects on the measurements. The CQNB is split into four subsystems, one per processor. At configuration time each subsystem is assigned a number of switches with associated values of Q and J. Base simulation objects involved are a *queue* with server and a *router*. Simulation results are collected on separate files, one per subsystem. Since the parallel virtual machine used is composed of workstations with sensible different computing speeds, a first concern was static *load balancing*. A CQNB with N=16, Q=20, J=32, an average service time of 3 time units, was preliminarily simulated for 1500 time units, both on a uniprocessor machine and on the four processor architecture. Different load configurations (see Table 1), labelled from L1 to L5, were considered. A load configuration specifies the number of switches allocated to each separated processor. Table 2 shows the effects of different load configurations on the completion time (i.e., the elapsed real time) and the "efficiency", the latter being defined as the ratio between the number of messages processed by the sequential simulator and the number of messages processed by the parallel simulator. As one can see from Table 2, L2 and L4 load configurations are almost equal in the completion time and efficiency. Figure 2 shows details about the distribution of the computing load among the four processors. In particular, the percentage of the messages processed by each processor with respect to the total number of messages handled by the parallel simulator is portrayed.

	Sparc1	IPC	HP9000	Sparc5
L1	4	4	4	4
L2	3	3	5	5
L3	3	3	6	4
L4	3	3	4	6
L5	2	4	5	5

Table 1: Load configurations

	L1	L2	L3	L4	L5
C.Time(sec)	104	75	81	71	86
Efficiency	0.42	0.60	0.55	0.60	0.51

Table 2: Static load balance measurements

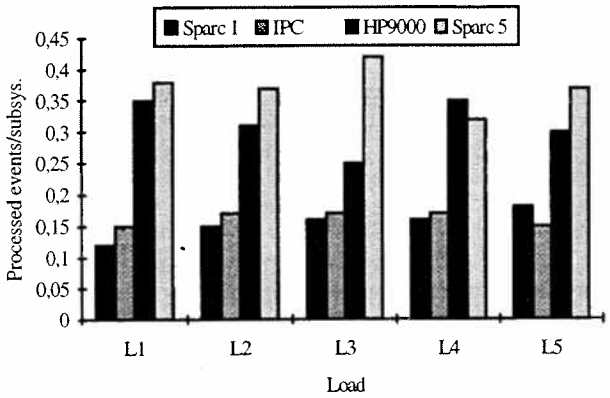


Figure 2: Processed events versus load configurations

The results in Table 2 and Figure 2 have been achieved using a state saving rate (SSR) of 6 and a maximum number of state versions (MNSV) equal to 12. A value *s* for SSR means that the subsystem status is copied on the occurrence of the *s*-th LVT change since last saved state version. The value of MNSV represents the maximum number of state versions existing before asking for a GVT-update. Figure 3 shows the completion time versus MNSV for different SSRs. It was used to select a pair <SSR, MNSV> for speedup evaluation. Figure 4 portrays the speedup as a function of Q keeping constants N=16 and J=32. Each data point has been achieved as the average of 5 runs. It is worthy of note that the sequential simulation uses the basic simulation scheduler, not TW, running on the fastest machine (Sun SPARC 5).

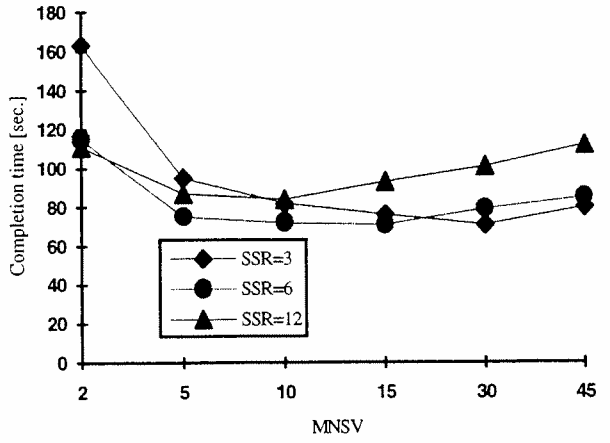


Figure 3: Completion time versus MNSVs

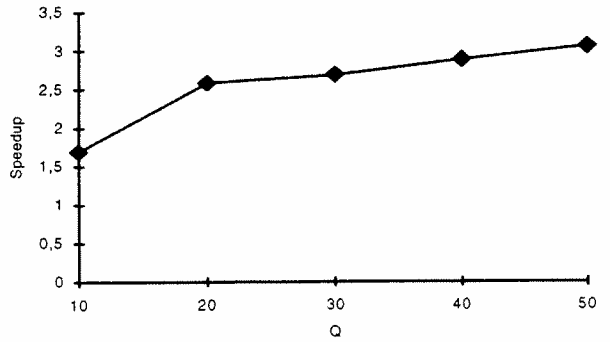


Figure 4: Speedup for CQNB (N=16,J=32,Q varying,4 processors)

4. Conclusions

Due to their low cost and high computing power, networked environments are today widespread. Their usefulness in general parallel computing is enhanced by recent high-speed communication means (e.g., based on ATM or FDDI). The effective use of a networked system is facilitated by a standard software layer like PVM which simplifies programming by abstracting details about protocols, data conversion operations, synchronisation concerns and so forth. All of this is of obvious interest in supporting parallel discrete-event simulation (PDES), provided adequate development tools are available. With respect to concurrent applications, PDESs add the difficulties of coping with causality errors which in turn introduce hard synchronisation problems. This paper presents DOSE - Distributed Object-oriented Simulation Environment - which has a flexible configuration tool allowing the application expert to configure a simulation model on the basis of reusable objects, and of spawning the model for execution either on a uniprocessor machine (Dos/Win or Unix platform), or on a networked context managed by PVM. The following factors are noteworthy: (i) DOSE relies on an asynchronous software model based on light-weight objects in the absence of pre-emption and of context-switch operations; (ii) DOSE has a programmable runtime support (scheduler); (iii) DOSE is programmed in the popular language C++. The paper furnishes performance data of a developed Time Warp mechanism. The ongoing activity is geared toward: (a) optimising the Time Warp implementation, e.g., by introducing an incremental object state saving technique; (b) completing the realisation of library objects for the parallel simulation of complex systems like mobile wireless networks.

Acknowledgements

The authors wish to thank B. Kirk and S. Marano for their helpful comments during preliminary drafts of this paper.

References

1. Fujimoto R.M. Parallel Discrete Event Simulation, CACM, **33**(10), 1990, 31-53
2. Kirk B. and Nigro L. A Distributed Architecture for Real-Time, in Oberon-2, in *Advances in Modular Languages*, P. Shulthess (ed.), Universitatsverlag ulm GmbH, 1994, pp. 325-366.
3. Nigro L. A real-time architecture based on Shlaer-Mellor object lifecycles, *JOOP*, **8**(1), March-April 1995, 20-31
4. Nigro L. and Tisato F. An object-based architecture for real-time applications, *in Advances in Object-Oriented Software Engineering*, Mandrioli D. and Meyer B. (eds.), Prentice-Hall, 1992.
5. *IEEE Communication Magazine*, issue on Network level modelling and simulation, March 1994
6. Beraldi R., Iera A., Marano S. and Sestito R. A per-burst Combined Policing Mechanism for ATM Networks in the case of ON/OFF and Loss Sensitive Sources: The DTAT, in *Proc. of Third Int. Conf. on Computer Comm. and Networks (ICCCN)*, San Francisco (CA), 1994.
7. Beraldi R., Iera A., Marano S., Ruggi R., Buffer Statistical Multiplexing for Bursty data Calls in ATM Networks, in *Proc. of the IEEE Singapore ICCS*, Singapore, 14-18 Nov. 1994.
8. Beraldi R., Iera A., Marano S. and Salerno P. A New dynamic Reservation Multiple Access Protocol for Supporting Multimedia Traffic in Third Generation Cellular Systems, *in Proc. of the IEEE Singapore ICCS*, 14-18 Nov., 1994.
9. Edbom G. and Stjernholm P. Simulation of Cellular Networks, in *Proc. of the IEEE/VTS, 44th Vehicular Technology Conference*, 1994.
10. Geist A. et al. *PVM:Parallel Virtual Machine - A Users Guide and Tutorial for Networked Parallel Computing*, (The MIT Press, 1994)
11. Jefferson D. et al. Distributed simulation and the Time Warp operating system, in *Proc. of the 11th ACM Symposium on Operating System Principles*, 1987, 77-93.
12. Wirth N. A plea for lean software, *IEEE Computer*, **28**(2), pp. 64-68, February 1995.
13. Shlaer S. and Mellor S.J. *Object lifecycles: modeling the world in states*, (Yourdon Press, 1992)
14. Nigro L. Control extensions in C++, *JOOP*, **6**(9), pp. 37-47, Feb. 1994.
15. Nigro L. and Veneziano G. Control abstractions in Modula-2: a case study using advanced backtracking, Informatica, 18(2), June 1994, 229-243
16. Carothers C.D., Fujimoto R.M., England P. Effects of Communication Overheads on Time Warp Performance: An Experimental Study, *8th Workshop on Parallel and Distributed Simulation*, Edinburgh, July 1994, 118-125.

Simulation-Based Optimization of Production Systems by SENSIM

Elke Stief
Institute for Manufacturing Automation and Production Systems
Prof. Dr.-Ing. K. Feldmann
University Erlangen-Nuremberg, Germany
91058 Erlangen, Egerlandstr. 7-9, Germany
E-mail: stief@faps.uni-erlangen.de

ABSTRACT

This paper describes the tool SENSIM which supports the optimization of production systems by simulation using design of experiment (DOE) methods. The tool has been developed by the Institute for Manufacturing Automation and Production Systems. The paper is structured as followed: First the implementation of DOE methods and their connection to the modular system SIMULATION is described. The performance of the DOE methods implemented in the system will then be demonstrated using a case study.

INTRODUCTION

Due to prevailing buyer's markets and to a growing dynamic of innovations, the economic situation for companies has changed fundamentally in recent years. This has led to a variety of changes and to an increasing complexity of production systems in the companies.

In order to support the planning of production facilities simulation has gained general recognition as a suitable tool. Simulation can for example be used to evaluate alternatives for improvement according to identified negative trends. In addition it is possible to obtain predictions of future system behaviour.

In general, a large number of experiments have to be carried out to optimize plant performance. With regard to the growing dynamic in the planning stage of both production and product it is necessary to increase the effectiveness of simulation. This can be supported by SENSIM, a tool which enables the user to work according to a fixed pattern implying the required experiments.

When creating a tool the implemented two issues are important: first the methods used and second the procedure of experimentation. Therefore, the tool SENSIM is based on a scientific approach. The design of experiments is part of it.

DESIGN OF EXPERIMENT

The DOE methods are considered to be the proprietary tool for improvement and optimization of processes and products according to given objectives. Information on how to improve an industrial process can be obtained by going through a cycle of minor modifications, or variants, of the current process and repeatedly running these in sequence. The methods can be used as a way to avoid large numbers of time consuming experiments.

Factorial design is a particularly useful method which may be used to investigate either quantitative or qualitative variables. The following Factorial Designs have been implemented in the tool:

- 2n-Factorial Designs.
- 2n-p-Factorial Designs.
- The Taguchi Method.
- The Shainin Method.

The 2n factorial experiments provide information on all possible two factor interactions if the experiment is designed properly. The Taguchi Method and the Shainin Method are special factorial designs.

IMPLEMENTATION OF SENSIM

The module SENSIM has been developed to automatically support the process of model optimization by DOE methods. It enables the user to specify the optimization problem to which the optimization process is to be applied.

Due to the scientific approach the tool is based on the following requirements which have been taken into consideration:

1. Recognition of an existing problem.
2. Formulation of the problem.
3. Agreeing on factors and levels to be used in the experiment.
4. Specifying the variables to be included.
5. Definition of the inference space of the problem.
6. Random selection of the experimental units.
7. Assignment of treatments to the experimental units.
8. Outline of the analysis corresponding to the design before the data are taken.
9. Collection of the data
10. Analysis of the data.
11. Conclusions.
12. Implementation.

The design of experiment methods are dealt with in the steps 5, 6 and 7. The sections prior to these three describe the preparation of the design, and section 8 allows the experimenter to modify the design before putting it into operation.

The architecture of SENSIM consists of the following three building blocks

- REVERS,
- a graphical user–interface and
- an interface for coupling the module with the modelling tool SIMULATION.

The building–block REVERS provides the four kinds of factorial designs mentioned above. REVERS runs the implemented program in accordance to the input received from both the design selected by the user and the results of the experiments received from the system SIMULATION.

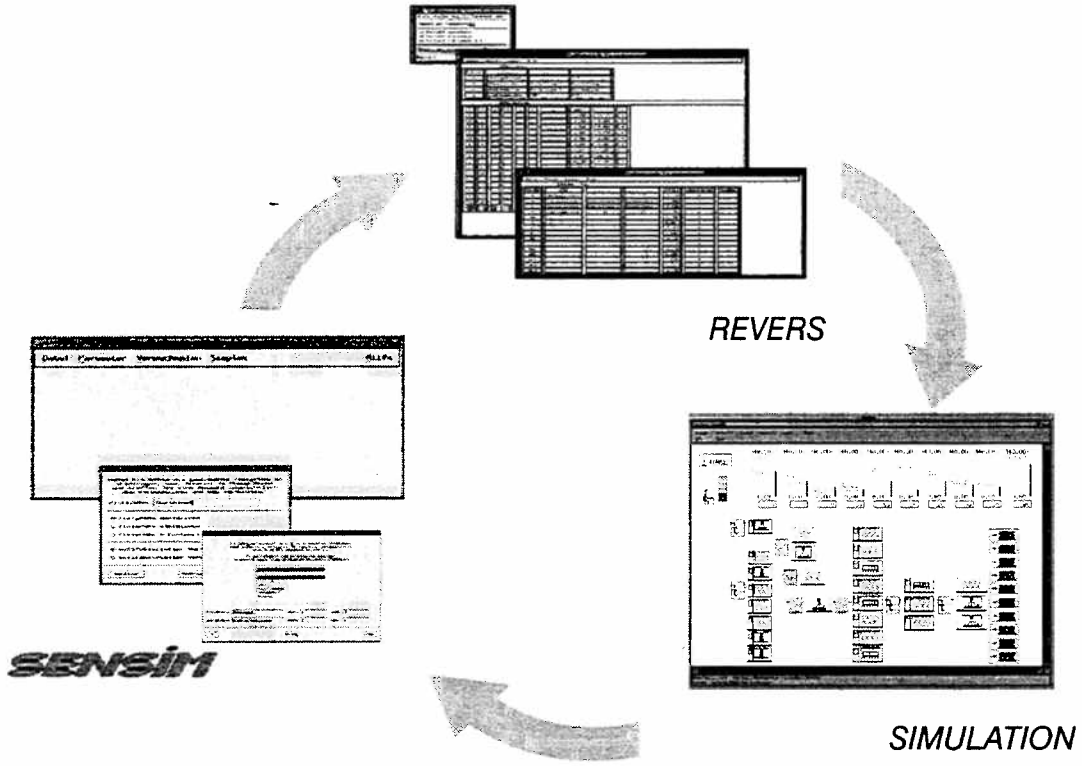


Fig 1 Interplay between SENSIM and the related tools REVERS and SIMULATION

When working with the module, the user has to handle a distinct number of input parameters as well as to cope with a high quantity of output data, generated by the tool SIMULATION. In order to get both an easy survey and a user friendly structure the user–interface has been divided into an input part, a control part and an output part.

The *input–part* enables the user to choose a factorial design. In addition to this the variables to be examined can be selected and a limited number of possible levels for each

of these variables and the objective function can be identified. This objective function however may be any performance parameter that can easily be received as a result of the experiments. The system offers defaults of useful performance parameters of the objective functions and gives support when fixing the variables and their possible levels. The input is passed forward to the building-block REVERS. Informations about relevant variables and components of the simulation model are needed for generation of batches and preparation of the simulation experiments. The task of the *output-part* is to visualize the results and to give recommendations for optimized settings for the parameters of the simulation model. On the one hand the *control part* of the interface enables the user to stop the running optimization process and to restart the process if necessary and on the other hand to skip to a help mode, which supports the user to handle the tool.

The interface couples the tool SENSIM with the modular system SIMULATION in order to run experiments.

CASE STUDY

The experience of using the tool SENSIM is presented by a case study. In this example the tool SENSIM and the simulation was used to optimize the behaviour of a production system. The model of the production system consists of about 20 components and approximately 30 different jobs are modelled. The object of the study was to obtain initial experience on the optimization of model-behaviour using simulation and DOE-Methods. Key concerns were to gain knowledge about the time involved by running the tool and about the validity of the results.

REFERENCES

Box, George, and Draper, Norman R., *Evolutionary Operation*, John Wiley & Sons, Inc. Petersen,1969.

Harro, *Grundlagen der Statistik und der Statistischen Versuchsplanung*, Ecomed Verlagsgesellschaft mbH, 1991.

Anderson, Virgil L. and McLean, Robert A., *Design of Experiments - A Realistic Approach*, Marcel Dekker, Inc., New York, 1972.

Abels, S.,*Modellierung und Optimierung von Montageanlagen*, Ph.D.,Fertigungstechnik - Erlangen; 37,Hanser,1993.

Abels, S., and Stief, E., *Knowledge-Based Optimatization of Simulation Studies*, in Proceedings of the Conference on Modelling and Simulation 1994 ESM'94, Barcelona, Spain, pp.186-190,1994.

`omsim2maple` – A translation utility for OmSim simulation code

James A. Sørlie

S3–Automatic Control; Royal Institute of Technology; S-100 44 Stockholm
`http://www.s3.kth.se/~sorliej/`

Abstract: The purpose of this contribution is to present a tool for translation of OmSim simulation code into Maple, as well as to illustrate some potentials of symbolic manipulation when applied in modeling and identification. OmSim simulation code is effectively a sorted set of modeling equations. The reported utility translates these equations into Maple statements, which then allows symbolic reduction and manipulation of the equations. A principle motivation in this work has been the desire to use analytical derivatives in parameter estimation. Furthermore, Maple provides facilities for optimized C-language code generation. To illustrate its use, the translator and symbolic manipulation are used in deriving an extended Kalman filter and generating numerically optimized C-language subroutines. Currently, most commercially available code-generators preclude the use of symbolic manipulation.

Keywords: software, symbolic manipulation, code generation, extended Kalman filter, grey-box modeling

1 Introduction

An ongoing project at the S3 department in Stockholm is the development of methodologies for Grey-Box Modeling and Identification – an approach to building mathematical models based on both mechanistic knowledge of the process, as well as empirical measurement data; *cf.* [1]. An identification tool-kit, IdKit, has been developed [2] consisting of application independent identification software. In using the tool-kit, a model builder provides model structures, *i.e.* modeling equations, in the form of C-language subroutines. The original objective behind the immediate project was to facilitate the definition and management of model structures. Through time, the focus has changed as a result of an awakening to the potentials of symbolic manipulation.

Omola and OmSim are respectively an object-oriented modeling language and a simulation environment which have been developed by the Computer-Aided Control Engineering (CACE) group in Lund [3–6]. The modeling language provides a high-level *acausal* data description of a dynamic object. The simulation environment provides a model compiler as well as state-of-the-art facilities for numerical simulation. Due to its developmental status, OmSim currently lacks export facilities of a low-level data description, *i.e.* the equations of the compiled model. This is unfortunate since such a modeling language is ideal for managing multiple realizations of a given model. This is of particular interest when investigating alternative model structures via a grey-box methodology; *cf.* [7, 8].

It would thus appear that in its present state, OmSim is a closed software system. However, as part of its debugging facilities, the software does provide a means of "dumping" the compiled simulation program code. This program code itself resembles the uninteresting primitives of a reverse-Polish-notation calculator. Of interest are the corresponding modeling equations which are interspersed throughout this debugging output as program comments. These equations appear in computational order and, rather serendipitously, the grammatical syntax of the Omola equations is strikingly reminiscent to that of the computer algebra program Maple [9]. This last observation, combined with knowledge of the configurable code generation capabilities in Maple [10], led us to investigate an interface between the two program packages.

2 Program Description

In order to keep matters simple, our objective has been, at least initially, to translate only continuous-time elements of the Omola language. This reflects the fact that, aside from transport delays, much of the mechanistic information used in grey-box modeling is most naturally expressed as continuous-time relations.

Two versions of the utility have been developed. The tool was initially prototyped as a Unix shell script in order to verify the feasibility of the idea. A second version has been developed using C++ and GNU's `flex` and `bison` utilities. This later version's sophistication was deemed necessary in order

```
1  #!/usr/local/bin/bash -norc
2  # FILE:  omsim-filter.sh -- Convert an OmSim log-file to Maple statements.
3  # $Id: omsim-filter.sh,v 1.7 1995/03/08 17:06:30 sorlie$
4  #-----------------------------------------------------------------
5  # Step 1:  Convert the OmSim simulation code to Maple definitions.
6  #-----------------------------------------------------------------
7  # Delete from the log-file all lines except the symbolic equations.
8  # These lines have a program-code line-number followed by two percent
9  # symbols, as well as containging an equals symbol.
10 #-----------------------------------------------------------------
11 cat $*  | sed -n '/^.*%%.*=.*$/p' |\
12 #-----------------------------------------------------------------
13 # Remove the line number, percent symbols and leading whitespace.
14 # Convert the end-of-line character to a Maple command terminator (:).
15 # Convert any equalities (=) into Maple bindings (:=).
16 # Convert the period in Omola names to an underscore.
17 # Finally, comment out equations designating the OmSim inputs.
18 # This makes the script-output suitable as "include" input for Maple.
19 #-----------------------------------------------------------------
20 sed 's/^.*%%[    ]*\(.*\)$/\1\:/;
21      s/ = / := /g;
22      s/\.\([^0-9]\)/_\1/g;
23      s/^\(.*\)\*input\*.*\,.*\,[ ]*\(.*\)\,.*$//'        |\
24 #-----------------------------------------------------------------
25 # Convert conditional assignment statements to piecewise function calls.
26 #-----------------------------------------------------------------
27 sed  's/^\(.*:=.*\) if \(.*\) then \(.*\) else \(.*\):$/\1 \
28 piecewise\(2,\2,\3,\4\):/g' | cat
29 #-----------------------------------------------------------------
30 # End of processing Step 1.
31 #-----------------------------------------------------------------
32 # Step 2:  Convert the OmSim parameter dump to Maple definitions.
33 #-----------------------------------------------------------------
34 # Find any assignment (:=) or relation (=) statements in the log file,
35 # excluding those in the comments to the simulation code.  The result
36 # should be the parameter mapping and propagation of parameters
37 # in computational order, found in the Parameter debug output.
38 #-----------------------------------------------------------------
39 cat $*  | egrep -v '%%' | egrep '[ :]= ' |\
40 #-----------------------------------------------------------------
41 # Convert the end-of-line character to a Maple command terminator (:).
42 # Convert any equalities (=) into Maple bindings (:=).
43 # Convert the period in Omola names to an underscore.
44 #-----------------------------------------------------------------
45 sed 's/^\(.*\)$/\1\:/g;
46      s/ = / := /g;
47      s/\.\([^0-9]\)/_\1/g;'     | cat
48 #-----------------------------------------------------------------
49 # End of processing Step 2.
50 #-----------------------------------------------------------------
```

Figure 1: Listing of `omsim2maple` shell script.

to deal more directly with the complexity of the Omola grammar.

To illustrate the basic mechanisms of the translator, and its sheer simplicity, we include a listing of the shell-script prototype in Figure 1. Worth noting is the translation of conditional statements using the Maple `piecewise` function (lines 27–28). Notice however that the shell script does not account for the possibility of nested conditional statements in Omola. This problem, along with the translation of relation-operator statements and various matrix operations are examples of the complexity which warranted the development of the C++ version of the translator.

2.1 Creating the Translator Input

The input to the translator is a debugging "dump" which the user creates using the OmSim simulator and its logging facilities. Upon successful instantiation of an Omola model (see [5]), one turns logging on and then selects the simulator's Debugging → Parameter part and Debugging → Simulation code menu selections. (It is assumed that the log-file is initially empty.) At this point, one may run the translator, specifying the log-file as input and redirecting the standard output to a file. This output is suitable for loading

```
1  SingleBlockModel ISA IdKit::SignalModel WITH
2  signals:
3    Wv, U,  Wy    ISA IdKit::SignalVector;        % Inputs.
4    Y            ISA IdKit::SignalVector;        % Outputs.
5    Xy           ISA IdKit::SignalVector;        % States.
6    dXy_dt       ISA IdKit::SignalVector;        % State derivatives.
7  parameters:
8    P            ISA IdKit::ParameterMatrix;     % Physical parameters.
9  END;
```

Figure 2: A "wrapper" class in Omola.

into Maple using Maple's File → Include ... menu selection.

2.2 Parameter and Signal Names

Once loaded into Maple, all variable signals and unbound parameters are available for symbolic manipulation. Logically, their names in the translated equations are fully qualified according to the hierarchical class structure of the Omola model. To facilitate processing in Maple, it proves useful to encapsulate the Omola model in a "wrapper" class which defines both parameter and signal name mappings. The wrapper class provides top-level (meaning unqualified) references to the parameters and signals of interest. Note that this encapsulation mechanism places no restrictions on the hierarchic structure of the actual Omola model.

An example of a wrapper class suitable for use with IdKit is shown in Figure 2. In using this wrapper, the translator's output will include Maple assignments for the signals `Wv`, `U`, `Wy`, *etc.*, and the unbound parameters `P`. It is these short names which one references Maple. The use of the wrapper class will be clarified in the following section's example.

3 Symbolic Manipulation

We intend to illustrate the use of the translator along with some of the potentials of symbolic manipulation of the model equations. In [3, 11], symbolic manipulation is discussed in the context of model compilation and index reduction, *i.e.* reduction of differential-algebraic equations to ordinary differential equations. In [12], symbolic manipulation is used in linearization and the *future* potentials of code generation are mentioned. In what follows, we assume we have run the `omsim2maple` translator on the OmSim log-file and loaded the model equations into Maple.

3.1 Extended Kalman Filter

Here we develop the equations for a continuous-discrete extended Kalman filter (EKF); *cf.* [13]. The system for this example is based on a drum-boiler/turbine model of a power plant [2]. Space does not permit going into the details of the model. Suffice it to say that the plant consists of two

```
1   IdKitDrumBoiler ISA IdKit::SingleBlockModel WITH
2     dboil ISA DrumBoilerSimulation;
3   signal_model:
4     Wv.dim := 2;  U.dim  := 2;  Wy.dim := 2;
5     Xy.dim := 4;  Y.dim  := 2;  dXy_dt.dim := 4;
6
7     Wv = dboil.StateDist.Wv;
8     Xy = [dboil.StateDist.Xv; dboil.Process.Xz];
9     dXy_dt = [dboil.StateDist.Xv'; dboil.Process.Xz'];
10    U  = dboil.Process.U;
11    Wy = dboil.Observer.Wy;
12    Y  = dboil.Observer.Y;
13  parameter_matrix:
14    P.rowdim := 7;  P.coldim := 2;
15    dboil.Process.Tau     := trans( P[1,1..2] );
16    dboil.Process.Alpha4  := P[2,1];
17    dboil.Process.Xz0     := trans( P[3,1..2] );
18    dboil.Process.Uss     := trans( P[4,1..2] );
19    dboil.StateDist.sigma := trans( P[5,1..2] );
20    dboil.Observer.sigma  := trans( P[6,1..2] );
21    dboil.Process.NonLin  := int( P[7,1] );
22  END;
```

Figure 3: Use of the Omola wrapper class.

```
1   A := jacobian(  dXy_dt, [ x[j]  $ j=1..vectdim(Xy) ] ):
2   G := jacobian( dXy_dt, [ wv[j] $ j=1..vectdim(Wv) ] ):
3
4   # Time-update of the estimated state.
5   convert(dXy_dt,list):
6   dx_hat_dt := convert(subs({'wv[j]=0' $ j=1..vectdim(Wv)},"),vector);
7
8   # Time-update of the prediction error covariance.
9   Rx := array( symmetric, 1..vectdim(Xy), 1..vectdim(Xy) ):
10  dRx_dt := evalm( A &* Rx  +  Rx &* transpose(A)  + G &* transpose(G) ):
```

Figure 4: Time-update equations in Maple.

state equations augmented by a linear shaping filter which models two state disturbances as a diffusion process. The stochastic inputs Wv and Wy are normalized continuous and discrete Gaussian white noise, respectively.

$$\frac{d}{dt}x(t,x,u,wv,P) = \begin{bmatrix} 0.002wv_1 \\ 0.003wv_2 \\ \frac{4.20u_1 - 0.99x_3u_2}{P_{1,1}} + P_{5,1}x_1 \\ \frac{0.2112x_3u_2 - 0.9600x_4}{P_{1,2}} + P_{5,2}x_2 \end{bmatrix}$$

$$y(t_k,x,u,,wy,P) = \begin{bmatrix} 1.056P_{2,1}x_3u_2 \\ +0.32\,(15 - 15P_{2,1})\,x_4 \\ +P_{6,1}wy_1 \\ x_3 + P_{6,2}wy_2 \end{bmatrix}, k \in N$$

A simulation model of these equations has been programmed in Omola. The signal model "wrapper" for the model is shown in Figure 3. Note in line 21 the mapping for an Omola "realization parameter" [4, Ch.8]. This parameter may be used to switch between nonlinear and linearized realizations.

Time-update equations: The calculation of the time-update equations in Maple is shown in Figure 4. The prediction error covariance is reduced *symbolically*, rather than performing the matrix multiplications numerically. Note that here we use continuously evaluated Jacobians rather than constant Jacobians (*i.e.* evaluated instead using the results of the last measurement update) across the interval between measurements. Technically this is

```
1   C  := jacobian( Y, [x[j]  $ j=1..vectdim(Xy) ] ):
2   H  := jacobian( Y, [wy[j] $ j=1..vectdim(Wy) ] ):
3
4   # Innovation vector, i.e. the prediction error or residuals.
5   convert(Y,list):
6   convert(subs({'wy[j]=0' $ j=1..vectdim(Wy)},"),vector):
7   innovations := evalm( vector(['y[j]' $ j=1..vectdim(Y) ])-" );
8
9   # Covariance of the innovations, and its square root.
10  Ry := evalm( C &* Rx &* transpose(C) + H &* transpose(H) ):
11  S  := CholeskyFactorization(Ry):
12
13  # The Kalman gain matrix.
14  K  := evalm( Rx &* transpose(C) &* transpose(inverse(S)) ):
```

Figure 5: Meas.-update equations in Maple.

necessary because of the coupling of the differential equations for the state estimate and prediction error covariance. These couplings are important if the measurement sample interval is long with respect to the dynamics of the system, particularly in the case of stiff systems; *cf.* [2].

Measurement-update equations: Using the standard filter equations, the expression for the Kalman gain involves the inverse of a full matrix. Taken symbolically, this introduces complexity which leads enormous expressions for the measurement update equations. Even for a state vector of moderate dimension (four in this example), it is far more efficient to evaluate the matrix multiplications *numerically* rather than symbolically. An additional reason for deferring to numerical evaluation of the measurement update equations is the desire to include a mechanism for multi-rate sampling and missing measurements; *cf.* [14]. Also, use of the square-root information variant of the Kalman filter (as is shown in Figure 5 and implemented in IdKit), requires instead only the inverse of a triangular matrix.

3.2 Code Generation

Here we use symbolic manipulation for the conversion of the analytically derived results into an *optimized* form, suitable for numerical application. Figure 6 shows a template for automatic generation of a C-language subroutine, using the `macroC` share-ware package [10] in Maple. Figure 7 shows the output, the time-update equations for the extended Kalman filter. Note again the use of Maple's `piecewise` function in handling the Omola realization parameter.

Plainly, this is nothing more than an application of already existing tool. Our contribution is this: There are a number of simulation packages on the market, that provide "configurable" low-level code generation capabilities. All these packages are geared towards accelerating *model* simulation. However, equation export at this low a level precludes the further use of the equations in symbolic processing. As the EKF example illustrates, only

```
genC( [ [defineC,'piecewise(dummy,cond,expr1,expr2)','(((cond)?(expr1):(expr2))'],
        [ commentC, '---------------------------------'],
        [ commentC, ' predictor time update equations '],
        [ commentC, '---------------------------------'],
        [ functionm, void, EKF_time_update,
          [ [ double, ['*x', '*u', '**Rx']],
            [ double, ['*dx_hat_dt'      ]],
            [ double, ['**dRx_dt'        ]]],
          [ [ commentC, ''                 ],
            [ commentC, ' state estimate '  ],
            [ matrixm, 'dx_hat_dt', dXy_hat_dt],
            [ commentC, ''                 ],
            [ commentC, ' erorr covariance ' ],
            [ matrixm, 'dRx_dt', dRx_dt      ]] ]] ):
```

Figure 6: A `macroC` template in Maple.

```
# define piecewise(dummy,cond,expr1,expr2) ((cond)?(expr1):(expr2))
/*---------------------------------*/
/* predictor time update equations */
/*---------------------------------*/
void EKF_time_update(x,u,Rx,dx_hat_dt,dRx_dt)
double *x,*u,**Rx;
double *dx_hat_dt;
double **dRx_dt;
  {
  double t1,t2,t4,t6,t10,t11,t15,t21,t23;
  double t7,t12,t14,t17,t19,t24,t28,t32,t35;

  /* state estimate */
  t1 = (P[6][0] != 0);
  t2 = x[2]*u[1];
  t4 = 1/P[0][0];
  t6 = P[4][0]*x[0];
  t10 = P[3][0]/P[3][1];
  t11 = x[2]-140/33*t10;
  t15 = u[1]-P[3][1];
  t21 = 1/P[0][1];
  t23 = P[4][1]*x[1];
  dx_hat_dt[0] = 0;
  dx_hat_dt[1] = 0;
  dx_hat_dt[2] = piecewise(2,t1,3/10*(14*u[0]-33/10*t2)*t4+t6,
    -99/100*P[3][1]*t4+t11+21/5*t4*(u[0]-P[3][0])-21/5*t10*t4*t15+t6);
  dx_hat_dt[3] = piecewise(2,t1,8/125*(33/10*t2-15*x[3])*t21+t23,
    34/161*P[3][1]*t21+t11-24/25*t21*(x[3]-14/15*P[3][0])+112/125*t10*t21*t15+t23);

  /* erorr covariance */
  t1 = (P[6][0] != 0);
  t2 = piecewise(1,t1,P[4][0],P[4][0]);
  t4 = 1/P[0][0];
  t7 = piecewise(1,t1,-99/100*u[1]*t4,-99/100*P[3][1]*t4);
  t12 = piecewise(1,t1,P[4][1],P[4][1]);
  t14 = 1/P[0][1];
  t17 = piecewise(1,t1,132/625*u[1]*t14,34/161*P[3][1]*t14);
  t19 = piecewise(1,t1,-24/25*t14,-24/25*t14);
  t21 = t2*Rx[0][3]+t7*Rx[2][3]+t12*Rx[1][2]+t17*Rx[2][2]+t19*Rx[2][3];
  t24 = t2*Rx[0][0]+t7*Rx[0][2];
  t28 = t12*Rx[1][1]+t17*Rx[1][2]+t19*Rx[1][3];
  t32 = t12*Rx[0][1]+t17*Rx[0][2]+t19*Rx[0][3];
  t35 = t2*Rx[0][1]+t7*Rx[1][2];
  dRx_dt[0][0] = 1/250000;
  dRx_dt[2][2] = 2*t2*Rx[0][2]+2*t7*Rx[2][2];
  dRx_dt[2][3] = t21;
  dRx_dt[0][2] = t24;
  dRx_dt[3][1] = t28;
  dRx_dt[0][3] = t32;
  dRx_dt[3][0] = t32;
  dRx_dt[2][1] = t35;
  dRx_dt[3][2] = t21;
  dRx_dt[1][2] = t35;
  dRx_dt[2][0] = t24;
  dRx_dt[1][3] = t28;
  dRx_dt[3][3] = 2*t12*Rx[1][3]+2*t17*Rx[2][3]+2*t19*Rx[3][3];
  dRx_dt[0][1] = 0;
  dRx_dt[1][0] = 0;
  dRx_dt[1][1] = 9/1000000;
  }
```

Figure 7: C-code for EKF time-updates.

through further processing is it possible to get truly optimized *predictor* simulation code. To our knowledge, none of the currently available packages support symbolic manipulation and, thus, preclude the possibility of simulating an EKF predictor.

4 Conclusions

We have presented a tool which translates a debugging "dump" from a modeling/simulation package into sorted modeling equations, suitable for symbolic manipulation. To demonstrate the tool's utility, an extended Kalman filter (EKF), which requires analytic derivatives, was derived through symbolic manipulation. Numerous commercial simulation softwares currently on the market now support automatic code generation, but not symbolic manipulation. As we have shown, today's computer algebra packages are more than capable of performing the final conversion to the low-level C-language code. We see the need for a choice in the level of equation export in order to fully realize the potentials of symbolic analysis and code generation. Consider, for example, implementing other *higher-order* nonlinear filter approximations [13].

References

[1] T. Bohlin. *Interactive System Identification: Prospects and Pitfalls.* Springer-Verlag, 1991.

[2] S. F. Graebe. *Theory and Implementation of Gray Box Identification.* Ph.D. thesis TRITA–REG–9006, Royal Institute of Technology, Stockholm, Sweden, 1990.

[3] S. E. Mattsson, M. Andersson, and K. J. Åström. Object-oriented modelling and simulation. In D. A. Linkens, editor, *CAD for Control Systems*, pages 31–69. Marcel Dekker, 1993.

[4] B. Nilsson. *Object-Oriented Modeling of Chemical Processes.* Ph.D. thesis TFRT–1041–SE, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1993.

[5] M. Andersson. *Object-Oriented Modeling and Simulation of Hybrid Systems.* Ph.D. thesis TFRT–1043–SE, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1994.

[6] M. Andersson. OmSim and Omola Tutorial and User's Manual. Technical Report TFRT–7504–SE, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1994. Manual and software available at URL: `http://control.lth.se/~cace/`.

[7] T. Bohlin. Derivation of a 'designer's guide' for interactive 'grey-box' identification of nonlinear stochastic objects. *Int. J. Control*, 59(6):1505–1524, 1994.

[8] J. A. Sørlie. On a general computer-aided methodology for building models of industrial processes. Technical Report IR–S3–REG–9504, Royal Institute of Technology, Stockholm, Sweden, 1995.

[9] B. W. Char, K. O. Geddes, G. H. Gonnet, B. L. leong, M. B. Monagan, and S. M. Watt. *Maple V Language Reference Manual.* Springer-Verlag, 1991.

[10] P. Capolsini. *MacroC - génération de code C depuis Maple.* INRIA - Université de Nice, Laboratoire I3S, June 1992. In Maple, type `with(share): readshare(macroC,numerics):`.

[11] F. E. Cellier and H. Elmqvist. Automated formula manipulation supports object-oriented continuous system modeling. *IEEE Control Systems Magazine*, 13(2):28–38, 1993.

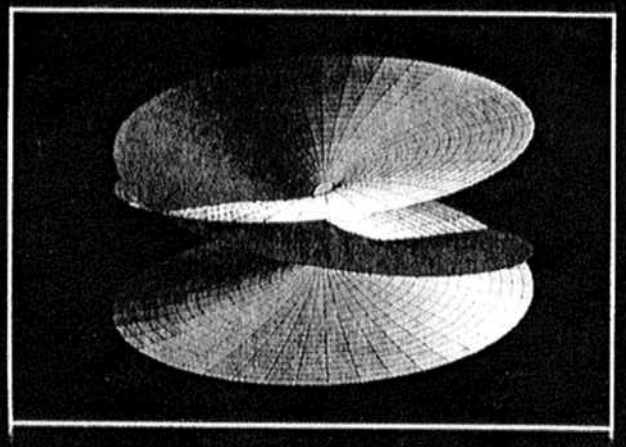
[12] P. J. Gawthrop. Symbolic modeling in control. In D. A. Linkens, editor, *CAD for Control Systems*, pages 127–146. Marcel Dekker, 1993.

[13] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering Series.* Academic Press, 1970.

[14] T. Bohlin. Computer-Aided Grey-Box Identification. Technical Report TRITA–REG–8403, Royal Institute of Technology, Stockholm, Sweden, 1984.

Author Index

MATLAB®
Die Software für math.-techn. Berechnungen

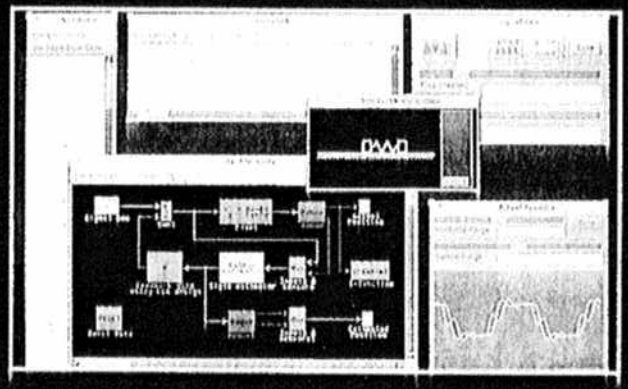


MATLAB® für Ingenieure und Naturwissenschaftler. Einfach anzuwenden. Ersetzt aufwendige Eigenprogrammierung.

Anwendungsgebiete:
- Gleichungsdefinition, Matrizenarithmetik
- Grafische Darstellung, 2D+3D
- Gleichungsbasierte Simulation nichtlinear. Systeme
- Auswertung von Versuchsdaten, Visualisierung, Animation, Algorithmen-Entwicklung
- Formelauswertung, Statistik
- Eigenwertrechnung, Polynomarithmetik

Eigenschaften:
- Interaktive Anwendung, einfache Syntax
- PCs, Workstations und Mainframes
- Eigene Funktionen mit Fortran und/oder C
- Speichern und Wiederverwend. benutzereig. Funktionen
- Lesen und Schreiben beliebiger Dateiformate
- MATLAB ab **DM 1.600,-**, TOOLBOXEN ab **DM 990,-**

SIMULINK™
In MATLAB integriertes Simulationssystem für nichtlineare dynamische Systeme

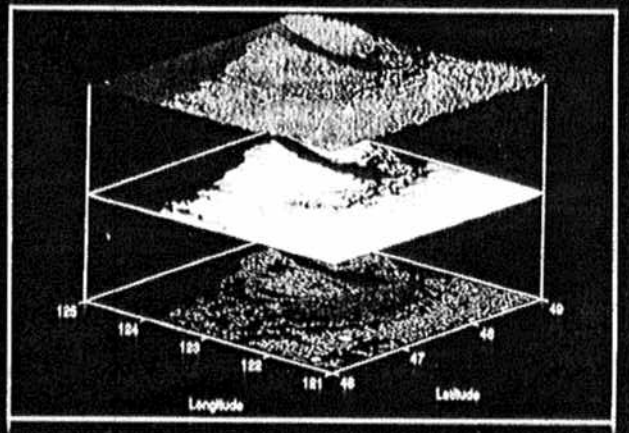


SIMULINK™ für die grafische blockbildbasierte Modellierung, Analyse und Simulation.

Modellierung:
- Lineare, nichtlineare, kontinuierliche und diskrete Modellteile in einem Modell
- Blockorientierte grafische Eingabe, aufbauend auf MS-Windows (PC), X/Motif (Unix-Workstation) oder Macintosh Windowing
- Teilmodelle, Zahl der Hierarchie-Ebenen praktisch unbegrenzt, viele Standardblöcke verfügbar
- Eigene Blöcke in MATLAB-, C- oder Fortran-Code
- Speicherung in lesbarem MATLAB-Code

Systemuntersuchung:
- Bestimmung des eingeschwungenen Zustands
- Linearisierung nichtlinearer Modelle
- Parameteroptimierung, Reglerentwurf, Signalanalyse mit MATLAB-Toolboxen
- Generierung von C-Quellcode: C Code Generator

TOOLBOXEN
Anwendungsspezifische Ergänzungen für MATLAB



TOOLBOXEN (TB) zur Ergänzung von MATLAB und SIMULINK mit leistungsfähigen, fachspezifischen Zusatzfunktionen.

Signalverarbeitung: Signal Processing TB

Regelungstechnik und Systemidentifikation: Control System TB, Nonlinear Control Design TB, Robust Control TB, µ-Analysis and Synthesis TB, System Identification TB, State Space Identification TB

Simulation mechanischer Systeme: MECHMACS (Ergänzung zu SIMULINK)

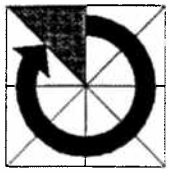
Meßdatenerfassung, -verarbeitung, Steuerung, Regelung in Echtzeit: ECHTZEIT-ERWEITERUNG (Ergänzung zu MATLAB und SIMULINK)

Universell einsetzbar: Optimization TB, Neural Network TB, Chemometrics TB, Spline TB, Statistics TB, Image Processing TB, Symbolic Math TB

Software mit Zukunft

scientific COMPUTERS

D-52064 Aachen, Franzstraße 107, Tel. 0241/260 41, Fax 0241/449 83
Geschäftsstelle München:
D-85744 Unterföhring, Firkenweg 7, Tel. 089/99 59 01-0, Fax 089/99 59 01-11



Unseld + Partner

Business- and Marketing Consulting Simulation!

What we can offer our customers !

Unseld + Partner :

- focuses on simulation topics only and provides efficient industrial simulation experience, gained in many industrial projects
- consists of a group of highly educated simulation experts representing the highest concentration of such know-how in Austria
- offers simulation expertise in industrial projects, while being receptive for sophisticated new strategies in freight centres and intercompany logistics, tackling especially multimodal aspects.
- provides advice on EU-programs (concerning railways, IT and Telematics) to leading members of Austrian industrial organisations and government bodies
- is a completely independent Austrian company
- has co-operation contracts and contacts with many prominent international research and university institutes

Of course state-of-the-art simulation software technology for instance SIMPLE++ and VISIO are used and professional project management skills are provided.

→ Without Simulation no Innovation ←

Unseld + Partner Business- and Marketing Consulting Simulation!
A-1080 Vienna Lerchenfelderstraße 44/9 phone +43-1-4030371-0* fax +43-1-4030371-90

Poster Book

F. Breitenecker
I. Husinsky
editors

ARGESIM Report No. 3

ISBN 3-901608-01-X

ISBN ebook 978-3-901608-01-8

DOI 10.11128/arep.1-2-3.ar3

ARGESIM Report No. 3
ISBN 3-901608-01-X
ISBN ebook 978-3-901608-01-8 DOI
10.11128/arep.1-2-3.ar3

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

Printed by *CA Druckerei*, Vienna, Austria

Foreword

Developments over the last years show that beside the classical tools theory and experiment simulation becomes more and more the third major tool for problem solving in application and research. Nowadays simulation is found in nearly every application area, research activities result in new methodologies and tools for simulation, and more and more simulation software, simulators, and simulation systems are offered on the market.
The *EUROSIM Congress*, the European Simulation Congress, an international event normally held every three years, aims to be a common forum for presenting European and international recent results and applications in simulation, and to stimulate the exchange of ideas and experiences among scientists and engineers active in simulation.
EUROSIM is the Federation of the European Simulation Societies, acting as a European forum for Simulation Societies and promoting the advancement of system simulation in industry, research, and education.

All these intentions are reflected in the 5th European Simulation Congress *EUROSIM 95*, the 2nd Congress after the formal foundation of EUROSIM. The scientific programme consists of invited and contributed papers to regular sessions and to "Special Interest Sessions", of contributions to the session "Software Tools and Products", and of posters.

The invited and contributed papers to regular sessions and to "Special Interest Sessions" are published in the Congress Proceedings printed by Elsevier Science B.V. and in a Late Paper Volume (ARGESIM Report no. 1, ISBN 3-901608-01-X). The Proceedings contain eight invited papers and 212 contributed papers, the Late Paper Volume contains 20 contributed papers. The papers were selected by the International Programme Committee from 459 abstracts received.

The session "Software Tools and Products" presents papers dealing with State-of-the-Art and new features of simulation languages, simulators, and simulation environments. These contributions also passed the review process and are published as ARGESIM Report no. 2 (Proceedings EUROSIM'95 - Session "Software Tools and Products", ISBN 3-901608-01-X).

The reviewed Poster Session completes the scientific programme. The abstracts of the 111 posters are published as ARGESIM Report no. 3 (EUROSIM'95 - Poster Book, ISBN 3-901608-01-X).

It is interesting to compare the titles of papers and posters presented at previous European Simulation Congresses with those at the present congress. Even a brief glance through the four volumes of Proceedings and Late Paper Volumes shows that in this twelve year period considerable, remarkable, and sometimes astonishing advances have been made in a number of different areas. For example, developments in parallelism and distributed processing are now not only being seen in simulation applications but are also frequently used.
Object-oriented methods are being implemented now, and artificial intelligence and knowledge-based tools appear to be an established part of system modelling and simulation methodology. The availability of improved graphic algorithms and tools is also leading to some very interesting and innovative research and application in terms of man-machine interface and of animation and visualisation, both for discrete-event and continuous-system simulation.

New developments in terms of mathematical modelling and simulation techniques as well as in terms of general methodology are of little significance unless they are stimulated by the requirements of the real world in terms of industry, business, agriculture and the sciences. We are very pleased, therefore, that application papers are so well represented. This also applies to papers on parallel and distributed simulation, where beside graphics the fastest development can be observed.

We are also pleased that the idea of "Special Interest Sessions" could be realized. These sessions deal with recent developments in areas where methodology and application are considered together. The results of the closing discussion at the end of these sessions are summarized in manuscripts which will be edited and published in abbreviated form in *EUROSIM - Simulation News Europe* (SNE), the newsletter of the EUROSIM member societies. Some of these papers will be prepared for publication in EUROSIM's scientific journal *SIMULATION PRACTICE AND THEORY*. A separate role is played by the Industry Session on "Model Exchange and Software Independent Modeling" where people mainly from industry report on this topic without necessarily having to publish a paper in the Proceedings. Furthermore, we are pleased, that the contributions to the session "Software Tools and Products" show a very broad spectrum of simulation software, and that the Poster Session presents new ideas under development.

The European Simulation Congress *EUROSIM 95*, held in Vienna (Austria) at the Technical University of Vienna from September 11 through September 15, 1995, is organized by *ASIM* (Arbeitsgemeinschaft Simulation), the German speaking Simulation Society, in co-operation with the other member societies of EUROSIM: *AES* (Asociación Española de Simulación), *CSSS* (Czech & Slovak Simulation Society), *DBSS* (Dutch Benelux Simulation Society), *FRANCOSIM* (Société Francophone de Simulation), *HSTAG* (Hungarian Simulation Tools and Application Group), *ISCS* (Italian Society for Computer Simulation), *SIMS* (Simulation Society of Scandinavia), *SLOSIM* (Slovenian Society for Simulation and Modelling), *UKSS* (United Kingdom Simulation Society).

The moral co-sponsorship of *CASS* (Chinese Association for System Simulation), CROSSIM (Croatian Society for Simulation Modelling), *IFAC* Advisory Board Austria, *IMACS* (International Association for Mathematics and Computers in Simulation), *JSST* (Japanese Society for Simulation Technology), *LSS* (Latvian Simulation Society), *OCG* (Austrian Computer Society), *PSCS* (Polish Society for Computer Simulation), *ROMSIM* (Romanian Society for Modelling and Simulation), *SCS* (Society for Computer Simulation), *SiE* Esprit Working Group "Simulation in Europe" supports this congress.

A successful conference is always due to the efforts of the many people involved. To this purpose, particular acknowledgement goes to the members of the Scientific Committee for their contributions in the paper selection process, to the members of the Local Organizing Committee, and more especially to the head of this committee, to Mr. Manfred Salzmann. We would like to thank *Unseld + Partner* and *CA (Creditanstalt)* for sponsoring the printing of this report.

Felix Breitenecker Irmgard Husinsky

Technical University of Vienna

Scientific Committee

F. Breitenecker (Austria), Chairman

H. Adelsberger (D)
M. Alexik (SQ)
W. Ameling (D)
S. Balsamo (I)
I. Bausch-Gall (D)
S.W. Brok (NL)
F.E. Cellier (USA)
V. Ceric (HR)
L. Dekker (NL)
J. Dongarra (USA)
V. De Nitto (I)
H. Ecker (A)
G. Feichtinger (A)
P. Fishwick (USA)
J.M. Giron-Sierra (E)
H.J. Halin (CH)
N. Houbak (DK)
R. Huntsinger (USA)
I. Husinsky (A)
T. Iversen (N)
A. Jávor (H)
K. Juslin (FIN)
G. Kampe (D)
E. Kerckhoffs (NL)
W. Kleinert (A)
P. Kopacek (A)
W. Kreutzer (NZ)
M. Lebrun (F)
F. Lorenz (B)
F. Maceri (I)
D. Maclay (GB)
H. Mang (A)
Y. Merkuryev (LV)
Z. Minglian (VRC)
I. Molnar (H)
D.P.F. Möller (D)
D. Murray-Smith (GB)
F.J. Pasveer (NL)
R. Pooley (U.K.)
W. Purgathofer (A)
P. Schäfer (D)
T. Schriber (USA)
A. Seila (USA)
W. Smit (NL)
F. Stanciulescu (RO)
A. Sydow (D)
H. Szczerbicka (D)
S. Takaba (J)
I. Troch (A)
G.C. Vansteenkiste (B)
W. Weisz (A)
J. Wilkinson (GB)
R. Zobel (GB)
B. Zupancic (SLO)

Organization Committee

Felix Breitenecker, Dept. Simulation Techniques,
Institute for Technical Mathematics, Technical University of Vienna

Irmgard Husinsky, Dept. for High Performance Computing, Computing Services,
Technical University of Vienna

Local Organizers are the Dept. Simulation Techniques of the Technical University of Vienna and the Dept. for High Performance Computing of the Computing Services of the Technical University of Vienna, and the "ARGE Simulation News".

Local Organizing Committee

M. Salzmann;

K. Breitenecker, B. Gabler, M. Holzinger, C. Kiss, M. Klug, N. Kraus, M. Lingl,
I. Mannsberger, J. Schuch, G. Schuster, H. Strauß, S. Wassertheurer, W. Zeller

About ARGESIM

ARGE Simulation News (ARGESIM) is a non-profit working group providing the infra structure for the administration of **EUROSIM** activities and other activities in the area of modelling and simulation.

ARGESIM organizes and provides the infra structure for

- the production of the journal EUROSIM Simulation News Europe
- the comparison of simulation software (EUROSIM Comparisons)
- the organisation of seminars and courses on modelling and simulation
- COMETT Courses on Simulation
- "Seminare über Modellbildung und Simulation"
- development of simulation software, for instance: mosis - continuous parallel simulation, D_SIM - discrete simulation with Petri Nets, GOMA - optimization in ACSL
- running a WWW - server on EUROSIM activities and on activities of member societies of EUROSIM
- running a FTP-Server with software demos, for instance
  - * demos of continuous simulation software
  - * demos of discrete simulation software
  - * demos of engineering software tools
  - * full versions of tools developed within ARGESIM

At present ARGESIM consists mainly of staff members of the Dept. Simulation Technique and of the Computing Services of the Technical University Vienna.

In 1995 ARGESIM became also a publisher and started the series **ARGESIM Reports**. These reports will publish short monographs on new developments in modelling and simulation, course material for COMETT courses and other simulation courses, Proceedings for simulation conferences, summaries of the EUROSIM comparisons, etc.

Up to now the following reports have been published:

No.	Title	Authors / Editors	ISBN
# 1	Congress EUROSIM'95 - Late Paper Volume	F. Breitenecker, I. Husinsky	3-901608-01-X
# 2	Congress EUROSIM'95 - Session Software Products and Tools	F. Breitenecker, I. Husinsky	3-901608-01-X
# 3	EUROSIM'95 - Poster Book	F. Breitenecker, I. Husinsky	3-901608-01-X
# 4	Seminar Modellbildung und Simulation - Simulation in der Didaktik	F. Breitenecker, I. Husinsky, M. Salzmann	3-901608-04-4
# 5	Seminar Modellbildung und Simulation - COMETT - Course "Fuzzy Logic"	D. Murray-Smith, D.P.F. Möller, F. Breitenecker	3-901608-04-4
# 6	Seminar Modellbildung und Simulation -COMETT - Course "Object-Oriented Discrete Simulation"	N. Kraus, F. Breitenecker	3-901608-04-4
# 7	EUROSIM Comparison 1 - Solutions and Results	F. Breitenecker, I. Husinsky	3-901608-07-9
# 8	EUROSIM Comparison 2 - Solutions and Results	F. Breitenecker, I. Husinsky	3-901608-07-9

For information contact: ARGESIM, c/o Dept. Simulation Techniques,
attn. F. Breitenecker, Technical University Vienna
Wiedner Hauptstraße 8-10, A - 1040 Vienna
Tel. +43-1-58801-5374, -5386, -5484, Fax: +43-1-5874211
Email: argesim@simserv.tuwien.ac.at

Table of Contents

Discrete Simulation

Parallel Simulation

Software Tools

Education

Methodology

PROCESSING SPEECH WITH NEURAL NETWORK

M.M. Al-Akaidi
Dept. of Electronic Engineering,
De Montfort University, Leicester,
P.O.Box 143, The Gateway,
LE1 9BH, UK.
e-mail: mma@dmu.ac.uk

EXTENDED ABSTRACT:

Adaptive quantisers with and without some form of adaptive prediction are becoming increasingly common in speech transimission systems where high quality speech is required with a reduced bit-rate. Such memory-based coders contrast with simpler memory-less quantisers whose action is dependent purely on the signal amplitude at a single instant. The characterisation and evaluation of such memory-based coders with a wide variety of input signals presents difficulties due to the dependence on the content of the coder's memory.

The state of any system with memory can be described as a vector in a suitable n-dimensional state space. As the state of the system and the corresponding vector changes, the system can be said to follow a trajectory around the state space. The form of this trajectory may be very simple. For example, if the system has no input, the coder may not change state at all and the trajectory will be a single point. If the coder exhibits small amplitude limit cycle behaviour, the trajectory will make a circuit through a small number of states.

These examples serve to show that the trajectory characterises the speech coder's performance in terms of its internal state. This behaviour is modified by the input signal. It is also modified by any error performance of the coder: e.g. slope overloading effects. The trajectory of the coder with a large amplitude signal, exhibiting some form of limiting behaviour forms an extremely complex figure.
Such figures can be obtained by simulation of coder algorithms, but their interpretation is less than straightforward.

The system is further complicated by the existance of noise on the input signal, or dither signals applied to the adaptive quantiser. The presence of such random purturbations on the signal are reflected in the state space trajectories; making their character less clear. The presence of such signals in coder algorithm simulation is desirable as results are more likely to reflect real applications.

A single memory adaptive quantiser executes a trajectory in a one dimensional state space. When combined with a single tap adaptive predictor, the coder exhibits complex behaviour in a two dimensional space. The figures traced by the states of the coder in response to a given input signal fully describe the internal operation of the coder at any instant. Noise present on the input signal (or dither at the quantiser) creates difficulties in the automated interpretation of the state space trajectories.

The paper describes the use of a neural network to identify aspects of speech coder simulation behaviour where the results are in the form of state space trajectories. The distinction between different modes of behaviour (limit cycle, overloading signal etc) is demonstrated.

EUROSIM CONGRESS '95

TU VIENNA, September 11-15

FUZZY-Control Simulation of Heat Supply by Circular Heat Networks

Balátě,J., Druckmüller,M., Starý,A., Sysala,T.

A Fuzzy-control application was examined for the heat distribution control by circular heat networks. In the large cities with **the centralized heat supply**, the heat is distributed to the consumers through the circular networks, that are suplied by several cooperating heat sources, that means by power and heating plants and heating plants.

The heat supply control lies in the optimal production control of the heat sources (power and heating plants and heating plants) and in the heat distribution control with the aim, to keep required values of hydraulic parameters on the given circular heat network configuration. In this time, this task is carried out by the dispatcher of the heat distribution (main dispatcher in the control point of the heat distribution), according to his knowledge, experince and capability, on the base of the information about hydraulic parameters of the controlled heat network. The dispatcher decides, what heat power output and from which cooperating heat sources it is necessary to supply during the specified day. His decision is used as a message for dispatchers in the cooperating power and heating plants and heating plants. These dispatchers fulfil the instructions and commands of the heat disribution dispatcher and they make their own decisions themselves, how the heat power requirements for their production plants will be provided.

The modern way of control, the Fuzzy-control theory, is offered for the operational control of the circular heat networks in the large systems.

In our case the Fuzzy-control application was examined on the example of miniaturized model of a heat supply network, by using of DEMO-version, as well as for a more extensive model of centralized heat supply system by using of full version of **expert system LMPS - Linguistic Model Processing System**. LMPS-system was produced on the Technical University in Brno, Faculty of Machinary Engineering.

Prof. Ing. Jaroslav Balátě, DrSc.

Ing. Tomáš Sysala

Department of Automation and Control Technology at Faculty of Technology in Zlín,

nám. TGM 275,
762 72 Zlín, CZ
Tel./Fax: + 42 67 33522

RNDr. Miloslav Druckmüller, CSc.
Mathematical Institute at Faculty of Machinary Engineering,
Technical Univerzity in Brno, CZ

Ing. Antonín Starý
Power and Heating Plant, Inc., Brno, CZ

MODELING OF CONTROLLED MOTIONS OF HUMAN LOWER EXTREMITY VIA OPTIMAL PROGRAMMING

Victor BERBYUK

Pidstryhach Institute for Applied Problems of Mechanics and Mathematics of the Ukrainian National Academy of Sciences
No.3-B, Naukova St., Lviv, 290601, Ukraine
Tel.: 380 322 636000, Fax.: 380 322 654240
E-mail: Kalyniak%IPPMM.Lviv.UA@LITech.Lviv.UA

Modelling of biodynamical system, especially human locomotor system or some of its part is very complicated and challenging problem. This problem is interesting and important for different applications in mechatronics, ergonomics, medicine, biomechanics of sport, rehabilitation technologies, etc.

In our report the modeling problem of the controlled motions of human lower extremity via optimal programming is considered.

It's necessary to model of the energy - optimal controlled motion of human leg in swing phase from the initial state to the final state over the given time. The modeling problem of human leg motion is formulated as the energy - optimal control problem for 8-th order nonlinear dynamical system with two point boundary conditions, given time of the control process and some constraints on phase coordinates and controlling forces. These constraints are based upon experimental data of the human locomotion.

From mathematical point of view the optimal control problem in question is very complicated problem with undifferentiable functional and nonlinear restrictions on phase coordinates.

The efficient approach for designing and simulation of the energy-optimal controlled processes of the human leg during swing phase have been proposed.

By means of the Fourier and spline approximations the considered continual optimal control problem have been reduced to some nonlinear programming problem which is solved by a standard algorithm of the constraints optimization.

A number of the energy-optimal controlled processes of human leg for different boundary conditions and restrictions on phase coordinates and controlling forces have been obtained.

The kinematic dynamic and energetic characteristics of the optimal control processes obtained are compared with respective characteristics of human leg swing phase during normal locomotion.

A MODEL: VEGETATION SHIFTS AND CARBON POOLS DYNAMICS UNDER GLOBAL CLIMATE CHANGE

B.G. Bogatyrev, N.V. Belotelov,

Center for Ecology and Forest Productivity Russian Acad.Sci.

ABSTRACT

Nowadays, modeling of biospheric response on expected climate change due to anthropogenic influence is intensively developing scientific problem. Within the task, there are a few model approaches for prognosis of long-term global vegetation dynamics. The principles of them have been discussed.

Model approach based on bioclimatic schemes has been chosen for further considerations. The schemes of global bioclimatic classification connect in some manner the climatic variables (such as temperature, precipitation, etc.) and characteristics of vegetation. The model approach could forecast potential vegetation pattern redistribution under climate warming. However such model does not permit to investigate the temporal characteristics of vegetation change.

An approach of time delays has been created to incorporate vegetation dynamics into the modeling (Belotelov et al., 1994). A model, designed within this approach, is based on Holdridge's bioclimatic scheme and completed by a procedure of time delays. The procedure was designed to take into account inertia of vegetation response on climate change. It was based on expert estimates of biome replacements.

Next step was to combine the suggested model and simple submodel of carbon cycle. Dynamics of two basic terrestrial pools (living phytomass and soil) were considered. The fluxes (production, dying off, destruction) are determined by external climatic parameters and by current biome type, defining its coefficients' values. The submodel structure is similar to the typical balance models (e.g. Esser 1984, Moiseev et al.1985). The cardinal difference is change of values of the flux coefficients when original biome is replaced.

A time-dependent climatic scenario for Russia has been designed. It is presented by various power functions of time variable, connecting fixed initial and final states of vector of climatic parameters. We assume that the climate reaches its final state after 100 years. The initial climate was extracted from IIASA database. The final climate corresponds to the "hybrid" scenario for doubled CO_2 in the atmosphere. Three scenarios of general circulation models and three paleo-analogy scenarios have been used for the hybrid scenario.

The designed model was analyzed to examine number of stable states and their stability.

A number of computer experiments has been conducted to estimate vegetation shifts and correspondent dynamics of terrestrial carbon pools. One of the important results was a compensation effect of the vegetation redistribution: the pattern shifts increase absorbing of the atmospheric carbon by terrestrial cover. Thus, long-term accumulation of the carbon by vegetation is bigger in the case of the pattern shifts than in the case of the fixed original state. It is interesting that even experiment, when forest biomes cannot migrate and replace the other biomes, also demonstrates compensation effect. The result shows the important role of non-forest vegetation in absorption of the additional atmospheric CO_2.

The model would be the basis for further development of suggested approach, including coupled description of processes at different hierarchical levels in vegetation.

Keywords: bioclimatic scheme; model; vegetation shifts; carbon pools; time-dependent response.

Student perception in the use of computational tools for building performance analysis

P E Booth & P E Oakey
Centre for the Built Environment
Leeds Metropolitan University.
Brunswick Terrace, Leeds LS2 8BU

Abstract

University teachers must take responsibility for ***what*** and ***how*** their students learn and be aware that '***concentrating hard is not necessarily understanding***'.Teachers should create the conditions in which understanding is possible- the student's responsibility is to take advantage of that situation.To facilitate these aims many different teaching and learning techniques are used nowadays and 'educational technology' is heavily funded, particularly in the Higher Education sector. Unfortunately, vast sums are often made available to facilitate the 'method' but the subject matter taught and student perception in its use are incidental aspects. Teaching strategies are undergoing radical re-appraisal and the focus is on 'presenting the learner with new ways of seeing' and the creation of a different interaction between student and teacher.

The study of building engineering services and environmental science on undergraduate construction management courses requires students to grapple with a wide range of peripheral aspects, particularly in building physics. There is now a plethora of computer software available for analysing the physical processes which affect the performance of occupied buildings and much of this utilises very sophisticated mathematical modelling techniques to examine heat transfer and fluid flow processes.

This paper describes the design of a project currently underway at Leeds Metropolitan University (LMU) which introduces second year undergraduate students, with minimal computer knowledge but having a reasonable building science background, to the use of a sophisticated, commercially -available thermal modelling package (APACHE) for analysing the energy consumption and internal environmental parameters of a building subject to real weather influences (albeit historical). The effects of, for example, architectural changes, material specification modifications and operational parameters can be simulated singly or in combination. To facilitate the use of the software a study guide has been written students being provided with individual copies, the software being accessed via twenty terminals on a Local Area Network (LAN).

There is no attempt to expose students to the mathematical concepts and techniques inherent in the design and operation of the software. The aim is to allow the student rapid access to the power of an excellent analytical tool in order to develop critical and investigative abilities, to stimulate curiosity and to exercise engineering judgement. Initially the student uses written scenarios, "self -building" the model using a step-by -step guide based on the software author's manual but prepared in a more user-friendly format. Model answers and discussion are provided and these lead the student to self-assessment questions about alternative scenarios.The detailed study guide is commercially available from LMU.

Student perceptions have been monitored in tutorial sessions with encouraging results, and overall opinion is that this approach is a genuinely useful alternative to traditional lectures. Feedback from students has suggested that some revisions/improvements are nontheless appropiate and this is to be addressed as the project evolves. Generally there was a marked improvement in understanding as the course progressed. and it is considered worth extending the approach for use with a computational fluid dynamics software package.

Modeling and Optimization of Carbon and Nitrogen Elimination

A. Vogelpohl[a], M.Sievers[a], D.P.F. Möller[b], B.R. Bracio[b], J. Jungblut[b]

[a]Clausthal Environmental Technology Institute (CUTEC),
Leibnizstr.21-23, 38678 Clausthal-Zellerfeld, Germany

[b]Institute for Computer Science, Technical University of Clausthal,
Erzstr.1, 38678 Clausthal-Zellerfeld, Germany

On the background of risen costs for nutrient removal from wastewater a new treatment process for carbon and nitrogen elimination has been developed. Using this process the extension or reconstruction of wastewater treatment plants becomes cheaper. The intention of this paper is to present a simulation model of the new process taking into account the activated sludge model (ASM) No.2 from the IAWQ. It will be shown that it is possible to optimize the complex process of carbon and nitrogen elimination.

The process consists of a three-step treatment and corresponds to a post denitrification with respect to the sequence of the unit operations. The first step – a carbon elimination – works as an activated sludge process. Here, an essential property is the carefully directed adsorption of organic carbon compounds by the activated sludge. The adsorption results in accumulation and storage in the bacteria and can be influenced by the aeration rate.

The second treatment step is a nitrification in a packed-bed reactor followed by a denitrification in a stirred tank reactor which represents the third step. The important characteristic of the whole three-step process is a bypass stream of recycle sludge from the first step to the denitrification tank. The adsorbed carbon in this stream is used as a substrate for denitrification so there is no need for an external carbon source. In addition, there exists a recirculation of recycle sludge from the third step to the oxidation/adsorption tank. This exchange of sludge between the first and the third step causes the complexity of the process since nitrification is relatively easy to handle with respect to process engineering.

Besides the regulation of aeration in the oxidation/adsorption step the bypass stream has to be controlled carefully. The main criterion is a minimization of the stream in order to minimize ammonia and COD (chemical oxygen demand) in the effluent. On the other hand sufficient carbon for denitrification has to be provided.

The whole three-step process has been tested in a pilot plant which is equipped extensively with measuring instruments. Observations of the dynamic behaviour of the process were used to develop a simulation model with special regard to adsorption, accumulation and storage of organic compounds in the sludge. This work has been done taking into account ASM No.2 and describing in addition the adsorption and its influence on the denitrification rate.

Based on measurement data from the pilot plant verification and validation of the model have been performed. In particular an accurate determination of the kinetic parameters was necessary. This way a reasonable agreement between predicted and measured data has been obtained. Therefore it is possible to use the model for process optimization and control.

Simulating Multimedia Communication Systems

Poster Summary

Mr. Peter Duffy
Dept. of Engineering
Glasgow Caledonian University
Cowcaddens Road
Glasgow G4 0BA
Telephone: +44 141 331 3505
email: pdu@gcal.ac.uk

Multimedia and the Internet have been the largest growth areas in the computing arena of the 1990s. The introduction of the WorldWide Web (WWW) saw the first major convergence of the two areas, and web traffic now accounts for some 30% of all Internet traffic, up from around 1% at its introduction. If the current rate of growth continues, multimedia traffic on the Internet will exceed worldwide telephony traffic before the end of 1996 [1].

While the web, with its friendly graphical front end and ability to access a wide variety of information, is a tremendous boon to end users, it places a great strain on networks. The problem which the web poses is twofold:

- firstly it makes it much easier for users to access information which has always been available;
- secondly it makes it easier to access newer multimedia information such as still images, video and audio and even to play these in real-time over the network.

Both of these effects result in a huge increase in the load placed on the network, with a resulting decrease in quality of service. The ability to quantify this quality of service is necessary for network operators to plan network expansion. Unfortunately, theoretical models for networks dealing with multimedia traffic are less mature than those for conventional packet-switched data networks. In addition, few commercially available network simulation packages have the ability to deal with multimedia traffic and new multimedia protocols built in.

This poster examines the possibility of using relatively low-end (and hence inexpensive) simulation packages to model two multimedia communication systems. We first develop traffic models based on some short MPEG clips and published research. We then compare mechanisms for generating traffic according to these models using the relatively simple statistical distribution functions built in to many simulation packages. Finally we use these traffic models to drive simulations of two communication systems: packet video on an Ethernet LAN and multimedia traffic on a newly proposed ATM switch architecture.

References

[1] *The Net Effect*, Roger Dettmer, IEE Review March 95, pp 67-71

Modelling and Simulation of Power Plants for Stability Studies

Michael Fette, Jürgen Voss, Torsten Wieborg

University of Paderborn
Department of Electrical Engineering
Pohlweg 47 – 49
33098 Paderborn, Germany
email: fette@eevmic.uni–paderborn.de

Key words: modelling, simulation, nonlinear analysis, power plants

With respect to process dynamics, boiler–turbine systems are very complex continuous processes, because of different concurrent reasons: structural complexity, strong nonlinearity, interaction among subsystems, relevance of distributed parameter phenomena, considerable uncertainty of process parameters at macroscopic knowledge level (heat transfer constants, friction, flame radiation, two–phase interaction etc.).

In view of their great technical and economical importance in the power industry, the dynamics of boiler–turbine processes has been the object of considerable research effort. Most of the works come from the process control community: for that reason, simple global boiler models suitable for control design were essentially proposed though some papers more widely looked at application of numerical techniques. However, global models are rather convenient to understand the gross behavior of an existing plant than reliable to predict the dynamics of a new plant during the design phase, since such models, incorporating drastic simplifying assumptions, generally need ad hoc tuning.

Analyzing the stability of the first swing of electrical power systems, the power plant can be modelled very simply by an exosystem only, which describes the ability to outstore the stored energy within the plant to stabilize the global system. For long term stability studies, the dynamics of the plants must be modelled more detailed. Compared to the models e.g. boiler manufactures use, they are simple ones with low order. These models are essential to study multimachine power system frequency dynamics, in which case very crude assumptions are made about the boiler while keeping a better representation of turbine and speed governor.

Fossil fired power stations with fixed pressure and modified sliding pressure working conditions are considered essentially. Well known power plant models suitable for dynamic studies of electrical power systems were used, to describe the necessary dynamical behavior from the view of the electrical net. The proposed model is mathematically in such a shape, that the nonlinearities of the plant are approximated by nonlinear functions with analytical properties, were the physical structure of the plant is preserved. Therefore, this model can be used in the future for analytical nonlinear controller design of more sophisticated frequency–power–controllers. Also, due to the smoothness of the nonlinear approximations, it has good numerical properties in simulation studies. The model is incorporated into a matched simulation structure based on MATLAB/SIMULINK and is conceived to be connected with other models of power system elements with respect to boundary conditions.

Further, the derived simulation model can be used as a stand–alone model and is designed to use it for teaching of power plant dynamics with an easy to use control mask for parameters an inputs.

Modelling of falling slurry film absorber applied to flue gas desulphurization process in combination with heat recovery

M. Gerbec, A. Stergaršek, D. Ban, R. Kocjančič

Department for inorganic chemistry and technology, Jožef Stefan Institute, Jamova 39, 61000 Ljubljana, Slovenia

ABSTRACT

Non stationary model of wet limestone desulphurization process for coal fired power stations was developed to study chemical and physical phenomena during operation. Model of SO_2 absorption from flue gas to counter current flow of falling slurry film inside vertical tube(s) was developed. Properties of falling film inside vertical tube were calculated according to literature data about laminar, wavy and turbulent film flow regime:

- From liquid entrance (at the top) to developed waves area, the properties of laminar film area were calculated
- In main area developed film waves were modelled by additional division. Single wave was divided to wave's front, back, trail and substrate. Each mentioned part of wave has specific hydrodynamic properties.
- At film flow regimes with Reynold's number greater than 2000, combined wavy and turbulent film flow was modelled.

Mass transfer of SO_2 across gas/liquid phase boundary was calculated taking into account mass transfer resistance on gas and liquid side, Henry's constant of SO_2, and absorption enhancement because of further chemical reaction in liquid phase. In addition to mentioned model of falling film and mass transfer also the following chemical reactions were modelled:

- Chemical reactions of absorbed SO_2 including chemical equilibria of sulphates (IV), sulphates (VI) and carbonates. Based on electrical balance of all charged ions, new equilibrium pH value in any modelled liquid element was calculated.
- Oxidation of sulphate (VI) ions to sulphate (VI) ions by forced air introduction to slurry conditioning tank(s).
- Polidisperse dissolution of limestone particles in slurry at given chemical conditions.
- Precipitation of gypsum ($CaSO_4 \cdot 2H_2O$)
- Partial precipitation and dissolution of calcium sulphite ($CaSO_3 \cdot 2H_2O$)
- Stripping of CO_2 gas liberated from limestone dissolution.

Also heat transfer from liquid film across tube wall to surrounding cooling media was calculated. It was found that all resistance is located on the side of surrounding flow of media. Simultaneous absorption and heat transfer have been verified in laboratory apparatus consisting of concentric tube type absorber heat exchanger. Good agreement between model and experimental data regarding SO_2 removal rate at various operating conditions was found.

Model divides falling film absorber to vertical slices of liquid and gas, which are in contact in appropriate iteration time. In that way, absorber profiles are calculated. Also chemical composition of slurry conditioning tank(s) contents are calculated. Model was implemented as computer program for IBM PC compatible computers.

Results of modelling show that low temperature heat can be recovered from flue gases in absorber/heat exchanger. Depending on flowsheet it has been estimated that 2.3 % of heat produced in the boiler can be reused for boiler air and condensate preheating and for the flue gas re-heating. Part of high-heat value can be recovered by flue gas cooling under dew point.

SIMULATION OF VISUAL CROSS-TRACKING

Nassimizadeh, H. Gough, N.E. and Musgrove, P.B.

School of Computing and Information Technology, University of Wolverhampton, UK

The most complex problem which can be considered in the theory of correlation and data association is that of multi-target tracking. .Ambiguities arise when targets are closely spaced as in the case of two crossing targets. When two target are crossing each other, it may be very difficult to determine which measurement belongs to which target. In this situation, miscorrelation could occur.

Although developments have been made on the crossing target problem, it remains an area which merits further consideration. The aim of this work is to introduce a new algorithm to deal with the crossing tracks in a multi-target environment. However, to establish some ideas for dealing with the problem and to test the suitability of different methods, the simpler problem of two crossing tracks is first examined.

In the processing procedure, the order of the recorded data is not important. Here, two aircraft with random acceleration perturbing their motions from a straight line are considered. The position of the targets are measured at uniformly spaced intervals T seconds apart, and each observation is noisy. The accuracy of position and velocity estimates depends not only upon the sensor accuracy, but also upon the perturbing acceleration and upon the sensor geometry. The problem is to determine the accuracy with which the aircrafts' trajectories can be estimated.

Three approaches have been investigated for dealing with the problem of two crossing targets. The first approach makes use of a limited batch processing technique to create all possible tracks based on four scans data and then decide on the most likely pair. Two criteria have been used for pruning, one terminating tracks which change slope too rapidly and a second tracks with higher total variances.

Here, the underlying assumptions of the least squares method were applied. The errors were computer generated with Gaussian distribution, zero mean, and constant variance. Then, these errors were added to the ideal data points and the simulation tasks were performed by means of 100 runs in each step. The simulations were carried out for the following three cases:

1 Situation in which the errors associated with y-values were dominant. Here, a second degree polynomial fit was applied.

2 Case where both x and y-values were subject to errors of about he same order of magnitude. A straight line was chosen to fit the data points. Here, the minimisation of the sum of the squares of the perpendicular distances from the data points to the line was considered as the minimisation function.

3 The same as case two except the data points were fitted by a circle.

Our aim in this paper is to make a comparison between the results obtained here with those of our previous approaches to come to a better understanding of the problem and therefore a better solution.

Address for correspondence:
Professor Norman E. Gough
School of Computing and Information Technology
University of Wolverhampton, Wulfruna Street,
Wolverhampton WV1 1SB
Fax: +1902 322680 E-mail cm1822@wlv.ac.uk

SIMULATION OF SHAPE CHANGE OF BIOLOGICAL CELLS DUE TO EXTERNAL HIGH FREQUENCY ELECTRIC FIELDS IN THREE DIMENSIONS.

Reinhard Grebe, Guido Bartsch and Martin Baumann
Dept. of Biomedical System Analysis, Institute of Physiology, RWTH-Klinikum, Pauwelsstr.30, 52057 Aachen, Germany. Email: martin@physiology.rwth-aachen.de

Introduction High-frequency homogenous and inhomogenous electric fields can be used to apply defined anisotropic forces on biological cells or parts of them. Mechanical properties of the cells can be calculated, if the resulting changes in cell geometry are quantified. Typical properties of interest are shear modulus, global elasticity modulus and others. The shape quantification can be done by light microscopy and image analysis. Most of the investigated cells do not show any kind of symmetry. In addition it turned out that it is extremely useful to use inhomogenous electric fields in these experiments. So, we can not compute *analytical* solutions, but use a finite-element method for *numerical* solutions of the three-dimensional field problems instead.

Methods and Materials Fig. 1 shows the principle design of our software package. The first step is to construct a discrete model using the FAM (Fegs, Cambridge, UK) preprocessor, which allows an efficient tetraeder meshing. Fig. 2 gives, as an example, the model of a spherical thin shell in a homogenous electric field. The first of three interfaces - *pre2solv* -

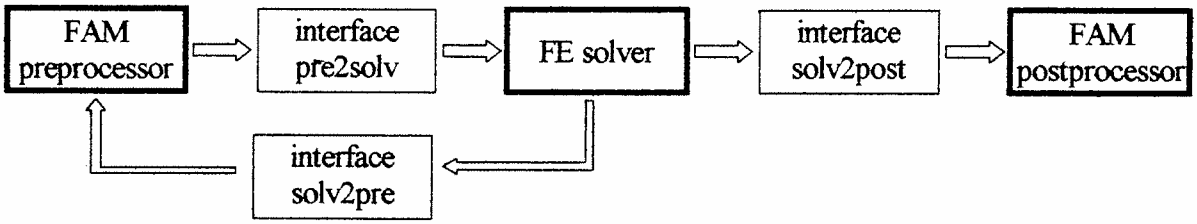


Fig 1: Dataflow during construction, meshing, solving, visualization and remeshing.

takes over the FAM ASCII data output containing the geometrical data and the electrical and bounding information and converts it into a matrix. This matrix is solved by our *FE solver*, a C++ program. This program runs under PVM, which makes it easy portable. The solver itself produces a file containing data needed to calculate forces and corresponding cell shape changes. The complete data set is redirected to the preprocessor for remeshing via the second interface - *solv2pre* - after the new geometry is computed. Since cell geometry and electric field lines and thereby forces depend on each other the loop described has to be reiterated as long as there are significant changes in any of the magnitudes. As soon as an equilibrium state has been reached, the iteration is stopped and the solver output is converted in the third interface - *solv2post* - to a FAM postprocessor readable file. This postprocessor performs the visualization of the results. Fig. 3 gives an example, where the calculated equipotential lines of the above mentioned shells are drawn.

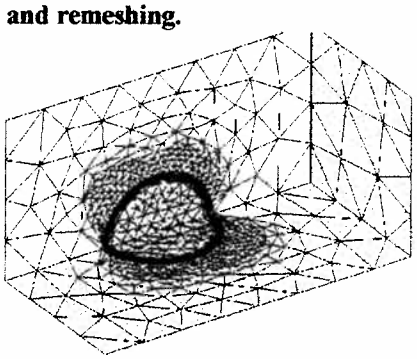


Fig. 2: meshing sample

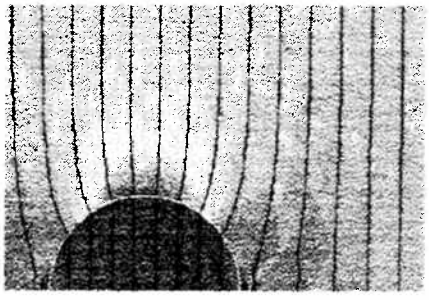


Fig. 3:equipotentlial lines

THE RED BLOOD CELL SIMULATED AS A CLOSED TANK-TREADING FLUID MEMBRANE UNDER NORMAL FLOW CONDITIONS IN TWO DIMENSIONS

Reinhard Grebe and Frank Kreuder
Dept. of Biomedical System Analysis, Institute of Physiology
RWTH-Klinikum, Pauwelsstr.30, 52057 Aachen, Germany

Introduction Nearly half the blood volume in normal healthy human beings consists of cells, mostly red blood cells (RBC). Blood behaves like a suspension of stiff particles in big arteries where mainly small shear forces are to be found. In small capillaries where high shear stress acts on each single RBC, blood is more similar to an emulsion of droplets. Even a concentrate of up to 90% blood cells is able to flow out of a reaction vessel. In contrast densely packed spheres can to the utmost hold 74% of a volume and will show properties of a solid body. The reason for the described blood dimorphism may be due to the single RBC's ability to undergo a complex cell and membrane movement called 'tanktreading'. Until now tanktreading has only be observed in vitro under the influence of high shear stress [1]. To get a better understanding of the described RBC properties we start here with a two-dimensional simulation to investigate single cell behavior under shear stress in a hydrodynamic field.

Foundations To simulate a RBC in a hydrodynamic field we model

- the medium as a fluid passing a volume bordered by fix boundaries driven by an arbitrary pressure difference,
- the interior of the cell as a fluid of constant volume and distinct viscosity and
- the cell membrane as a two-dimensional fluid with constant area of surface.

To compute the properties and time dependent changes of each of these fluids we use a set of equations including

- equations for to establish the needed continuities especially of volume,
- specially adapted Navier-Stokes equations and
- boundary conditions for the kinematics and forces of the different fluid regions.

Methods We use a combined method to describe the discretized mesh-elements of the different fluids - *Lagrange* for the external medium with unchanging boundaries and *Euler* for the variable geometry of cell interior and membrane. According to the so-called MAC-scheme [2] there is a shift of half the length of a grid element between the mesh of pressures and of the two vector components of velocity respectively. Furthermore we increase the resolution for the elements of the moving boundaries by 100. The program is written in C and runs under PVM.

Tests To test the algorithms and their implementations we solved some standard problems well documented in literature. We simulated the stationary flow between two plates as establishing after a step functioned rise of pressure. More complex has been the 'driven-cavity' problem where systems of timedependent vortices appear and disappear in a cavity driven by a peripheral flow [3].

Simulations To simulate the RBCs behavior under shear stress, we introduced RBC-models of different viscosity in driven-cavities. Cells with an internal viscosity 10 times higher than the surrounding medium showed the behaviour of a solid, while cells with normal viscosity showed the 'tanktreading' called movement of the membrane around the cell interior. The results have been visualized in a computer animation.

Problems To avoid instabilities during a simulation run, the single time step t should be chosen according the formula $\Delta t_{max} \leq (\Delta x^2) / (4\ \nu)$. A model on the realistic microscopic scale would therefore need in our present implementation a CPU-time of 500 years for 1s of simulation. That is why we have so far performed our simulation only in the macroscopic regime.

[1] Fischer,T.M., Stöhr-Liesen,M. and Schmid-Schönbein,H.: The red cell as a fluid droplet: tank tread-like motion of the human erythrocyte membrane in shear flow. *Science* 202, 894-896, 1978.

[2] Harlow,F.H. and Welch,J.E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 8, 2182-2189, 1965.

[3] Soh,W.Y. and Goodrich,J.W.: Unsteady solution of incompressible Navier-Stokes Equations. *J.Comput.Phys.* 79, 113-134, 1988.

AUTOMATED DEVELOPMENT OF KNOWLEDGE BASE IN FAILURE ANALYSIS OF TECHNICAL SYSTEMS BASED ON STRUCTURAL MODELLING METHOD

Janis Grundspenkis
Department of Systems Theory and Design
Riga Technical University, 1 Kalku Street, Riga, LV-1658, Latvia

There are many methods and tools for fault diagnosis. A common approach has been to apply techniques, based on structures. A more recent phenomenon is the application of artificial intelligence in this problem domain. In two papers Grundspenkis (1993, 1994) proposed to combine these techniques in methodology of structural modelling for failure analysis of complex technical systems in conditions of incomplete information. The essence of structural modelling method is that an expert's understanding of how a system operates is organized as a representation that reflects the morphological (physical) structure of a system, its functions and parameters of each component. Three kinds of corresponding structural models are used, which are visualized by digraphs.

The development of knowledge base is divided into two phases. The results of these phases are the topological knowledge base (TKB) and the deep knowledge rule base (DKRB), respectively. TKB is developed using a systematic knowledge acquisition process where the main sources of the knowledge are documents about system and human experts. TKB is organized hierarchically and represents the knowledge as structured objects consisting of named slots with attached values. A frame hierarchy as knowledge representation language is used to encode information about systems components, their functions and behavior as well as information about relationships between components. TKB is a passive component of a knowledge-based system for failure analysis.

TKB supports both well-known inferences, namely, forward and backward chaining, because causal event chains allow to generate a tree structure from causal relations represented in that part of TKB that contains knowledge captured in a model of functional structure in a space of parameters (FSM SP). Formal method based on decomposition of FSM SP is used to derive a set of event trees. Each event tree has its root - the final event (symptom to be observed), and leaves represent possible faults of components (primary events). To make inferences effective when implemented, the set of cause-consequence (C-C) rules are used. The decomposition of each event tree is carried out in such way that each obtained subtree contains only one logical operation. Every subtree corresponds to one C-C rule. The set of C-C rules which are derived from all subtrees compiles the deep knowledge rule base (DKRB).

REFERENCES

Grundspenkis,J. (1993). The Use of Knowledge Based Approach for Simulation of Fault Propagation in Complex Technical Systems. *Modelling and Simulation 1993. Proceedings of the 1993 European Simulation Multiconference, (A.Pave, Ed.)*, pp.217-221, SCSI, USA.

Grundspenkis,J. (1994). A Methodology for Knowledge Acquisition and Reasoning about Faults in Conditions of Incomplete Information. *Preprints of IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, Vol.1, (T.Ruokonen, Ed.)*, pp.362-367, Helsinki University of Technology, Finland.

MONTE CARLO CALCULATION OF ENERGY LOSS DISTRIBUTIONS AT TILTED INCIDENCE OF ACCELERATED ELECTRONS TO SOLIDS

*Y.Gueorguiev, G.Mladenov, and D.Ivanov**
Institute of Electronics, Bulgarian Academy of Sciences,
Tzarigradsko shausse 72, 1784 Sofia, Bulgaria
**DIPI Co. GROUP, 65.B.Petkov str., Sofia, Bulgaria*

Abstract

Monte Carlo simulation (MCS) is a most widely used technique for obtaining the spatial distributions of absorbed electron energy density in resists as applied to electron beam lithography (EBL) due to its advantages over other methods. However, one of these advantages, namely the possibility of modeling the scattering of fast electrons impinging on targets of complex topography is, as to our knowledge, not properly studed and exploited so far in regard to EBL, but it may be of interest in some particular applications. This problem can be brought, as the first approximation, to the simulation of tilted incidence of accelerated electrons to solids. There are several papers addressed to the latter one [1-4] but they consider mainly backscattering of electrons and particularly energy, angular and spatial distributions of backscattered electrons. As far as we know, all results on energy deposition of fast electrons reported hitherto are obtained for normal incidence of primary electrons.

Therefore, the aim of this work is to simulate the scattering of fast electrons impinging at angles different than 90 to the target surface and to obtain spatial distributions of absorbed energy density in the resist, deposited on substrate of the material most widely used in microelectronics, namely silicon.

Such distributions in 125 nm resist layer on bulk Si substrate are obtained numerically by MCS for angles of incidence 30, 45, and 60 at two beam energies - 25 and 50 keV. The results show strong asymmetry of the distributions. Their peaks, commonly associated with the forward scattered electrons, are of lower values and 50-100 nm shifted and the shapes are different in comparison with those for normal incidence of electrons. These peculiarities of exposure distributions may lead to an enhanced proximity effect in the part of the exposed structure as well as to the absence of the proximity effect in the other part. The latter one can be used for improving the shape of the exposed pattern when a periodic (e.g. comb-shaped) structure is to be created. In all cases, if a tilted incidence of accelerated electrons to any surface occur during EBL it has to be taken into account.

Acknowledgments

The present work was partially supported by the Bulgarian National Science Fund.

References

[1] K.Murata, T.Matsukawa, and R.Shimizu, Jap. J. Appl. Phys. 10, 678 (1971).
[2] K.Murata, J. Appl. Phys. 45, 4110 (1974).
[3] S.Hasegawa, Y.Iida, and T.Hidaka, J. Vac. Sci. Technol. B 5, 142 (1987).
[4] Ding Z-J., R.Shimizu, T.Sekine, and T.Sakai, Appl. Surf. Sci. 33/34, 99 (1988).

A NUMERICAL INVESTIGATION OF A REFORMER OPERATING IN FUEL CELL POWER-GENERATION SYSTEMS

W. He
Department of Mechanical Engineering and Marine Technology
Delft University of Technology
2628 CD Delft, The Netherlands
Phone: 31 15 786978 Fax: 31 15 782460
E-mail: Wei@dutw6.tudelft.nl

ABSTRACT

Fuel cell systems convert chemical energy derived from a fuel directly into electrical energy. A reformer, which converts the higher hydrocarbons in the natural gas into hydrogen-rich gas that is generally more appropriate fuel for the fuel cells, has often been used in the fuel cell systems. A reformer operating in fuel cell systems differs essentially from its operating in process plant, where continuous full load operation is the rule. A reformer for fuel cell systems is required to operate in such a manner that the production of hydrogen-rich gas to a fuel cell always follows the changes in the demand of power. It is therefore required that the reformer should display rapid responsiveness to load changes and flexible controllability. There are still few investigations on the dynamic performance of a reformer in fuel cell systems. Hence, the dynamic performance of a reformer operating in fuel cell systems is studied here.

The main phenomena in a heat exchange reformer (HER) from Haldor Topsoe are firstly analysed, and then the modelling approach and the evaluation are carried out.

The HER uses a principle with a combination of counter- and co-current heat exchange between the process gas and the flue gas, in order to maximize thermal efficiency and to optimize usage of construction materials. Consequently, the idea of HER is to combine the two heat transfer principles in a two-bed catalyst system. That is, the heat exchange of counter-current flow is used at the low temperature side and co-current flow is used at the high temperature side. In order to reach equilibrium faster, the fuel processing reactions are performed in catalyst beds. The heat passes through the tube walls by conduction and is transferred to the catalyst bed by convection and radiation. At the same time, the chemical reactions create temperature and concentration gradients in the radial direction of the tube and around and within the porous catalyst particles.

The overall performance of a reform is related to the burner duties, the feed stream characteristics, catalyst properties, and reformer geometry and properties. The main processes of HER, are identified as:

- chemical reactions in the process and flue gas
- heat transfer from the flue gas to the process gas
- mass-transport of flue and process gases

Accordingly, a reformer model is developed, which can consider simultaneously the dominant processes of a reformer, such as chemical reactions and heat transfer as well as mass-transport. The present reformer model has been successfully implemented in the SpeedUp code.

To evaluate the reformer model, step change testing is carried out. The processed gas from the reformer determines the fuel cell power-generation. Therefore, the dynamic responses of the product-gas's flow-rate, pressure, temperature and H_2 concentration are demonstrated. Particularly the following three step change cases are used to illustrate the dynamic behaviour of the reformer model:

- flow rate of process gas ± 10%
- temperature of flue gas at inlet ± 10%
- pressure of process gas at outlet - 10%

These response results appear to be reasonable and have provided valuable insight into the characteristics of the reformer operating in fuel cell systems.

A Telecontrol Display Simulatiom Which is Composed of Virtual Environment and TV Realistic Image.

Prof. HE Han gen, XI Piang, LIU Jien Piang, Gao Zeng, ZHU Huayong
Department of Automatic Control of Changsha Inst. of Technology
Changsha, Hunan, P.R. China
FXA: +86 731 4448307 Tel: +86 731 4434720—87302

ABSTRACT:
In this paper, a new telecontrol system which is composed of Virtual Reality and TV Realistic image is described.
This system adapts an unmanned vehicle. A camera with a following-up system and a special optical system which can make two TV images, that is one is left image and another is right image is mounted on the vehicle. The TV (left image and right image) images are transmited to head-mounted binocular display system which displies the stereo environment around the vehicle.
The color three-dimensional graphics which are made by a computer display the environment inside steering room. The graphics which are separated into two graphics: one is left graphic and anather is right graphic are transmited into the head-mounted binocular display system too.
Then the virtual reality and the TV image make a truly realistic pictures of telecontrol environment.
In order to develop above system, we must solve the following key techniques:
1: Research on a fast matching algorithm which can complete the match of virtual environment and TV image in real-time.
2: Research on a knowledge-based hidden algorithm whose time-coplerity is so low that it can be finished in real-time.
In order to use this system for simulating driving a car along some road, we can use the special TV camera system and a videorecorder to recorde the sceneries which stand by the side of the road, when we drive a vehicle along the road.
Then using the videorecorder image and the graphics which display the environment inside steering room, we can simulate the full process of drving a vehicle along the road.

Electro-Magnetic Field Simulation of Cable in Metallic Duct

Junya ISHII[a], Katsuhiko AOKI[b], Shigeki HONTSU[a] and Masamitsu ASAI[a].

[a]Kinki Univ., Faculty of BOST, Uchitacho, Wakayama, Japan, 649-64, Fax:+81-736-77-5745,

[b]President of NIHON ELECTRIC WIRE & CABLE Co. Ltd., 2-2-302, Suminodo, Daito City, Osaka, Japan, 574, Fax:+81-720-71-0341.

An accident was reported of parched cable at the bend in the metallic duct for the use of instrumentation of a factory in Japan. The cross sectional area of metallic duct is the closed rectangular form of 250 × 200 mm where 62-wires are installed. The current is $3.5(DC)+25.5\cdot\sin(2\pi ft)$ [A] for each wire, where f=26kHz. The cable was manufactured at 1989. The parched part of the cable was discovered in the part of bend at 1994. The bend is the tansition between horizontal and vertical part where the wires were loosely bound but tightly at the part of bend to fix. Detailed discussion was made[1].

Here, the useful results are reported by using electro-magnetic field simulation software named "hfss" istalled in HP 9000 series 750 CRX APOLLO. The electro-magnetic fields are observed in consentarion at the part of bends, resulting overheating mainly by resistance of wires. The diagram gives the useful data of current density versus the angle of bend of cable for future installation of cable in the factory. In general, the bridging is the future interesting theme between the electro-magnetic field approach and the heat conduction in some materials.

Reference: [1] Report on the damage of HKPCV cable, Nihon Electric Wire & Cable Co. Ltd, 28, April, 1994.

AN ADAPTIVE USER INTERFACE FOR CONSULTATION OF A MULTIMEDIA DATABASE. APPLICATION IN PRINTED TEXTILES.

Saadallah JOMNI, Amir HAJJAM, Gerard METZGER
ESSAIM - LSI
12, rue des frères Lumière
68 200 MULHOUSE - FRANCE
Tél. (33) 89 59 69 75
Fax. (33) 89 59 69 69

Abstract:

An adaptive man-machine interface requires a dynamic user-model and an adaptation strategy in agreement with the logic of utilization.

The work presented in this paper deals with the consultation of the multimedia environment that we have developed for the MISE[1]. This environment, with its three software programs, allows us to build and to present the museum's collection of fabric patterns. These 3 million fabric patterns are accompanied by extracts of textual and aural information in several languages .

The users of this environment are from various origins (textile manufacturers, scientists, students, etc....) and have very different objectives. Choosing a pattern can take from a few hours up to a few days. In this context, the interface has to take account of the user's behaviour or help him in his choice, thus minimizing the consultation time. In order to achieve this goal, we are developing a modeller to add to the interface of our environment which defines a model describing the user, according to his tastes and needs. This modeller is based on an annotated knowledge graph and is used to gradually replace the functioning logic of the system by the logic of utilization.

Thus defined, this interface will be capable of following the evolution of the different media used by the environment as well as the user's behaviour.

Keywords:

Consultation, User model, Adaptive user interface, Modeller.

[1] Musée de l'Impression Sur Etoffes (Museum of Printed Textiles - Mulhouse - FRANCE) . The MISE has a collection of 3 to 4 million of textile samples dating from the second half of the 18th century.

EXPERT KNOWLEDGE IN THE MODEL DEVELOPMENT PROCEDURE FOR THE NITRENDIPINE SUSTAINED RELEASE DOSAGE FORM DESIGN

R.Karba, * A.Mrhar, M.Atanasijević-Kunc
Faculty of Electrical and Computer Engineering,
Tržaška 25, 61000 Ljubljana
* Faculty of Pharmacy
University of Ljubljana, Slovenia

The goal of the present study is to develop the sustained release dosage form for Nitrendipine (only one dose per day) by the aid of pharmacokinetical modelling and computer simulation stressing the points where the inclusion of expert knowledge is crucial for the successful course of the study.

Here the nitrendipine can be introduced as a modern antihipertensive drug being administered via peroral dosage form. It is known that it has linear kinetics with complete absorption to the portal circulation and extensive metabolism, which represents the main route of elimination from the plasma. Very precise analyses with gas chromatography indicates the possibility of deep peripheral compartment existence which maintains pharmacodynamical effects of drug by the slow transportation back to the central compartment-plasma.

Mathematical modelling procedure based on the data of the four dosage forms of nitrendipine namely: intravenous application, oral application of the solution in the capsule, tablet with normal and tablet with sustained release. So by the aid of expert knowledge, the nine compartment model was developed including shallow and deep peripheral compartments which represents the mechanisms of nitrendipine acting. In the procedure of model development the very interesting fact came out. Namely the model responses were matched very well to the measured data (plasma concentrations time profile) only by changing the dissolution rate constant what shows that the dosage form is mainly responsible for the nitrendipine kinetics. Finaly the dissolution rate constant was found, which enables nintrendipine sustained release dosage form design for only one dose per day regimen.

DETERMINING THE CONDITIONS OF RELIABLE OPERATION OF CERMET HEATING ELEMENTS BY MEANS OF SIMULATION

Andrey O. Kostikov, Dr. Oleg S. Tsakanyan

Institute for Problems in Machinery of National Academy of Sciences of Ukraine

2/10 Pozharsky 310046 Kharkov Ukraine

Abstract.

At the present time the cermet heating elements (CHEs) are widespread. They are used in various devices and apparatuses from household appliances to thermostating systems of spacecrafts.

CHE has the form of a plate in which resistance element is placed. This resistive element is a wolfram-molybdenum conductor. The electric leads are soldered to lands on CHE surface. Such a structure of the CHE possesses the number of valuable properties (possibility to work in aggressive media and with high temperature, satisfaction to ecological standards and conditions of electric safety). However, one of essential lacks is the destruction of ceramic plate at relatively small temperature gradients $(\mathrm{grad}\,T)_{cri}$ which cause the appearance of thermal stress. The value of electric current has essential influence on reliable operation of the CHE too. Exceeding the permissible value of current can cause the burning of the resistive track (the wider the track the grater the permissible value of current I_{perm}). There is also a limitation on the temperature of heater, for example at the zone of attaching the electric leads. Exceeding the temperature at this zone over a permissible value T_{perm} causes the disconnection of leads.

It is not a success to determine the values of critical thermal stresses and critical temperature gradients with necessary accuracy by direct measuring because of the number of technical and technological difficulties. It is possible to determine only temperature at CHE surfaces and electrical performance by direct measuring.

Temperature gradients, temperature field and the value of electric current depends on voltage U applied to the CHE and intensity of heat removal from its surfaces, which is specified by environment temperature T and heat transfer coefficient h. Therefore the life area of the CHE is determined by three conditions

$$(\mathrm{grad}\,T)_{cri} \geq \mathrm{f}_1(U,h,T),$$

$$I_{perm} \geq \mathrm{f}_2(U,h,T),$$

$$T_{perm} \geq \mathrm{f}_3(U,h,T).$$

Determining the life area of the CHE in accordance with regime parameters allows to reduce experimental-calculating investigations (such performances as fatigue strength and lifetime), number of regime tests and, hence, to reduce the cost of investigations.

The mathematical model of the thermal process taking place in the CHE, the calculating procedures of the determination of its thermal and electric performances, the calculating procedure of determination of its life area in accordance with regime parameters and calculated life areas of several CHEs are presented in the paper. It is also described the experimental-calculating procedure of identifying the intensity of heat removal from the surfaces of CHE with the simultaneous determination of thermal and electric performances on the basis of the temperature measured when thermocyclic tests are carried out. The abovemantioned procedures are realized as software that allows to carry out thermocyclic tests of CHEs at workstation for the investigation of their operation.

MULTILEVEL SIMULATION OF A MILITARY MESSAGING PROTOCOL OVER TACTICAL COMMUNICATIONS NETWORK

Stephen C. Cook, Miro Kraetzl, Marcel L. Scholz

Communications Division
Defence Science and Technology Organisation
Salisbury SA 5108, AUSTRALIA

Tactical military communications networks differ in many significant ways from conventional computer and communications networks. Unlike civil radio communications systems that are carefully planned and have well placed antennas, military radio systems have no such advantage and also have to cope with high levels of interference, both deliberate and otherwise. Thus tactical communications links are, and will continue to be, less than ideal. Traditionally, military networks employed purpose-designed protocols and architectures that took the nature of the links into account. Tactical networks also differ from their civil counterparts in that the network topology is not fixed and can be highly dynamic because of failing links and node mobility. In addition, military requirements such as avoidance of single-point failures mandate the use of automatic, distributed network control. Thus gaining an understanding of the performance of military networks is not a simple matter particularly as much of the network modelling and simulation work performed by the civil communications research sector is not entirely applicable. In this paper we discuss our philosophy on how to model tactical links and networks and illustrate it using civilian messaging protocols as our network application.

Messaging protocols comprise a methodology to exchange messages between originators and recipients. Currently the main example of a layered system of protocols is OSI (open systems interconnection) which has been developed by the ISO organisation as a framework for development of protocols. Common protocols are TCP/IP, X25, X400 and all of them satisfy basic requirements in order to provide a message exchange service (the message must not get lost or be changed in transit, the entities handling the message must be able to interpret the message, etc.)

In specific, military tactical environment some additional factors determine the efficiency of message handling protocols. Messaging security issues are of the prime importance, protocol overheads definitely slow down the throughput of information. Confidentiality, integrity (the content of received message is the same as that which was originated), data origin authentication (the message was originated by the user as indicated by sender), access control (messages sent and received must not violate the security policies of either originator or the recipient) are just a few desirable properties of such message handling. Directory service availability is also a slowing factor in the performance of military tactical messaging, as well as the provision of the message retention (maintaining the time traceability of documents exchanged over a network).

Multiple communications architectures that have to be used in tactical situation influence the messaging efficiency. We simulate (using OPNET) messaging architectures where links are established using the HF, satellite, cable, line-of-sight, fibre optics, ATM or other communications media. We explore the efect of additional overheads over a broad range of known protocols, as well as some new ones.

SIMULATION OF THE MECHANICAL BEHAVIOUR OF DENTAL IMPLANTS

Paolo Marchetti[1] – Mario Martignoni[2] – Luigi Baggi[2]

University of Rome "Tor Vergata", Via della Ricerca Scientifica, 00133 Rome (It)
[1]Department of Civil Engineering [2]Faculty of Dentistry

Man's chewing mechanism is a biomechanical system where the different components (mandible, muscles, teeth, etc.) are subjected to various kinds of forces and are shaped in such a way as to bear and resist to such loadings. Loss of the natural teeth brings about the need for a replacement structure which is usually embedded in the jaw bone by means of abutments (osseointegrated prosthèsis). The close apposition of bone to implant fixtures results in a stiff connection that transmits chewing forces through the implants directly to the surrounding bone of the mandible, without such natural stress-absorbing elements as the parodontal ligament. This may lead to the bone being overstressed with the likelihood of bone-resorption phenomenon, loosening of the implants and even microfractures occuring. The study the biomechanical interaction of mandibles with endosseous implants becomes then the basis for designing long-lasting and stable prostheses. In recent years quite a few studies have been aimed at determining the major factors of influence on the stress concentration. Among those: the number of implants, their length and the presence of a more or less rigid connecting bar. The object of this study is the stress distribution in a mandible with implants in both cases, with and without connecting superstructure. In particular, different implants' typologies and different load cases were compared in order to quantify the state of stress occuring both in the implant and in the mandible.

Materials and methods

A three-dimensional linearly elastic model with hexahedral elements was built and implemented in a Finite Element Analysis Program. The jaw's cross-section's geometry was surveyed by slicing the chalk mould of a real jaw; afterwards it was meshed into two distinct zones so that the outer cortical bone and the inner spongy bone could be taken into account. This cross-section was then projected into a third direction to form the basic three-dimensional element into which the abutment can be inserted. The three dimensional development of the jaw was described with enough accuracy by a parabola. As a result the basic three dimensional element was copied in a number of point described by this parabola. The last step was simply connecting the three dimensional elements. In total, the model consists of 1272 elements with 1784 nodes. In the model both the tissues, the abutments and the bar were regarded as homogeneous, isotropic, linearly elastic materials. The load cases considered were a 70 kg, static and axial force on the solitary abutment, on the connecting bar and finally on the extension of the bar beyond the abutment (this is an ultimate value for the force). All of them were combined with an horizontal component.

Results

The stress distribution within the elements of the abutments and connecting bar was expressed in terms of Von Mises equivalent stress, whereas for the bone it is important to quantify the tensile stress which leads to biological resorption. It is necessary to point out that among the parameters deemed influent on the stress distribution we disregarded the boundary conditions because the distance from the abutment is such that De Saint Venant's hypothesis are valid and the connecting bar's stiffness which practically the same for a gold (180 HV) or titanium alloy. The results indicate that the largest stress concentration takes place when the cantilever arm of the connecting bar is loaded. The values of Von Mises equivalent stress in the connecting bar are about one and a half times those obtained by loading in between two abutments and twice those obtained by loading the single abutment, reaching the material's failure values. In this case the force is not distributed in the mandible and the failure mechanism will involve the node connecting the bar and the abutment. On the other hand the worst loading condition for the bone is the force distributed on a solitary abutment. The largest compressive stress is located in the upper cortical layer and decreases rapidly into the spongy bone. This result can be explained by the relative (10:1) high Young's modulus of the cortical layer and confirms why the loosening of the implants occurs in the upper portion of the implant stem. Finally, the values of displacements and tension in the other abutments were found in order to understand how the stress is distributed by the superstructure and how, under cyclic loading conditions, this causes the resorption of other abutments.

Simulation of Dynamics and Control of an Electromechanical Robot With Elastic Elements

Sergei MIKHAILOV
Institute for Problems in Mechanics
the Russian Academy of Sciences, Moscow, Russia

The control of rotary motions of the last link of manipulator is considered within the framework of elastic beam model. We consider the lowest elastic vibration mode. In the basis of the assumptions mentioned the equations of motion can be represented in the form :

$$\varepsilon\ddot{\alpha} + \ddot{\alpha} + \dot{\alpha} = U(t) \quad (1)$$

$$\ddot{x} + \omega^2 x = -\ddot{\alpha} \quad (2)$$

Here α is the d.c. motor angle, x is the elastic displacement, U is the voltage in the d.c. motor, ε and ω are pure parameters. Points denote time derivative.

The following control problem is considered for the system (1)-(2): the voltage of electric motor is assumed to be the control function $\mid U(t) \mid \leq 1$; at the initial moment of time $t=0$ the values of state variables are given; the objective is to turn the beam to the preassigned angle α^0 and to reduce elastic displacement; the functional characterizing the control quality is the time of operation T, $T \Rightarrow min$.

The smooth control for the system consisting of vibration link and the electrodrive is constructed . The time optimal control for the rigid link is bang-bang. This means that applying control for the rigid link in the system (2) the large amplitudes of the elastic vibrations will be obtained. We would like therefore to correct bang-bang control to avoid the large elastic deflections on one hand and to reach higher productivity (minimum time of working operation) on the other hand. This can be made by means of the smoothed control. The input of the vibration link is angle acceleration $\ddot{\alpha}$ therefore at the moments of the control switches, large elastic displacements occur. Our aim is to smooth angle acceleration near the points of control switches. The elastic oscillations can be remarkably reduced by means of rational choice of the smooth control functions. In this case low elastic deflections and insignificant increasing the time of operation with respect to bang-bang (time-optimal) control can be achieved. System of fifth order (1)-(2) have been numerically integrated by Runge-Kutta method with automatical control of step and accuracy.

Conclusions. The outcomes derived by numerical computation are as follows: the amplitudes of elastic vibrations and values of current at the moments of switch during bang-bang control are rather big, which makes difficult the realization of this control; the elastic displacement of the link can be remarkable reduced using smoothed control. As a result the elastic displacement of the link can be practical removed ; almost the same time of operation as the bang-bang (time-optimal) control can be reached using the smoothed control. The time of operation slightly increasing (approximately 10%).

SIMULATION ANALYSES OF IRRATIONAL TEMPERATURE STATE IN THE MAIN COMBUSTION CHAMBER OF CFBC BOILER

Weidou Ni, Zheng Li, Guangxi Yue and Xin Sun

Department of Thermal Engineering

Tsinghua University

Beijing 100084, P.R. China

ABSTRACT

Presently, irrational temperature state in the main combustion chamber is one of the serious problems which hinder the development of Chinese domestic CFBC boilers. Generally speaking, the temperature is quite high at bottom part and rather low at the top part of the main combustion chamber.

With a view to solving this problem, it is proposed to increase the mass fraction of fine particles in feeding coal. Such a measure is thought to be able to increase the solid entrainment and cause more coal particles to burn at upper part of the chamber.

In order to verify the rightness of this proposal, the general mathematical model of CFBC boiler of Tsinghua University is applied in the simulation of a 50 MW_e CFBC boiler. The model has the capability of considering the broad size distribution of feeding coal and bed inventory. Three operation cases with different size distribution of feeding ash are simulated. It is turn out that, due to the restriction of the characteristics of separator, the temperature state of the main combustion chamber can not be always improved by increasing the mass fraction of fine particles in the feeding ash. This is true only when the mass fraction of those fine particles which are the main body in the circulating ash and could be well separated by the separator increases. Hence, an important understanding is recognized, i.e. **the particle mass composition of the dense bed is the ket factor which determines the entrainment rate and moreover the teperature level and profile of the main combustion chamber.** At the same time, it is also recognized that the low separator efficiency is responsible for the irrational temperture state and has become one of the primary restraints for the development of domestic CFBC boiler of China.

In order to clarify the effect of separator efficiency on the behavior of CFBC boiler, simulation is carried out with separators with different efficiency. When separator is improved, the following changes take place. Firstly, the mass fraction of fine particles increases in dense bed. Secondly, the solid concentration increases in the upper part of the chamber. Thirdly, the teperature level lowers down and the temperature difference between the bottom and top parts of the main combustion chamber decreases. In other word, when the separator is improved, the behavior of the CFBC boiler is also improved.

Therefore, it can be concluded:

1. the particle mass composition of the dense bed is the key factor which determines the entrainment and moreover the teperature level and profile of the main combustion chamber;
2. the temperature state of the main combustion chamber could not be surely improved by increasing the mass fraction of fine particles in feeding ash;
3. low separator efficiency is responsible for the irrational temperture state and has become one of the primary restraints for the development of domestic CFBC boilers of China.

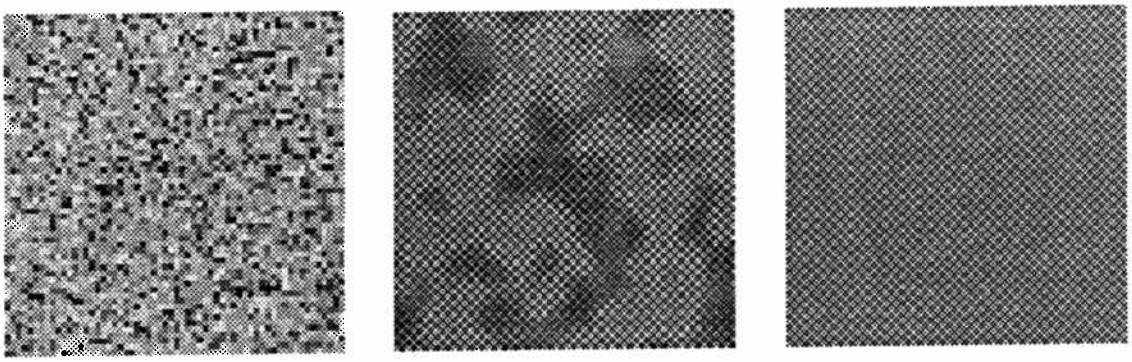
Controlling Spatio-Temporal Chaos in Two Dimensional Copuled Map Lattice

Yuji Ohishi, Hirotada Ohashi, and Mamoru Akiyama
Department of Quantum Engineering and Systems Science,
University of Tokyo, 7-3-1 Hongo, Bunkyou-ku, Tokyo 113
E-mail address: ohishi@crimson.gen.u-tokyo.ac.jp

Recently many studies of controlling chaos have been done and its application to higher dimensional systems, in which the space structure becomes important, is coming to the attention.

Taking consider of controlling spatio-temporal chaos, all-site control is ideal, but not practical. Therefore, local controlling is thought to be necessary. Hu Gang et al. suggested a pinning method which is one of the local control methods, and confirmed its effectiveness in the one dimensional Coupled Map Lattice (CML)[1]. However, in experimental systems most of the spatio-temporal chaos is two dimensional. In addition, more complex behavior, probably corresponding to a higher chaotic potential, is expected, so that two dimensional controlling is more realistic and useful, provided that the successful control can be realized.

For this reason, we extended the pinning method into two dimensional CML. With computer simulation of CML, each site of which was described by a chaotic logistic map, we investigated the effect of pinning density and the pinning pattern in the CML. Some of the simulation results are shown in Figure 1.



(a) 0 step (b) 10 step (c) 120 step

Figure 1. An example of successful controlling in two dimensional CML. We can change the pattern from (a) spatio-temporal chaos into (c) checker board. In this case pinning density is 0.5.

Refereee

[1] Hu Gang and Qu Zhilin, Phys. Rev. Lett. **72**(1994)68

Automatic Configuration of Electronic Circuit

Yuuichi ONO and Nakatomo FURUHASHI

We reported a new electronic circuit simulation program for the investigation of analog and digital circuits. The program was written by use of the C language, so it can be easily implemented on many kinds of computers ranging from personal computers to supercomputers.
If a good idea for a new circuit comes to our mind, the characteristics of the circuit can immediately be checked out by the simulation program. The function of the circuit is determined and can be adapted to design targets.

But, in case of no circuit diagram, the design of the circuit cannot start. Present electronics circuit simulation system is forming by bottom up, namely electronics circuit constructed by small functions. In an analog circuit, the design of an excellent circuit needs high experience and long learning time. Our development aim is to get easy design methods for electronic circuits.

On the other hand, various electronics circuits are made in IC. These ICs are used not only in the electrical and the electronic field, but also in the mechanic field, e.g. for robots, machine tools, motorcars and so on. Engineers except from the electrical and electronic field are easily trained in the use of the simulation program for electronic circuits.

COMPUTER AIDED ANALYSIS AND SYNTHESIS OF THE CONTROLLABLE VEHICLE SUSPENSION

ALEXANDER OSTANIN, ALEXANDER LEOSHIN

Department of Computer Science, Technical University of Bialystok, Wiejska 45A str., 15-333 BIALYSTOK, POLAND. Email: ALEKS@ii.PB.BIALYSTOK.PL

Vehicle suspensions must satisfy several conflicting purposes. Besides supporting the vehicle weight, the suspension should isolate the body from vibrations resulting from road unevenness and maintain firm contact between the tires and the road. Moreover, these tasks must be accomplished under a number of constraints. Suspension systems built to perform these tasks can be roughly divided into passive, semiactive and active [1].

Passive suspensions are those found on most conventional vehicles. Roughly, they are characterized by the absence of external power sources, whereas the active suspensions require additional sources such as compressors and pumps, to achieve superior ride and handling performance. The semiactive suspensions then fill the gap between their passive and active counterparts, since they essentially act as time varying dampers and, thus, offer potentially significant performance improvements, while requiring relatively small external power sources. The advantages of semiactive suspensions are low cost and good reliability. The active and semiactive suspensions allow adaptation to changing road surface quality and driving situations and they are now under development at nearly any major car manufacturer. The semiactive and active suspensions are named here "controllable".

Paper deals with the next problems of computer aided analysis and synthesis of the controllable vehicle suspensions: building of the mathematical vehicle, road models and their computer representation; task formulation for optimal control law synthesis and it design; working out of the methodics for closed loop system quality analysis; using of the advanced investigations during training for students.

The paper presents a structural approach, used for creation of the base with car suspension models, which is integrated with MATLAB (The Math Works Inc.). Base consisting of the next car model classes [2] : lumped parameters, space (a full-car), plane (a half-car, a quarter-car, etc.), one mass, many-mass, nonlinear, linear, passive, semi-active, active. It was used four levels during base creation: structural level I, structural level II, analytical level and software level. The base is open. It is planned to add distributed parameters vibration isolation models.

As an simulation example a lecture presents investigation results of isolation effectiveness roll, pitch and bounce motion at independent active car suspension. The vehicle model presented in the research work comprises six bodies: a driver seat, a sprung mass and four wheel-axle assemblies (unsprung masses). The car is assumed to travel at a constant velocity. The driver seat has vertical degree of freedom, the sprung mass is modeled with three degrees: roll, pitch and bounce, and the unsprung masses each has a vertical degree.

The results of this investigation make it possible to give recommendations for application control vibration isolation to a car. It are used during training for students at the courses: "Simulation and Optimization of Dynamical Systems", "Modern control theory".

REFERENCES:

1. Hrovat D. Applications of Optimal Control to Advanced Automotive Suspension Design. /Trans. of the ASME. Jour. of Dynamic sys., meas. and control, Vol 115, June 1993, pp. 328-342.
2. Ostanin A., Leoshin A. A Hierarchical Database for Simulation of Vibration Isolation Systems. /Modelling, Measurement & Control, B, AMSE Press, Vol. 59, No 2, 1995, pp. 57-63.

SIMULATION OF POWER ELECTRONIC CIRCUITS BASED ON HOMOGENOUS STATE SPACE REPRESENTATIONS

Adriana Pastravanu
Faculty of Electronics and Telecommunications, Technical University of Iasi, 6600-Iasi, ROMANIA

1. Introduction

Power electronics circuits are usually represented as cyclically switched systems, each *cycle* being modeled by a sequence of *modes* governed by the commutation of the switching devices (diodes, thyristors, GTO's etc) presuming they operate as ideal switches. The commutations have two distinct natures : *controlled commutations* (which occur as a result of external commands, representing the actual control function of the circuit) and *uncontrolled commutations* (which occur as a result of the circuit dynamics). Each mode lasts between two successive commutations, regardless of their nature and is characterized by a unique topology. In computer-aided design of power converters with complex structure, the simulation of the circuit behavior in the vicinity of a steady state plays an important role.

The usage of a general purpose differential equation solver for reaching the steady state exhibits many disadvantages due to the a priori unknown commutation times. Some approaches are exploiting the fact that the topology corresponding to any mode can be described by a linear and time invariant state space model, but the manipulation of these models becomes cumbersome for high order systems when many circuit topologies are sequenced within a cycle.

2. Basic Results

Assume that N_k modes are sequenced within the k-th cycle and the commutation times are given by $\tau_1^k < \ldots < \tau_l^k < \tau_{l+1}^k < \ldots < \tau_{N_k}^k < \tau_1^{k+1}$. Consider the normal form of the state equation corresponding to the topology of the l-th mode which lasts between τ_l^k and τ_{l+1}^k

$$\dot{x}_l^k(t) = A_l^k x_l^k(t) + B_l^k e(t) \quad , \qquad (1)$$

with A_l^k nonsingular and its spectrum located in $\mathrm{Re}\, s \le 0$. Since e(t) has a particular form, e(t) can be expressed as

$$\dot{u}(t) = Eu(t) \quad , \quad e(t) = Du(t) \quad , \quad u(0) = u_0 \qquad (2)$$

Consequently, (1) may be transformed into a *homogenous state space representation:*

$$\tilde{\dot{x}}_l^k = \tilde{A}_l^k \cdot \tilde{x}_l^k \qquad (3)$$

Denote by Φ_l^k, Γ and $\tilde{\Phi}_l^k$ the *transition matrices* of (1), (2) and (3) respectively, corresponding to an arbitrary time interval $t_0 < t_1$, with $\tau_l^k < t_0 < t_1 \le \tau_{l+1}^k$. Denote by $\tilde{S}_l^k$ the commutation matrix at τ_l^k which ensures the link $\tilde{x}_l^k(\tau_l^k + 0) = \tilde{S}_l^k \tilde{x}_{l-1}^k(\tau_l^k)$.

The following two propositions will be further exploited :

(P1) : $\tilde{\Phi}_l^k$ can be expressed in terms of Φ_l^k and Γ.

(P2) : Between two arbitrary points of the state space trajectory there exists a linear mapping expressed in terms of an appropriate product of transition ($\tilde{\Phi}_l^k$) and commutation ($\tilde{S}_l^k$) matrices.

3. Case of A Priori Known Sequence of Modes

Denote by N the total number of commutations within a period of a steady state; introduce the vector of controlled commutation times, $\theta^T = [\, \theta_1 \ \ldots \ \theta_M \,]$, M<N, and the vector of uncontrolled commutation times $\sigma^T = [\, \sigma_1 \ \ldots \ \sigma_{N-M} \,]$. The steady state value of the normal state vector at $\theta_1 = 0$, denoted by $x_0 \in \mathbb{R}^n$, satisfies the matrix equality

$$x = [\, I_n \ \ 0 \,] \tilde{\Psi}(\sigma, \theta) [\, x \ \ u_0 \,]^T \quad , \qquad (4)$$

where the structure of $\tilde{\Psi}(\sigma, \theta)$ results from (P2). On the other hand, the uncontrolled commutation conditions :

$$c_j(x, \sigma, \theta) = 0 \quad , \quad j = 1..N-M \qquad (5)$$

can be also expressed in terms of (P2). The values of x and σ can be determined as solutions of the algebraic system (4) and (5), since θ and u_0 are known. In practice, spurious solutions may be avoided regarding (4) as a nonlinear optimization problem:

$$\min_{x, \sigma} \left\| x - [\, I_n \ 0 \,] \tilde{\Psi}(\sigma, \theta) \begin{bmatrix} x \\ u_0 \end{bmatrix} \right\|_2 \quad , \qquad (6)$$

with equality constraints (5) and range-type constraints for x and σ (resulted from a priori knowledge of circuit operation). From (P2) the analytic derivatives are available.

4 . Case of A Priori Unknown Sequence of Modes

The uncontrolled commutation conditions must be checked permanently until the steady state is reached. The proposed method relies on (P2); for a given time step h, the simulation advances through left multiplication by $\tilde{\Phi}(h)$. If h satisfies the condition $h \cdot |\, \lambda_j \,| << 1$ for all the eigenvalues λ_j of the homogenous state-space representation (3) of the current mode, then $\tilde{\Phi}(h)$ has an accurate linear approximation for $t \in (0,h)$. Therefore any commutation condition may be approximated for $t \in (0,h)$ by a linear scalar function whose coefficients are modified at each simulation step. Thus, once a change of sign of a commutation condition is detected, a fast estimation of the corresponding commutation time is given by the unique solution of the linear approximation.

5. Behavior in Vicinity of a Steady State.

In the vicinity of a steady state, characterized by x, σ and θ, the sequence of modes is preserved. Therefore a small signal model can be derived in terms of perturbations δx, $\delta\theta$ for two successive cycles :

$$\delta x(k+1) = F\, \delta x(k) + G\, \delta\theta(k) \qquad (7)$$

(P2) is extremely useful for the computation of F and G directly by the linearization of (4) and (5) around x, σ and θ. The asymptotic stability of linear sampled system (7) with $\delta\theta(k) = 0$ represents an important criterion for rejecting the spurious solutions to steady state. Time and frequency domain analysis of (7) give a deeper insight into the role played by controlled commutation times in achieving the desired steady state performances.

7. Concluding Remarks

Exploiting the basic properties of the state space homogenous representations, the simulation method and the procedures for steady state, analysis have been successfully applied to power electronic circuits with complex structure (e.g. the PWM AC/DC converter presented in [1]). It is worth noticing that the framework developed in this paper increases the flexibility and applicability of some concepts and results in [2] which, in their initial form, are not suitable for higher order systems and many topologies sequenced within a cycle. The algorithms can be simply implemented in MATLAB and allow a detailed analysis of the influence of different modes upon the global behavior, problem that usually cannot be addressed using dedicated circuit simulators.

REFERENCES

[1] Alexa, D., Pastravanu, A. - "AC/DC Current Source GTO Converter with High Power Factor", ETEPE, vol.3, No.6, pp.421-428, Nov./Dec. 1993

[2] Verghese, G., Elbuluk, M., Kassakian, J. - "A General Approach to Sampled-Data Modeling for Power Electronic Circuits", IEEE Transactions on Power Electronics, vol. PE-1, no. 2, pp. 76-88, Apr. 1986.

MATHEMATICAL MODELING OF HYDRAULIC SERVOMECHANISMS: ROBUST SYNTHESIS AND SIMULATION

by Florica Popescu - Scientific Researcher - IMFDZ - Bucharest
Felicia Ursu - Scientific Researcher - STRAERO - Bucharest
Ioan Ursu - Scientific Researcher - IMFDZ - Bucharest
Mihai Vladimirescu - Scientific Researcher - IMFDZ - Bucharest

ABSTRACT

The electro-hydraulic servomechanisms of the control chains of modern aircraft have to cope with ever increasing requirements of stability, response time, band-pass width, damping of the transient regime. There are paper belonging to the specific technical literature: H.V.Panasian-"Reduced order observers applied to state and parameter estimation of hydro-mechanical servo actuators" Journal of Guidance, Control and Dynamics 2/1986; S.A.Ermakov s.a."Proektirovanie ustroistv korrektii elektroghidravliceskih slediascih privodov na osnove nabliudateliu sostoianiia" Pneumatica I ghidravlika, 12/1986, Moskva, Masinostroenie, which have already refereed to the application of the modern theoretical methods of the optimal and robust control to electro-hydraulic servomechanisms. These methods generally improve the performances of the servomechanisms acting as automatic tracking systems in the presence of the perturbations transmitted in the input and output, as well as, of the modelling incertitude. The numerical applications of this paper are an example of these assumptions. Starting from some requirements regarding the frequency response of the servomechanism we can design an optimal full-state knowledge LQG regulator that will have good qualities concerning the robustness (gain margin infinite and phase margin more than 60^0). These qualities are lost when we use observers. They would be recovered asymptotically using the LTR method that results from the assumption that the intensity of measurement noise tends towards zero. The computational method elaborated on "MATLAB-ROBUST TOOLBOX" was completed as a design stage to derive the LQG/LTR compensator. The behaviour of the electro-hydraulic servomechanism model together with the optimal LQG/LTR regulator is simulated both for the linear and the non-linear models in order to validate the theoretical results. We made a comparison between the behaviour of the electro-hydraulic servomechanism with LQG/LTR controller and a servomechanism with mechanical displacement feedback loop designed by Dowty Boulton Paul Ltd. (having the same design frequency domain requirements) to be used as an actuator of the stabiliser of the IAR 93/ORAO fighting plane.

Risk Analysis in Crop Production

(Abstract)
Peter Racsko, University of Veterinary Science
Budapest

Crop production in Hungary is highly dependent on climatic conditions. Local ecological conditions and proper agrotechnology can reduce the risk, associated with the production.

Crop growth simulation models have been widely used for prediction of the yield, depending on environmental conditions and agrotechnology. Our model combines the "traditional" growth models with a stochastic "weather generator".

The "weather generator" in one run produces a multidimensional time series, whose components are the most important daily weather parameters. The set of time series, received in several runs, is statistically close to the data at a given meteorological station, measured for several decades.

The weather generator of course must be identified for the given data set prior to its application.

The set of pseudo random time series provides a tool for constructing a probability distribution of a deterministic crop growth simulation model.

The presentation demonstrates the main parts of the crop growth, and the weather generator model, and also probability distributions, derived from the model.

MATLAB-BASED SIMULATION OF OPTICAL SOLITON PROPAGATION

Atef M. Rashed, Jan Engelen, and Guido François
Katholieke Universiteit Leuven, Departement Elektrotechniek-ESAT / TEO,
Kardinaal Mercierlaan 94, B-3001 Heverlee, Belgium

Adopting the nonlinear regime in the optical communication instead of the linear regime aims at overcoming the effects of dispersion and loss on the transmitted bit rate. Hasegawa and Tappert [1] were the first who proposed the utilization of the nonlinear Kerr effect to compensate the negative group velocity dispersion (GVD) in the single mode (SM) optical fibers. The optical pulse which carries out such balance between the GVD and the fiber nonlinearity was shown to form an envelope soliton. The dimensionless nonlinear Schrödinger equation (NLSE) which describes the optical soliton propagation through a lossless fiber is:

$$i\frac{\partial u}{\partial \zeta}+\frac{1}{2}\frac{\partial^2 u}{\partial s^2}+|u|^2 u=0 \quad (1)$$

where ζ , and s are the normalized quantities corresponding to z (axial distance), and t (time) respectively. u is the dimensionless wave amplitude. The dimensionless variables are related to the real world quantities by:

$$\zeta = z/z_C \text{ and } s = [t-(\partial k/\partial \omega)z]/t_C \quad (2)$$

where $z_c = 0.322\frac{2\pi c}{\lambda^2}\frac{\tau^2}{D}$, and $t_C = 0.567\tau$. c is the light speed in vacuum, τ is the intensity full width half maximum, λ is the wavelength, and D is the group velocity dispersion. k is the wave number, and ω is the wave frequency. We note also that the soliton period, which is the distance over which higher order solitons repeat their shapes, is defined as $z_0 = (\pi/2)z_C$.

We demonstrate the use of Matlab [2] functions and environment to simulate the soliton pulse propagation through the optical fibers. We have solved the NLSE by using the split-step fast Fourier transform numerical method [3]. The algorithm splits the fiber length into small segments, each of length $\Delta\zeta$, the pulse is first propagated with the GVD effect only till the end of the first half of the segment where the nonlinear effect is added, and then repropagated again through the second half with the GVD effect only.
The propagation of different orders of solitons through SM fibers has been analyzed. The analysis has been done on the basis of the evolution of the pulse shape, the pulse amplitude, and the pulse width. The source code was written inside the Matlab environment to make use of its efficient functions and graphical output.
The graphical output of the program is dedicated to monitor the change in soliton shape during propagation by dividing the screen into several figures, each shows the pulse at certain distance. Our results have shown a good agreement with the theory and the published literature. A comparison of the main pulse characteristics calculated by our algorithm and the analytical solution of Eq.(1) for some soliton orders was also made to confirm the accuracy of the Matlab functions.

References

[1] A. Hasegawa, and F. Tappert, "Transmission of stationary nonlinear optical pulses in dispersive dielectric fibers. I. Anomalous dispersion", *Appl. Phys. Lett.*, vol. 23, no. 3, pp. 142-144, Aug. 1973.

[2] "MATLAB: High-performance numeric computation and visualization software: Reference guide", *The Math Works, Inc.*, 1992.

[3] M. J. Potasek and G. P. Agrawal, "Self-amplitude-modulation of optical pulses in nonlinear dispersive fibers", *Phys. Rev. A*, vol. 36, no. 8, pp. 3862-3867, Oct. 1987.

Simulation models of plants for grain primary processing

P.Rivza

Institute of Informatics, Latvia University of Agriculture,
2 Liela street, Jelgava,LV-3001,Latvia

Simulation models may offer an effective remedy while designing plants of grain primary processing.

These models are created for the grain reception section, for the grain preliminary storage section, for the dryer as well as for the whole plant.

Two technological schemes were used in the models - the first scheme provides for the reception of only one grain flow simultaneously; the second one - for three grain flows simultaneously.

The assumption is that the grain delivery consists of separate portions which remain unchanged during the further processing, only their parameters varies.

The mass, temperature and volume weight of a grain portion are simulated with a normal probability distribution, but the moisture and cleanness of grain by Weibull's distribution. These parameters of grain portions simulated are recorded in its certificate as variables. The certificate has also to contain the codes of crop, variety and reproduction, as well as codes of its use and the owner.

The cleaning, sorting and drying time are calculated for each grain portion according to different empirical formulas.

The machinery safety is estimated by taking into consideration the elapsed time between the breakdown of the machines and by the time of repair , both of them are simulated with the normal probability distribution.

All simulation models, except the numerical outlet date, receive also histograms, e.g., the histogram of the number of ventilation bins, the histogram of the relationship between the dryer productivity and the grain daily thrashing etc.

The simulation models are most of all susceptible to the changes of the arrive rate and of the grain moisture as well as to the machinery safety.

All simulation models are created by making use of the programming and modelling system SITA/C, designed by the Institute of Mathematics and Informatics, at Latvia University.

Stochastic Simulation of Ligand-Receptor Interactions

Ursula Schweiger, Michael Berger and Mario Veitl

Institute of Medical Cybernetics and Artificial Intelligence
University of Vienna, A-1010 Vienna, Freyung 6/2

Institute of Biochemical Pharmacology
University of Vienna, A-1090 Vienna, Borschkegasse 8A

The interaction of radioligands with binding sites on biological and analytic material may involve a variety of different mechanisms, which can not be observed directly. Detailed kinetic studies show a summary of the underlying processes.
Previous studies with simulation experiments based on differential equations indicated, that a stochastic simulation approach could get more realistic models. Here we present the development of a stochastic simulation tool, which allows easy generation of different scenarios and a subsequent analysis of the obtained data.
We have chosen a population of 10000 fictitious binding sites, which can exist in up to four different stages. The transitions between the states are driven randomly, gauged by freely selected probability constraints. By varying these constraints in a systematic way a series of results was obtained and analyzed with classical (exponential, bi-exponential) fitting algorithms. The comparison of these simulation results with true experimental data and predicted values derived from classical diffential equations demonstrates the validity of the model.
The main advantages of our approach are the flexibility to model new scenarios and more realistic appearance of results including random noise.

Numerical Simulation for IR-Fiber Optic Sensor Development

M. Sengeis, B. Edl-Mizaikoff, R. Kellner
Vienna University of Technology, Institute for Analytical Chemistry
Getreidemarkt 9, A-1060 Vienna, Austria

W. Theiss, P. Grosse
1. Physikalisches Institut der Rheinländischen Westfälischen
Technischen Hochschule Aachen
Sommerfeldstraße Turm 28, D-xxxxx Aachen, Germany

Today the main application of ray-tracing is to be found in film- and entertainment industry engaging high performance computers to satisfy the computational efforts. In contrast, scientific applications require versatile and flexible software packages capable of wavelength dependent calculations for applications such as designing optical arrangements for spectroscopic experiments.

During this work a new software for numerical simulation of optical sensing schemes is being developed with the aim to provide a flexible tool for the optimization of the IR fiber optic sensor developed in our research group during the last years[1]. The most outstanding feature of the program is the possibility of wavelength dependent calculating with respect to the spectral behavior of the used materials of complex optical arrangements such as IR transparent optical fibers coupled to a FTIR spectrometer in three dimensions. IR spectra can be calculated from bulk substances or from layer-stacks of different materials. The optical properties of different media are calculated from the real and imaginary parts of their dielectric function (DF). Layer-stacks are treated as one single material after folding the DF of each layer into one resulting DF. The new software SPORT (by M. Sengeis) is based upon the software SPRAY (by W. Theiß) which has successfully been used for simulating IR-microscopic experiments[2,3].

The software SPORT is written in C++ and uses a command-line interface handling all input and output data in ASCII format. Hence, SPORT can be used on any computer that is equipped with a C++ compiler (PCs, Macintosh, workstations or supercomputers) providing a flexible simulation tool. The calculation procedure is based upon simulating optical components such as mirrors, stops, detectors, virtual screens, light-sources and bulky objects coated with layers of dielectric materials using a Monte-Carlo like wavelength-dependent simulation procedure and Fresnel's equations to calculate the amount of radiation absorbed, reflected or redirected. Various fiber geometries such as fibers with integrated microlenses, coiled fibers or tapered fibers can be calculated and optimized. The virtual screens do not effect the path of the radiation and can be used to visualize the intensity of radiation at any place in an optical setup. Due to the modular program structure a wide variety of objects can be simulated by combination of these objects.

It is of considerable interest for the development of fiber optical evanescent field sensors to be able to optimize the optical sensor configuration such as fiber/fiber interfaces with numerical simulation procedures in order to avoid time consuming experiments.

Hence, this work demonstrates that wavelength-dependent numerical simulation is a valuable tool for scientific research and commercial product development to verify experimentally obtained results and to reduce experimental efforts in IR fiber optic sensor design.

---

[1] R. Krska, R. Kellner, U. Schiessl, M. Tacke, A. Katzir, *Appl. Phys. Lett.* **63(14)**, 1868-1870 (1993)

[2] B. Mizaikoff, K. Taga, R. Kellner, W. Theiß, P. Grosse, *SPIE* **2089**, 164 (1993)

[3] B. Mizaikoff, K. Taga, R. Kellner, *Appl. Spectroscopy*, **47(9)** 1476 (1993)

AN ATTEMPT AT CONSTRUCTION OF A CONCEPTUAL MODEL OF SHIP HULL CONSTRUCTION MATERIALS PRODUCTION PREPARATION DESIGN

Vilim Simunic, Branimir Cizmulin
"3.MAJ" Brodogradevna industrija d.d., 51000 Rijeka, CROATIA

When presented on a high semantic level, large shipbuilding systems assume rather identical organizational structure. In such presentations, the shipbuilding systems structures have been featured as " connected concepts groups". These groups may be further divided into subgroups. Such systems oscillate in their development and entropy in accordance with new situations arising within a system. Essential differences in situations within particular systems occur, primarily, in relation to whether connected concept groups have been built of open or closed structural elements. Another essential problem consists in how to recognize where the limits of the groups, as concepts, commence. Should the clue be sought within reciprocal dependence of the groups?

Existence of subgroups as concepts may be unrecognizable or their existence may be taken as a "formal" one, in cases when particulars of these subgroups seem to be insufficiently recognizable. Hereinafter, complexity of recognition requirements in the ship hull structure materials production preparation design system has been dealt with.

The production preparation design system related to preparation of ship hull structural elements materials, as isolated connected concepts units, has been given the assignment of enabling the connected concepts groups to function in a process-like manner within the domain of the production preparation activities. Fundamental particulars of the subject system are a result of the following elements: a development of activities within a time span, a need for overlapping actions - procedures through which the same are being monitored, a requirement for the ship properties to be copied, and a requirement as to the dynamics of the group to be controlled. These premises/ requirements have been accompanied by both spiral and intermittent flow of production preparation and construction of a ship within an open system context. Spiral method of conducting the ship construction process has been generated by activities cycles - procedures. Frame close development cycles are, however, the missing chain links in an analysis of the system. Integrating concept for this connected concepts unit is based on an organization which learns from and acts within the informatic technologies.

Within this study, postulations as to existence of ship hull structure materials production preparation design system have been elaborated, as well as the structure of the system proper, with a presentation of the dynamism of connected concepts groups paralel action. Monitoring of work dynamism employs symbols contained within the activities cycles graphic presentation.

Analysis of some biodegradation processes using a mathematical model

Mircea Sularia

Institute for Informatics, 8-10 Maresal Averescu Avenue,
71316 Bucharest 1, Romania

In order to represent the bacterial growth, the substrate removal and the oxigen consumption kinetics in the biological oxidation of wastewaters, we consider the following equations :

$$\frac{dX}{dt} = -a \cdot \frac{dS}{dt} - k_d \cdot (X - X_{min})$$

$$\frac{dS}{dt} = -\mu_{max} \cdot \frac{S - S_{min}}{Ks_0 + \beta \cdot (S - S_{min})} \cdot (X - X_{min})$$

$$\frac{dO}{dt} = k_O \cdot \frac{dS}{dt} - k_e \cdot (X - X_{min})$$

where the basic variables are biomass concentration (**X**), substrate concentration (**S**) and disolved oxigen concentration (**O**) and the parameters are time (**t**), cell yield coefficient (**a**), cell decay coefficient ($\mathbf{k_d}$), minimal level of biomass concentration needed for starting the biological oxidation process ($\mathbf{X_{min}}$), substrate removal rate coefficient (μ_{max}), remanent substrate concentration ($\mathbf{S_{min}}$), constant **ß** having two values 0 or 1, constant $\mathbf{Ks_0}$ representing Michaelis-Menten constant for ß=1, S_{min}=0 and X_{min}=0, oxigen requirement per substrate utilized ($\mathbf{k_O}$) and oxigen consumption per biomass for endogeneous respiration ($\mathbf{k_e}$).

The above mentioned equations define two types of mathematical models, one type for **ß=1 (nondegenerated)** and another type for **ß=0 (degenerated)**. For ß=1, S_{min}=0 and X_{min}=0 one obtains also the well known Monod-Heukelekian-Eckenfelder & O'Connors differential model. In this paper we propose a new method for the evaluation of the kinetic coefficients **a**, $\mathbf{k_d}$, μ_{max}, $\mathbf{Ks_0}$, $\mathbf{k_O}$ and $\mathbf{k_e}$ together with values $\mathbf{X_{min}}$ and $\mathbf{S_{min}}$.

Many computer numerical simulation experiments have been realised. We conclude that the mathematical model considered here is in accordance with experimental data in batch reactors and the proposed parameter identification method is very useful in study of biological degradation of organic substrate and elaboration of kinetic coefficients of some processes (e.g. biodegradation processes of phenolic wastewaters or domestic wastewaters).

Acknowledgements. This work was supported by the Research Institute for Waste Water Treatment, Bucharest. The author wish to express its thanks to Mrs.dr.Eliza Leonte-Pena, Carmen Istrate and Camelia Cristea for experimental data delivered.

Cryosurface Frostdeposit Dynamic Model & Simulation

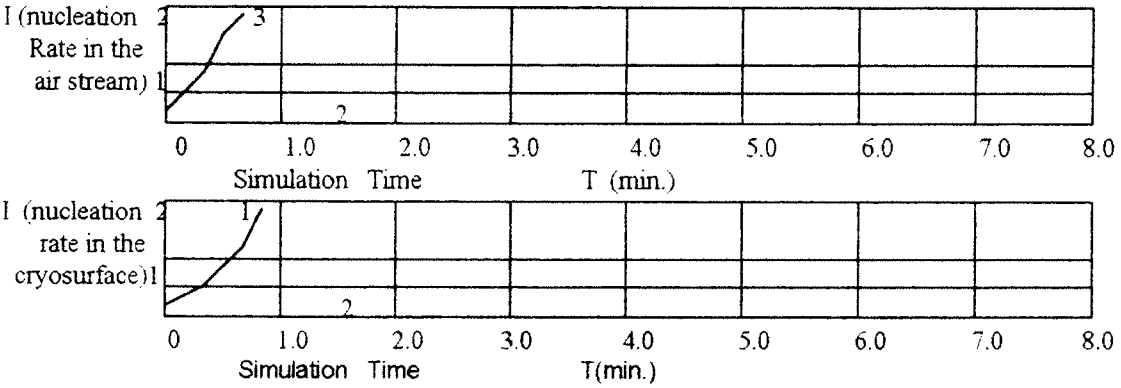
Sun Yuqing Shu Hongji Liu Huizhi
(Dalian Maritime University P.R. China)

The fundamental research on cryosurface frostdeposit has been carried out from long ago. And the main subjects have been: (1)the types of structure of the cryosurface frostdeposit; (2) the frost growth and deposit velocity during frost formation; (3) the decrease of heat transferred; (4) the decrease of air flow rate, etc. All these are important in frost formation. In 1980, the authors of this paper initiated the investigation of " The Effect of Cryosurface Coated with a Hydrophobic Film on Frost Formation" in the P.R.China. And after that, some similar papers have been published abroad. Nowadays, there are some physical and mathematic models established of " The Cryosurface Frostdeposit Dynamic Processes".The effectiveness has been proved both in theory and in practice to prevent the frostdeposit on a cryosurface coated with a high hydrophobic film. The models are presented in two aspects in this paper. They are: first, the frost is formed on the surface itself, that is the phenomenon on the initial stage of frost formation; second, particles being suspended in the air stream change into ice crystals, which will then deposit on the surface at random. These models refer to the dynamic process on an intermediate stage before the frost layer is formed on the surface. And according to the theory of critical supersaturation depending on cos θ for desublimation (where θ is the contact angle between the new phase and the substrate), nucleation and interface energy, etc., the frost on the substrate is formed in a dynamic process, in which, ice crystals grow or disappear, and the ice embryos in the air stream grow or disappear when they collide with the substrate. According to the models, computer simulation has been carried out to find out the relationships among the parameters. The results of this simulation are as follow: (1)The cryosurface covered with a hydrophobic film will prevent the frost formation on the substrate; (2)The ice embryos on the cryosurface grow or sublimate when the surrounding parameters are different; (3)Suspended particles in the air stream may play an important role in the frost formation; etc. The computing results show as the figure bellow, in the figure , I is the nucleation rate in the unit area (volume) per minutes, cure 1 is θ=110°, Cure 2 is θ=83°, Cure 3 is some big particles suspended in air. To summarize on the frostdeposit on the cryosurfaces, the following conclusion can be drawn:

(1) The initial stage of frostdeposit depends upon the local, different surface properties;

(2) The phenomenon of supercooled water on an intermediate stage of frostdeposit can be explained by the conventional and contemporary theory of heterogeneous nucleation;

(3) The interfering interface energies play an role.



Conclusion: if some conditions are satisfied, the cryosurface would not have frostdeposit. It is industrial available.

Computer Simulation Methods for Treating Instationary Excitations of Building Load Carrying Structures

Dr. Béla Szentiványi MSc. civil eng.
Hungarian Academy of Sciences, Research Group
in Technical Mechanics, Working Section
at Civil Engineering Mechanics Department
of TU Budapest
Mûegyetem rakpart 1-3, H-1111 Budapest, Hungary

Summary of the poster

Linear viscoelastic buildings with composite structures are investigated by new time-history method in case of horizontal earthquake ground motion excitation, which is an instationary generally Gaussian stochastic process. Ground water effects are excluded here from the scope regarding load carrying structure-foundation-soil interaction. Instead of Monte Carlo approach a new simulation method: the extended version of the realizations weighted with its probabilities is used. Here one realization is a stochastic solution in itself with fixed circular frequency of excitation, to which a probability value can be associated. Solution for one realization is performed on multi-degree-of-freedom model of the composite structure with the aid of modal matrix method and direct statistical evaluation of stochastic processes of response till second moments. Light elastoviscous material damping is assumed. The stochastic input here is an earthquake ground motion excitation family, in horizontal direction, which is assumed as sum of doubly stochastic sinusoid functions. Computer program was worked out for analysis of multi-storeyed buildings for mainframe computers. Numerical results for horizontal absolute and relative displacements of floor-levels show significant effects of horizontal stiffness of structure-subgrade contact and of viscous linear damping coefficient. Effect of increasing mass at the first level of storeys above the foundation proved to be advantageous. Own experiences on site of 5. 6. 1994 Csepel (Budapest) South-Middle earthquake on panel buildings (3 floors) negate opposite opinions for similar buildings.

Keywords: earthquake modelling, high-rise buildings, instationary ground soil motion (horizontal).

References

1. Szentiványi, B. (1989): Simulation of Multi-Degree of Freedom Model of Earthquake Response of High-Rise Buildings ESM'1989 Rome, Italy circulated lecture abstract, 2 p.
2. Vértes, Gy. (1985): Structural Dynamics, Elsevier Science, Amsterdam-Budapest
3. Szentiványi, B. (1987/90): Application of the Reliability Theory to Problems of Structure-Subgrade Interaction, PhD-Thesis defended at TU Gdañsk-Wrzeszcz, 1991.
4. Vértes, Gy., Szentiványi, B. (1993): Statistical Displacement Analysis of Composite Building Structures ... 10th ECEE, P4/106 Abstracts Vol. II. TU Wien, Austria, Aug. 1994.

Acknowledgement

T-OTKA (1994-96) Hungarian Research Fund ensured a fixed sum to support further research work.

System Dynamics Simulation for Regional Planning in Bavaria

NGUYEN XUAN THINH
Institute for Ecological and Regional Development, Weberplatz 1, D-01217 Dresden
RÜDIGER SCHULZE
Technical University of Dresden, Faculty Computer Science

In this paper we report on a system dynamics simulation model for regional planning in Bavaria which has been developed by Klatt, Kopf, and Kulla [3]. We have implemented this model in UNIX-MATLAB and carried out different experiments with them and compared the simulated data to actual. To support the experimentation with the model a user interface was also written in MATLAB. Moreover we have applied a method which has been elaborated by Vester [4] to classify the model variables.

Why system dynamics simulation for regional planning?

For regional planers it is important to have informations about the distribution of population, economy, infrastructure, land use, housing market, the changes that are occuring, and the proportions of such changes. Such information is, of course, important to governmental officials at various levels, as well as to legislators involved in the determination of the regional planning policy. For understanding the dynamics of change in a region we have to regard the region - with their population structure, economy structure, land use structure, etc. - as a coherent system, as a kind of whole that cannot be really understood from its separate components only.

By applying the system dynamics established by Jay W. Forrester [1,2] we can develop a simulation model that interconnects different sectors of a region and reflects their crosslinkage, interfacing in reality, and their feedback behaviour.

Structure of the Bavaria Model

Based on system dynamics Klatt, Kopf, and Kulla [3] developed a complex simulation model for planning in Bavaria and programed in DYNAMO (a computer language for simulating models).
The model consists of six model sectors: land use, population, economy, infrastructure, housing, and government (national budget). Objectives of the simulation model are

- studying the system behaviour of the Bavarian region,
- forecasting land requirements for agriculture, industry, manufacture, infrastructure and housing, and
- investigating the dynamics of population, economy, and housing market.

For the model construction the four sectors land use, population, economy, and infrastructure have been divided into further level variables. We have to integrate 19 level equations (e.g. the model has 19 level variables). Moreover the model has 38 table functions, and about 200 flow variables, auxiliary variables, external factors, and parameters. The level variables have been calculated by the following formula:

$$\text{New level} = \text{Old level} + \int_{t_0}^{t} (\text{Inflow rate} - \text{Outflow rate})\, dt$$

With this pattern of interactions, one can adapt the inflow and outflow rates by using of table fuctions - the non-linear interdependencies that result from the cybernetik interpretation - to reality at any time and thus to the changeswhich have taken place within the system as a whole.

Two simulation results shall be presented now.

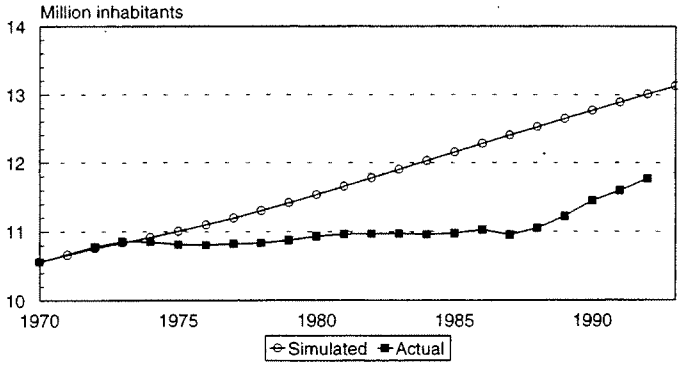


Figure 1: Population dynamics of Bavaria

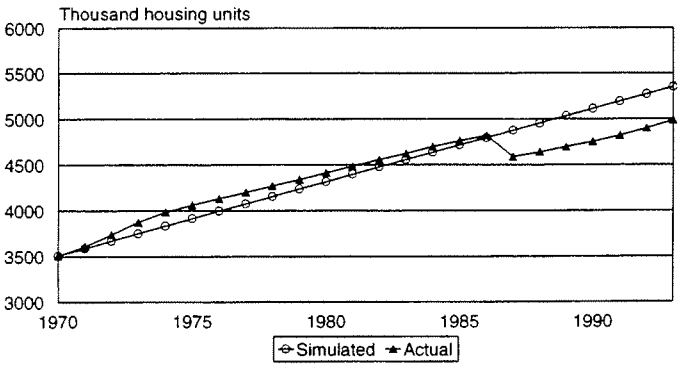


Figure 2: Housing stock of Bavaria

The source of actual data is [5].

Conclusions

As shown in figures 1 and 2 , the simulation model can make good forecast. Regional development requires a system-oriented treatment. With the help of this systems simulation we can investigate the dynamics of a region such as Bavaria under systemtheoretical and holistic aspects. Therefore systems simulation is a good method to assist in regional planning.

References:

[1] Forrester, Jay W., Industrial Dynamics, The MIT Press, Cambridge, 1961.
[2] Forrester, Jay W., Grundzüge einer Systemtheorie, Wiesbaden 1972.
[3] Klatt, Sigurd; Kopf, Jürgen; Kulla, Bernhard, Systemsimulation in der Raumplanung, Veröffentlichungen der Akademie für Raumforschung und Landesplanung, Hannover, 1974.
[4] Vester, Frederic, Ballungsgebiete in der Krise, dtv Sachbuch, 5. Auflage, 1994.
[5] Statistisches Bundesamt, Statistisches Jahrbuch 1990, 1991, 1992, 1993, 1994.

SIMULATION CONDENSATE TRANSPORT SYSTEMS

Verlan A.F.(Ukraine) Frid A.V.(USA)
Tikhonchuk S.T.(Ukraine)

One of the most important requirements to pipeline transportation of natural gas condensate is to provide a single-phase liquid state of carried mixture along the entire trunk lines. Failure to meet this requirement may cause pipes to be locked with gas bubbles and essentially decreases the efficiency of pipeline operations due to the additional power consumption on pump stations that maintain the flow of two-phase gas-liquid mixture. The solution of this problem can be achieved by pipeline operation control based on mathematical simulation of condensed gas flow.

Presented mathematical model and computation algorithm for computer simulation of condensate pipelines can be partitioned as follows.

- Model and algorithm for vapor-liquid balance computation based on Peng-Robinson equation. It permits to calculate mole fractions of mixture components in vapor and liquid phases for given mixture composition, temperatures and pressure.
- Model and algorithm for computation of temperature distribution along the condensate pipelines.
- Model and algorithm for hydraulic computation of condensate flow in a single-phase (gas or liquid) mode and two-phase (vapor-liquid) mode with gas bubbles in liquid or gas/liquid layers separated.

Program package based on the described model provides computation of condensate flow variables (pressure, flow rate, temperature and phase composition) along the pipelines for given initial conditions on the main field pump station. Calculations on each pipe section are completed through implicit iteration procedure with temperature and pressure as independent variables. To increase simulation speed express-algorithm for temperature calculations is also implemented. This algorithm utilizes Shuhov formula for temperature calculations and allows to separate hydraulic and thermal computations for each pipeline section. The overall gain in express simulation time is about 4-to-1 with some deterioration of accuracy.

The suggested condensate flow model and its program implementation enables condensate pipeline operation based on simulation results. Specific application examples are also considered.

"DYNAMIC SHIP STABILITY MODEL UNDER BORA AND JUGO WIND EFFECTS ON THE ADRIATIC SEA".

Živko Trošić
Meteorological and Hydrological Service
Marine Meteorological Center
Glagoljaška 11
58000 Split, Croatia
E-Mail: trosic@cirus.dhz.hr

ABSTRACT

System dynamic modeling of dynamic ship stability with the help of digital computers has special meaning because system dynamic method gives a real knowledge about behaving of dynamic ship stability in continued time area.

In this paper the author shows a model of ship dynamic behavior under wind effect during a navigation in the Adriatic Sea. Bora and jugo are characteristic winds in the Adriatic Sea that can be dangerous for ships' navigation, especially bora. Bora which blows over east shore of the Adriatic is a gusty wind, whose gusts can achieve a hurricane force in a short time.

Dynamic stability happens during heeling of the ship under effect of external forces, in other words heeling moments on the ship, wind and waves, or they are a result of ship loading by cargo or by water overrunning.

The model shows wind speed changes by level based on known legitimacy, wind moments as a result of course change, heeling amplitude (including capsizing angle) and rolling period, all in time period.
Dynamic ship stability modeling was executed by using the system dynamic method which defines:
1. mental, 2. verbal, 3. structural, 4. mathematical, and 5. dynamic ship stability model.

Modeling was made on digital computer using Powersim higher program language.

Parallel results of calculation weather criterion according to the recommended method of International Maritime Organization - when the moments of steady and gust winds do not change for the whole period of ship heeling and according to the method when these moments do change in time, are shown in this paper.

COMPUTER SIMULATION OF INDUSTRIAL DUST MIGRATION

G. Verechtchagina
Kiev National Shevchenko's University
Email: raspopov@compuchair.icyb.kiev.ua

Analytical and simulational model of wind dust transport of contaminations (radioactive fractions, chemical desperive contaminations etc.) in the air surfase layer is suggested. The model was developed and tested on bases of long-time forecast of radionuclide migration in the radioacontamination areas due to Chernobyl catastrophe. The above model is a part of program complex modelling different kind of ridionuclide migration in the air and in the aqueus medium.

A computer program has been developed illustrating on a display the dynamics of contamination spread. It can be used as a visual aid in teaching ecologists as well as in studying transport equations in the mathematical physics course.

Two-dimensional difference diagram of transport equation is analyzed. There are used the following parameters: the concentration of suspension in air, the intensity of distributed source of dust formation, the intensity of distributed dust absorbtion, the wind velocity field, the earth ability of dust formation, vegetation cover of the area and its relief, the presense of reservoirs, constructions as well as weather conditions. Statistical data collected during several years by hidrometeorological center were used in computer simulation. Time modelling step is equal to a calendar week. Wind rose determining averaged wind directions for every month is used for imitation modelling of wind direction which is calculated using random-number generator. The choice of element recalculation sequence in difference diagram is made considering this wind direction. The random-number generator is also used for imitation of wind velocity characteristics of one or another week of simulation.

This work was supported, in part, by the International Soros Sciense Education Program (ISSEP) through grant N GSU 041015.

SIMULATION OF TURBOMACHINE BLADING VIBRATIONS

Vorobev Io.S., Kanilo S.P., Shepel A.I., Korsunsky M.L.
Institute for Problems in Machinery of Ukrainian National Academy of Sciences
Pozharsky Str. 2/10, Kharkov, 310046, Ukraine

Vibrations present the main hazard to the steam and gas turbine blading.

The requirements to turbomachine blading mathematical models are complex and contradictory. They must represent the system vibrational properties as fully as possible at the same time contributing to the development of efficient methods of casculation. Therefore, it is advisable to adopt an hierarchic system of models: rod, two-dimensional and three-dimensional. The turbomachine blades have the most complex structural shape.

In a number of cases rod models of blades can be used for complex blading system, and particulary for systems with mistuning. The mathematical model of blade is constructed on the basis of classical or the refined engineering theory of pretwisted rods, wthich takes into account the inertial and deformational relationship of vibrations due to the initial pretwisting and non-symmetry of the cross-section, shear, turrning inertia, deplanation of the cross-section and the influence of rotation.

However, blades of complex geometry with flanges, roots and often internal cooling channels are used in the-state-of-the-art turbomachines. For such blades, it is necessary to use three-dimensional models in combinations with the method of finite elements. In this case, one can take into account the matherial non-uniformity due to temperature fields with high gradieents and non-uniformity of the object.

Extensive work has been carried out in comparing the results of systematic numerical analysis based on models of different levels with experimental investigations. As a result, we have built the boundaries of the domains of applicability of different models of turbomachine blades depending on the dimensionless parameters presenting their geometric forms.

The principle of system calculation by parts (diacoptic) can be used to reduce the complex problem dimensionality. At calculating the rod models the variational methods using the diacoptic idea turned out to be efficient. Typical partial problems with considerably lower order of the system of equations are selected. The results of solving partial problems are used for constructing basis functions meet all boundary conditions orthogonal to the partial problems operator and have the property of strong minimality. This makes it possible to bring down the order of equations solved at consequtive growth of the system complexity.

For three-dimensional finite element models the methods of condensation, multilevel superelement approach are widely used. Of special importance are the convenience of specifying the input data and the full visualization of results.

For turbomachine impellers possessing rotating symmetry, an approach is used which is based on reducing the order of resolving equation by considering one period of symmetry, viz. the sector of the disk or rotor with one blade and sections of bonds.

A comparison of simulation results and the data of the experimental studies shows the high efficiency of the technique developed.

XMosis: X-Environment for the Simulation System mosis

W. Zeller, G. Schuster, F. Breitenecker
TU Vienna

1. Introduction

XMosis was developed at the Technical University of Vienna, Department of Simulation Technique as Unix X-Window based frontend to mosis[1]. The *XMosis*-project has now reached a stable beta stadium.

2. Features

XMosis offers the user a full configurable interface to mosis, that means the user can define his own menu and dialogs. A global configuration keeps all necessary settings for *XMosis* to work properly and the user can add with his private configuration file his modifications and extensions.

Menus with userdefined Icons [6] to send commands to mosis or to popup userdefined dialogs can be setup by the user. Dialogs can be used to enter parameters and variables and to pass this information to mosis by the user without typing everytime a long command. All entered commands are stored so the user can scroll through this history and save this commands to a file for later reloading.

XMosis provides up to 32 windows to display graphics. In this window the user can zoom, resize and save the graphic in Postscript or Gif-format for later use.

Online help is also implemented using a WWW-client to browse through a HTML-manual.

XMosis communicates with mosis via PVM [4], because *XMosis* is launched from mosis and is therefore a child process to the main-mosis task. Due to this technology and the client-server-concept of X-Window it is not necessary to run *XMosis* and mosis on the same workstation.

3. Future Plans

To get the optimum in parallelizing models the next release of *XMosis* will include a task tracer to allow the user to watch the progress of his parallel-model and to optimize his algorithm.

To setup up a PVM-Cluster a cluster manager will be added also to allow dynamic adding and deleting of workstations.

References

[1] ATS and ARGESIM, Vienna, Austria, Mosis User's Guide, 1.02 edition, 1994
[2] X Toolkit Intrinsics Reference Manual, Third Edition for X11 Release 4 and Release 5, O'Reilly & Associate Inc. 1990, Volume 5, Third Edition ISBN 1-56592-007-4
[3] X Toolkit Intrinsics Programming Manual for Version 11, Adrian Nye and Tim O'Reilly, O'Reilly & Associate Inc 1992, Volume 4, ISBN 1-56592-003-1
[4] PVM3 User's Guide and Reference Manaual, Oak Ridge National Laboratory, 1994
[5] Athena Widget Set - C Language Interface,1989, Chris D. Peterson, MIT X Consortium
[6] XPM Manual, The X PixMap Format Version 3.4c, 1994, Arnaud Le Hors, Groupe Bull

[1]The Modular Simulation System, developed by G. Schuster at the TU Vienna see[1]

Simulation Analysis of the Job-shop Priority Rules: Under Three Different Processing Time Distributions and Exponential Job-Arrival Time.

Mujanah Ezat Agha [1] . & M. A. El-Baradie [2]
1 Simulation Consultant Office, P.O Box 6537 Aleppo/Syria
2 Dublin City University, Dublin 9 Ireland

ABSTRACT

This paper identifies the simulation of the job-shop priority rules. Priority rules are used to assign the next job to be processed on a specific machine. The paper has the following objective: To evaluate the effectiveness of the seven priority rule on the five well-used criteria as follows:
- The priority rules under study are:
1- First Come First Service (FCFS).
2- Last Come First Service (LCFS).
3- Shortest Processing Time (SPT).
4- Longest Processing Time (LPT).
5- Last Arrived At Shop First Service (FASFS).
6- Earliest Due Date (EDD).
7- Static Slack (StS).
- The criteria used are:
1- Minimize job lateness.
2- Minimize mean flow time.
3- Minimize idle time.
4- Minimize work-in-process.
5- Minimize completion jobs or production rate.
This evaluation will be under the dynamic conditions of the job arrival times which behaves the exponential pattern. The job processing times are exponential, uniform and normal patterns. The objective tested were under the steady state period of the observations collected using SIMAN simulation technique. Finally, the experimental results and plots show that the shortest processing time (SPT) rule was the dominant rule with respect to the most criteria tested under the most job-shop conditions. But, the last come first service rule (LCFS) tends to give the best rule than other with respect to the flow-time criterion when the normal processing time pattern is applied.

Dr.-Ing. Stefan Arndt
Business Information Technology and Operational Research,
Faculty of Economic Science,
Martin-Luther-University Halle-Wittenberg

Simulating and optimising stochastic inventory Models

An objective of inventory management is the provision of material in the right quantity at the right time. Inventory levels should reflect the level of demand and supply conditions. The level and duration of demand and delivery time are usually unpredictable stochastic variables. As strategy control variables, maximum and minimum inventory levels can be computed with optimising techniques. The objective function is a cost function comprising the costs of re-ordering, inventory holding and inventory shortages.

Both continuous and discrete-event models are appropriate procedures for simulating inventory systems. The continuous model gives effect to a control theory approach. Both approaches make use of familiar simulation languages, namely, SIMAN / ARENA and VisSim respectively.

Simple interfaces facilitate the interaction between the simulators and appropriate optimising programmes.

The slightly uneven surface of the objective function has hitherto revealed a clear global cost minimum. Modified for this application, the familiar GAUSS-SEIDEL method, a simple optimisation procedure, yielded reliable results. Genetic algorithms are also successful optimising techniques. The VisSim integrated optimisimg procedure are not always suitable for these models.

Analysing Traffic Flow by a Cellular Automaton

Heike Emmerich[1], Ernst Rank

Further expansion of individual road traffic will lead sooner or later to a traffic collapse at least in conurbations. It is hoped that this collapse can be avoided by intelligent traffic control systems. Those systems will heavily depend on simulation tools for a detailed prediction of critical traffic situations. In our contribution we will extend an approach which was first presented by Nagel and Schreckenberg in 1992 [1] and which uses a cellular automaton as a microscopic traffic flow model.

The poster introduces the concept of cellular automata for simulation of discrete dynamical systems. It explains the basic microscopic model being capable of reproducing so-called "stop-and-go" waves which characterize traffic flow on a freeway without further external impacts. These waves can be visualised by displaying the density of cars on a road versus time. The new component of our model [2] is that it is not only capable of simulating undisturbed traffic flow but also flow in case of additional road obstructions. As an example we consider a discretized road corresponding to about 30 km on a real road. In our model – due to some given hindrance – drivers are forced to reduce their velocity on parts of the segment to only half of its original value. Investigation of this model results in hysteresis loops which display two qualitatively different kinds of traffic behaviour.
So our approach makes a step towards new perspectives in dealing with problematic road situations. One future field of application is the support of variable driving instructors. The model can also be understood as a guidance for traffic planers dealing with routing. It predicts the critical density which in case of existing hindrances has to guide shut down of a road rather than the value obtained for undisturbed traffic so far.

References

[1] K. Nagel and M. Schreckenberg: *A celluar automaton model for freeway traffic,* J.Phys.I.France 2, 2221-2229 (1992)

[2] H. Emmerich and E. Rank: *Investigating Traffic Jams by Cellular Automata,* Physica A, No. 216, 435 (1995)

[1]address: University of Dortmund, Fakultaet Bauwesen, Lehrstuhl NMI, August-Schmidt-Strasse 8, D-44221 Dortmund
email:emm@busch.bauwesen.uni-dortmund.de
phone:++49 231 755 2093

STOCHASTIC MODELING OF MODERN COMPUTER SYSTEMS

L.P.Feldman, professor, I.I.Trub, PhD,
O.G.Shcherbakova, engineer Computer Science Department,
Donetsk Technical University, Ukraine

The paper is concerned with appropriate methods for computer system's performance evaluation, based on the queueing and graph theory. Some of problems are solved on the stage of designing, other arise by the operation, when non-satisfactive characteristics become clear. That is why preliminary estimations and recommendations are of great importance. In many cases these estimations are obtained by means of stochastic modeling and simulation.

It should be stressed that performance evaluation for parallel computing is on the stage of development, because the interest on mathematical tractability of modern architectures has almost always led to an oversimplified models.

As the title implies the paper describes new techniques for modeling some kinds of computer systems, to be exact, the Dataflow machines. The distinctive feature of all methods to be proposed is linear dependency of model size on the number of system's components. This fact makes possible their practical using.

Report consists of two parts. Simulation of two alternative approaches to Dataflow computations, such as data-driven (or eager) and demand-driven (or lazy) are considered in these parts respectively. In the first part method of performance analysis of static data-driven machine has been proposed. It is shown that taking into account assumption about fully dynamic assignments of PEs to knodes of data-flow graph and provided programming languages with one-assignment rule operation of data-driven machine is plausibly described by Markovian chain. Special queueing system for data-driven simulation is proposed.

Concept of stationary in such systems is introduced and condition of it's existence is proved, that is following: the average arity of program instructions must be exceeded by the average number of instruction results propagation. This condition wonderably remindes in appearance the basic condition $RO < 1$ for the simplest M/M/1. It is specially noted that Dataflow algorithms, taken for example, satisfy it. Model of data-driven computation by non-stationary conditions is also implemented as the system of ordinary non-linear differential equations. The results of digital experiments are presented and discussed. They were aimed to reveal influence of some input parameters on performance and, as conclusion, to give recommendations for their choice. The second part is devoted to the simulations of demand-driven computation. The main subject of research is the mixed directed demand / data flow graph. Special class of such graphs is defined. It is shown that this class is the subclass of skew-symmetric Berg's graphs and corresponds to the coarse-grain granularity of parallel algorithm. Two disciplines of demands service in the knodes are considered: switching to the firable state with demand of any consumer and of all consumers. Simulation techniques of moving demands/data are implemented. The optimal rule of demand choice from the queue is proposed, which is the further development and expansion of famous Hu's strategy. Theory of logical determinants is applied to the performance analysis. Effectiveness of methods is illustrated by some real examples, such as direct and inverse Gauss elimination and function's plot construction.

The first part of the report is fulfilled by I.Trub, the second part - by O.Shcherbakova and I.Trub. Prof. L.Feldman has carried out the general guidance of the investigations as a whole.

ON GRAMMAR OF QUACOL - LANGUAGE FOR PROCESS MODELLING

Author: Franjo Jović, Faculty of Electrical Engineering Osijek[*]

ABSTRACT

Modelling has been always a method for better product control. Many approaches have been used toward efficient model building. The primal controversy in model building is the relation of quantitative data expressed through process variables and qualitative product data expressed through product features.

The language for process modeling QUACOL (Qualitative Correlation Language) is described based on the extended creative possibilities of qualitative relations.

Grammar G(X, V, S,P) of the QUACOL is proposed with the set of grammar productions P in general iterative form.

$S \Rightarrow S_1$, $S_1 \Rightarrow$ If C then S, $S_1 \Rightarrow$ IfD then S_2 else S_3
$S_2 \Rightarrow R$, $S_3 \Rightarrow T$, R⇒SUCCESS, T⇒FAILURE, C⇒INPUT EXPRESSION,
D⇒INPUT EXPRESSION SIMILAR TO OUTPUT EXPRESSION

Derivation tree enabling parallel processing is given in the structural form:

if C then if D then R else T.

The qualitative relations are obtained from quantitative process data by means of a quantitative/qualitative transformation. Particular transformation is described enabling the application of QUACOL to the case of a thermoelectric power plant with "input" and "output" expressions

$$a + b\,v_1 + c\,v_2 / (d + e\,v_3) :: f\,v_4 + g\,v_5\,v_6 / v_7$$

where: v_1 - feedwater intake, v_2 - fuel intake, v_3 - fuel temperature,
v_4 - superheated steam flow, v_5 - temperature of exhaust gases
v_6 - burner air intake, v_7 - differential pressure of air for burners, and a, b, c, d, e, f, and g are model constants.

Results

The repetitive procedures in QUACOL enable for up to 97% of the process determination coefficient and less than 3000 ppm error in stohastic process prediction for traffic accidents in elderly people.

---

[*] Prof. dr. Franjo Jović, Institute of Applied Informatics
Faculty of Electrical Engineering,
University of Osijek
54000 Osijek, Istarska 3
Croatia
Telephone and telefax numben: +385 31 123 255
e-mail adress: jovic @culen. etfos.hr

Combining structural and behavioral views for modeling discrete event systems

A. Koukam and P. Gruer
Department of Computer Science
Institut Polytechnique de Sévenans
90010 Belfort Cedex - France.

Research in discrete event systems has resulted in the development of computer based tools that aid modelers in many parts of the modeling and simulation process. Modeling is intended for an abstract representation of a real world system, that describes the static structure and the dynamic characteristics of the system. Such description is then used for simulation in order to carry out the performance evaluation of the system. This work advocates that when modeling discrete event systems, the modeler must consider two views. The structural view describes the system components and the relationships between them. The behavioral view focuses on the state change of the system in response to the occurrence of discrete events. Several approaches have been proposed to model these two views. The object oriented or entity relationship approaches seem to be better to model the structural view, while finite state automata or Petri nets are widely used to model the behavioral one.
We feel that a better understanding of the link between the structural and behavioral models and its relationship to the modeling process is needed. We have developed some practical tips to fulfill that need. In particular, the problem encountered by modeler is the lack of guidelines to show the traceability between the two models. For example, the abstraction paradigms used in the structural model such as generalization and aggregation relationships have not their equivalent in the behavioral model based on the classical finite state automata or Petri-nets. To deal with these deficiencies, we provide a high level behavioral abstraction mechanisms that show how to specify explicitly the generalization and aggregation relationships in the behavioral model. This is accomplished by using the hierarchical modeling property of Statecharts; a graphical formalism that extends the finite state automata by concurrency and hierarchical constructions. This property augmented by the extension we propose, leads to a more structured and hierarchical behavior modeling and allows a system to be simulated at different level of abstraction.

Fuzzy Logic as Decision Strategy in Discrete Simulation

N. Kraus, F. Breitenecker; Department of Simulation Technique
Technical University of Vienna; Wiedner Hauptstr. 8-10; A-1040 Vienna

INTRODUCTION

The approach of Fuzzy Logic, developed by Zadeh in 1965, provides a tool for modelling most human thinking processes which are the decisive point in a lot of real world systems. This theory has helped to solve a lot of standard problems in the fields of control engineering, pattern recognition, operation research and so on in a better way than classical methodology could have. Simulation on the other hand gives us a possibility to visualise such a real world process on the computer for better understanding, planning, and experimentation. It is precisely the problem of vague rules and facts that often appears during the development of a discrete event simulation model. The main question is now how you can use the advantages of Fuzzy Logic and Approximate Reasoning in Discrete Event Simulation and also how it is possible to implement this theory so that the expanded Simulation Programme can be used for numerous new tasks.

THE DECISION CLASS OF FUZZY LOGIC IN DISCRETE EVENT SIMULATION

The uncertainties in a real world process can come from various sources. There is the possibility that parameters of a process can vary in a indefinable way during the operation. In most industrial processes human operators are integrated into the system and so the possibility to describe every decision in a precise way is no longer as possible as in a "purely technical" process. A third point is that, as the complexity of a system increases, the possibility to make precise statements decreases. All these uncertainties have to be considered during the planning phase of such an industrial process and so you have to implement them into your simulation model as well. The point I want to focus on is the possibility to replace complex crisp priority strategies which are a central point in most manufacturing plants by fuzzy decision rules. It is normal that in such a plant the crisp strategies implemented in the computer system are often overruled by the human operator, who is influenced by experience and feeling. By using a Discrete Event Simulation programme expanded by the decision class of Fuzzy Logic, it is possible to optimise such processes, implementing not only the crisp information but also the vague human decisions. The major purpose of using fuzzy decisions in simulation is to handle crisp and fuzzy information in the same model and to change between the different decision classes (stochastic, fuzzy, deterministic) in a short time.

IMPLEMENTATION

The main point of the fuzzy modelling technique is the idea of a linguistic variable. In order to use Fuzzy Operators, Modificators and Relations on such linguistic variables, you have to define a related Fuzzy Set for every variable. The concept for working with these linguistic variables is the concept of approximate reasoning. A linguistic variable is defined by its name, the universe of discourse, and the names of the labels of the variable. Each label is defined as a Fuzzy Set with a left *zero point*, one or two *max points*, a right *zero point,* and a type of Fuzzy Set. Approximate Reasoning means that you draw approximate conclusions from uncertain preconditions. In a statement using quantitative logic, a dependence between two numerically valued variables x and y is usually characterised by a table which may be expressed as a set of conditional statements such as "if x is 5 then y is 10". The same technique can be used for a qualitative approach, in which a sequence of instructions may contain fuzzy assignments and conditional statements. So, instead of using the IF-THEN-ELSE decision with crisp parameters, it is possible to call the Master Fuzzy Method and to pass the crisp parameters to it. The Master Fuzzy Method controls the other Fuzzy Methods such as Fuzzification, Inference, Defuzzification. A crisp return value can be given back to the primary Method for further operations. Using C or C++ for handling the necessary data and calculations, you can use the Fuzzy Methods with all discrete simulators, which provide you with a C-Interface.

CONCLUSION

A lot of different simulation programmes have been developed in the last years and each of them has its advantages and disadvantages compared to the others. But all of them have one thing in common, decisions can be programmed either deterministically or stochastically. These methods were developed for modelling mechanical systems and are highly suitable for modelling processes in which human thinking does not play a central role. But viewed in the perspective of new challenges in the future, the traditional techniques of system analysing are not well suited for dealing with systems directly influenced by people because they fail to deal with the reality of the fuzziness of human thinking and behaviour. So Fuzzy Logic provides a method to reduce as well as explain system complexities. So the benefits of Fuzzy Logic for Discrete Event Simulation can be summarised as follows: The amount of computer power that is needed can be reduced by decreasing the complexity of the model, the system is made more intelligible by integrating linguistic rules and human thinking is encoded more directly and exactly than by using standard mathematical techniques. Fuzzy Logic can expand the deterministic and stochastic decision classes used in simulation and so create the possibility of more efficient solving of new and specific questions in the fields of traffic, business, finance and others, in which human thinking plays a major role. The goal is to implement fuzzy decisions as a third decision strategy in between poor deterministic decisions and poor stochastic ones.

Knowledge acquisition for discrete event systems using machine learning

Dunja Mladenić[1], Ivan Bratko[1,2], Ray J. Paul[3], Marko Grobelnik[2]
[1]J. Stefan Institute, Dept. of Computer Sc., Jamova 39, Ljubljana, Slovenia
[2]Faculty of Electr. Eng. and Computer Sc., Tržaška 25, Ljubljana University
[3]Brunel University, Dept. of Computer Sc., Uxbridge, England
e-mail: Dunja.Mladenic@ijs.si

Knowledge acquisition for an understanding of discrete event simulation systems is a difficult task. Machine Learning has been investigated to help in the knowledge acquisition process. Our approach involves consultation with a domain expert, and the use of discrete event simulation models and machine learning as tools for the intelligent analysis of simulated systems. Current methods for the analysis and interpretation of such systems are restricted to statistical techniques that say much about the reliability of an output, but little about the output inter connectivity. The objective of our work is to improve the ability to interpret the model to the level of explanation that might loosely be described as "How the simulated system works". The new interpretation techniques are based on knowledge acquisition concerning the system using machine learning tools. "How the simulated system works" is important for a decision maker's understanding of the system in terms of relationships between the various parameters, the utilisation of resources and the location of (potential) bottlenecks. This sort of understanding is important when managing queuing problems in production systems, or when planning a new system. It should be emphasised that the goal of the presented work is not a prediction nor an optimisation. The main goal here is to find interpretations of the simulation output and discovering regularities, thus helping the user to develop an intuitive understanding of the domain.

Discrete event simulation produces example situations that can be used as input data for machine learning tools. In the presented research, three simple and commonly used discrete event simulators were interpreted using different machine learning tools. The attribute based learning systems RETIS and ASSISTANT were chosen as appropriate for learning in noisy, real-world domains. The interpretation obtained by these systems was intuitive but obviously expressed in a complicated way. To enable a more powerful knowledge representation, the Inductive Logic Programming (ILP) system MARKUS was used. In addition to compact knowledge representation, MARKUS also highlighted some attribute combinations that could be useful in the attribute based learning.

The knowledge gained from the machine learning tools was evaluated by the domain expert, who considered that this knowledge contributed to an understanding of the system, and how it might profitably be modified. This is important because the success of the present study should not be measured in terms of prediction accuracy, but in terms of the interpretative ability of the induced descriptions.

A methodological point of interest is the combination of ILP learning with attribute-based learning, and "chaining" of classification trees. (1) The ILP system MARKUS, could not really cope well with the noise in the learning data, but made important contribution as a generator for new attribute combinations. These were then beneficially applied in the subsequent attribute-based learning. (2) The knowledge provided by classification trees generated for the target problem (waste in steelworks) was analysed by domain expert and decided to be helpful but not enough detailed. Thus, depending on the already generated classification tree, a new target problem was formulated and the experiments repeated. This approach enables a more flexible analysis of the simulator from different, dynamically chosen, views and was considered by domain expert to be very beneficial for the simulated system understanding.

MATLAB Simulation of Idempotent Structures for Discrete Event Manufacturing Systems

O.C. Pastravanu*, Ayla Gurel**, F.L. Lewis***
* *Department of Automatic Control and Industrial Informatics, "Gh. Asachi" Technical University of Iasi,, Romania*
** *Department of Electrical and Electronic Engineering, The Eastern Mediterranean University,, Northern Cyprus,*
*** *Automation and Robotics Research Institute, The University of Texas at Arlington,, USA*

Research supported by NSF Grants MSS-9114009, IRI-9216545

Key Words: Max/plus algebra, flexible manufacturing, discrete event systems, MATLAB simulation

1. Introduction

In [1] a modern systems theory point of view is offered for the design of sequencing controllers for flexible manufacturing systems (FMS), whereby the controller is considered as separate from the workcell. In [2] the equations of the closed-loop system can be used to derive the max/plus representation for performance analysis. Starting from this matrix framework, MATLAB capabilities are extended to accommodate max/plus models.

2. Max/plus models for FMS

According to [2], for a given FMS (whose controller, if necessary, incorporates a conflict resolution mechanism) a max/plus model can be directly derived in terms of the task sequencing matrix, resource requirement matrix and duration of activities (processing times, transportation times, setup times etc.). System input u(k) represents the arrival times of input parts and system output y(k) represents the delivery times of the final products, where k denotes the iteration index. The maximum order of the delay operator is given by the multiple resources (pallets, buffers slots etc).

3. MATLAB functions for idempotent calculus in the semiring ($\mathbf{R}\cup\{-\infty\}$,max,+)

The semiring ($\mathbf{R}\cup\{-\infty\}$,max,+) involves non-standard operations which are not available in MATLAB. However the existence of the built-in function *Inf* (which returns the IEEE arithmetic representation for positive infinity) allows developing new routines for matrix addition and multiplication. These new matrix functions are further exploited to approach the complex max/plus models associated to various FMS structures.

4. FMS simulation and performance evaluation

The main objective of the simulator is to create a relevant picture of the workcell dynamics, including both transient and periodical steady state. Thus the analysis of the role played by the arrival times of the input parts becomes a straightforward task. Moreover, comprehensive tests can be performed to underscore the influence of the duration of various activities on the FMS performance. Changes in processing sequences and/or resource availability are simple to handle. A graphical interface based on MATLAB plotting functions is provided. Well designed simulation experiments are extremely useful for the optimization of the machine utilization.

5. Illustrative example

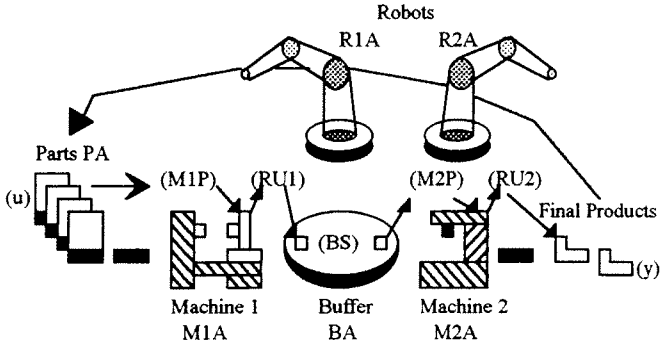
Consider the workcell in fig.1 [3], where both machines are automatically loaded; four pallets are available and the buffer has two slots. Fig. 2 depicts the workcell dynamics, in the case of a non-uniform sequence of arrival times.

6. Conclusion

The paper aims to enlarge the application area of MATLAB, by developing new modules devoted to the simulation of FMS described by max/plus models.

7. References

1. Lewis,F.L., O.C.Pastravanu and H.H.Huang (1993). Controller design and conflict resolution for discrete event manufacturing systems. *Proc. of the 32-nd Control and Decision Conf.*, 3288-3293.
2. Pastravanu,O.C., Ayla,Gürel, F.L.Lewis and H.H.Huang (1994). Rule-based controller design algorithm for discrete event manufacturing systems. *Proc. American Control Conference.*, 299-305.
3. Lewis,F.L., H.H.Huang, O.C.Pastravanu and Ayla Gürel (1995). Control system design for flexible manufacturing systems. In A. Raouf and M.Ben Daya (Eds.) *Flexible Manufacturing Systems: Recent Developments*, Elsevier.



	M1A	M2A	R1A	R2A	BA	PA
Processing time [min]	10	11	5	4	0	0
Setup time [min]	3	3	2	2	1	3

Fig.1 Sketch of the workcell

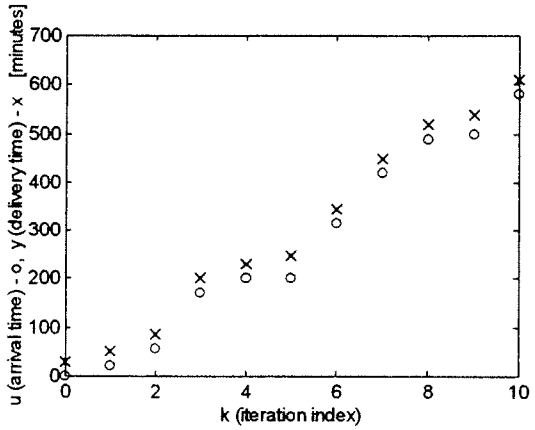


Fig.2 Graphical plots of input and output

EXPERT DECISION SUPPORT BASED ON THE GENERALIZED NETWORK MODEL : AN APPLICATION TO TRANSPORTATION PLANNING

Spyros Tzafestas*, T. Pimenides**, and G. Kapsiotis

* Intelligent Robotics and Control Unit
National Technical University of Athens
Zographou 15773, Athens, Greece

** Systems and Control Division
Department of Electrical Engineering
University of Patras, Patras 26500, Greece

ABSTRACT : An expert (knowledge-based) decision support system (KBDSS) is presented which is then specialized to the Generalized Network (GN) domain. The resulting KBDSS developed by the authors and called GENETEXP is actually useful for decision making in all real problems that can be put in the GN form and need additional higher level expertise. Practical results in the Optimized Transportation planning area have been obtained that illustrate the functioning and usufulness of the KBDSS.

THE GENETEXP TOOL : GENETEXP is a tool for analysing GN problems within a generalized decision making environment. With this package, any problem that has the GN mathematical formulation can be modeled and analyzed. Actually, the problems to be treated with GENETEXP cover a wide range of applications since it can deal with boundary condition constraints of inequality type, i.e. the nonzero supply for a given node can be specified either as a resource to be compulsorily conveyed through the network, or as the available quantity of the resource considered. In the same way, the value of a demand can be considered either as a specific requirement to be met, or as a minimum of the required quantity.

GENETEXP is the outcome of the integration of GENET OPTIMIZER (a previous pure numeric GN-DSS package developed by the authors) with an expert system shell developed at NTUA and used extensively for Engine Fault Diagnosis (ENGEXP) and Medical Therapeutic Treatment (BIOEXP). The package was developed in PASCAL under DOS on an IBM-PC compatible with sufficient memory to support the model's data and knowledge structure. A number of experiential rules have been accomodated to the knowledge-base for heuristic (expert-based) post optimization analysis. Work is under development by the authors to incorporate in the knowledge-base subsystem fully heuristic models accompanied by user/decision maker preferential rules. This is done in cooperation with a European-level supply-distribution chain. The system will then be capable of evaluating the solutions and using the one that satisfies not only numerical but also experiential preferential requirements.

Simulation in the planning of Construction Operations

Lic.Eng. Per Sagert and M.Sc. Ulf Månsson
Chalmers University of Technology
S-412 96 Göteborg, Sweden

Construction Operations have characteristics that make them suitable for simulation. For example they are resource intensive and the resource flow between operations often leads to a queuing behavior. Due to the uniqueness of a facility, construction tends to be like prototyping. When a project is completed, the production is ended and the prototype is hopefully built according to the specifications. Although facilities themselves are unique, the construction methods used during the project are often repetitive and cyclic. Construction operations are regularly vulnerable to external factors, e.g. weather conditions, that are difficult to predict and control.

All these conditions make simulation very applicable as a planning tool for construction operations. Simulation's ability to test and evaluate systems and plans before the actual implementation ought to attract construction engineers and managers. Despite these factors, simulation has not yet gained the expected success in the construction industry.

There are of course several reasons for the limited use of simulation in construction planning: the flow based structuring approach in most simulation tools, developed for simulation of manufacturing processes, is not suitable for modeling of construction operations. There is a also lack of useful input data, and the limited knowledge about modeling, statistics and how to interpret simulation results among construction professionals also effect the usage.

Models are often developed in the construction planning stage and later transferred to the construction site. The models must therefore be easy to communicate to members of the construction team and the modeling approach should be able to capture the cyclic flow of resources in construction processes.

To make the modeling quick, the tool ought to be integrated with other planning software and use available equipment data, previous captured production data and saved models. The interface should be based on a graphical modeling approach, in which the models can be visualized both during design and run.

Based on the requirements above, a prototype tool for discrete-event simulation of construction operations, called TOWERS, have been developed. Some of the suggested requirements, have been successfully exemplified and tested with TOWERS.

TOWERS is capable of modeling complex resource flows and contributes new capabilities in visual model building. A simulation model is conceptualized and built by drawing an activity cycle diagram. The activity cycle diagrams can capture the cyclic flow of resources at the construction site. The diagram is formed by simple symbols, but can also contain pictures and digital video. This enhances the visualization capability and improves the understanding for the model among occasional users. The usage of templates provides quick and easy model building combined with the usage of captured data.

It is too early to draw any far reaching conclusions, but initial field tests indicate promising results for the visual modeling and structuring approach suggested. The simulation technique can, in the future, be naturally integrated with other computer based tools for planning and controlling construction projects.

Discrete Event Simulation of a Real Assembly System A Tool to Double Throughput

Markus Vorderwinkler[1], Michael Schleicher[2], Johannes Preissl[1], Stefan Graebe[2]

[1] Vienna University of Technology, Institute of Flexible Automation
Gusshausstrasse 25-29 / E361, A-1040 Vienna, Austria
Tel: (+43-1) 504 18 41, Fax: (+43-1) 505 59 83
E-mail: mv@flexaut.tuwien.ac.at

[2] PROFACTOR-Produktionsforschung GmbH
Wehrgrabengasse 5, A-4400 Steyr, Austria
Tel.: (+43-7252) 884-220, Fax: (+43-7252) 884-244
E-mail: Michael.Schleicher@pr-steyr.ac.at

This poster describes the use of event-simulation for optimizing highly configurable manufacturing facilities for large-scale production. Within a research project carried out at the Institute of Flexible Automation, TU Vienna and Profactor, Steyr, event simulation was successfully used to demonstrate that the substantial benefits of configurable automation can also be achieved in areas traditionally relying on rigid production lines.

Problem Formulation

Traditional production lines frequently rely on a rigid transportation system with constant processing times between a sequence of simple pick and place work stations. In such an assembly line, disturbances will inevitably prevent the achievement of the planned production volume and result in inefficient operation. Indeed, it is known that the overall availability of a rigid production line equals the product of the availability of the individual stations. In a sequence of 20 stations with 97% availability each, for example, the overall availability sinks to as low as 50%.

Strategies aimed at improving the productivity, must observe the following typical constraints: highly variable disturbance frequencies, large variations in repair times, a wide spread in product variation and quantities, high product volume and large variations in processing times.

To investigate these phenomena, the influence of various disturbances on both loosely coupled and rigid production lines with short repair times were simulated and compared. The long-term objective of the study is to highlight the importance of flexibility in large scale productions and to establish generic strategies for increasing productivity.

Solution Strategies

To obtain quantifiable results, potential manufacturing facility layouts were described by parametric models. The following optimization was based on the sum of invested and running costs divided by productivity as measured by units manufactured per hour. An iterative algorithm was then used to estimate the optimal processing times, buffer capacities, supply rates of assembly parts, number of pallets and workforce size.

Further effects modeled in the optimization included various product distributions, as well as different disturbance frequencies and durations. Finally, the economic feasibility of „chaotic manufacturing" was evaluated.

Results

The above optimization study shows convincingly that the flexible manufacturing paradigm holds substantial benefits for large scale productions subject to disturbances with short repair times. In particular, by introducing and optimizing buffer capacities and variable processing times it was possible to double productivity and drastically down-size the working force in spite of uncertain production and failure data. The cost of introducing these upgrades was evaluated to be an investment increase of 15% and running cost increase of 20%. It was also shown that a rigid production line with the same nominal throughput (no failures assumed) would require similar investment and running costs.

Thus, there were only minor additional costs associated by introducing the advantageous flexible automation.

❑

SIMULATION OF MANUFACTURING SYSTEMS

Simeon Simeonov

Dynamic characteristics of Manufacturing System are represented by using a computer simulation model. This model allows to find out lacks of simulated system before his realization in nature. A model shows the behavior of the system and impact of changes of parameters and conditions. FACTOR/AIM (Pritsker-Corporation) is used for evaluating the simulation model of FMS (Flexible Manufacturing System). This FMS consist with six technological workstations which provide milling and drilling operations. The simulation model is utilized in process of FMS projection. Therefore, numbers of workstation and other devices are an object of simulation experiments. Flow of technological pallets ensures the automatic transporter with linear transportation path. The transporter takes workpieces from the fixture workstation and carries to technological workstations under process plans. When a technological workstations is busy, a workpieces is put on the linear storage area. The speed of transporter and number of cells of the linear storage area are an object of simulation experiments, too. Input/Output storage situated before fixturing workstation creates a boundary - line between FMS and other production facilities. Two workers are used for fixturing of workpieces in fixturing workstation. For representing of production requirements of FMS such as technological workstation, storage areas, workers etc. AIM has prepared four modelling components: Resources, Resources group, Pools and Materials. The FMS model used both resources and resource groups. To model linear storage area and Input/Output storage are selected two approaches. Input/Output storage was modeled as WIP (Work-In-Process) because this storage has small capacity and its behavior is like as a conveyor. For quick modelling of the linear storage area is used WIP again. But, this approach for modelling of large storage area with transporter is not precise. Therefore, the following approach is used: each storage location (cell) is modelled as a resource. The transport system has a control point at each location. For traveling between storage location are defined appropriate segments. The resources are all in a resource group. This group and control points are tied together through AIM Control point Group structure. For releasing one or more orders into the FMS are used several DEMANDs. The demand limits the number of orders released and time between releases. Time between releases (Interarrival Time) is generated by using Normal distribution. FMS produces 22 different types of workpieces (parts). Technological process, witch routes flow of parts through FMS is described using process plans and jobsteps. Each operation is defined by using one or more jobsteps. The FMS model includes a transporter system which is setting with transporter vehicle, transporter segments and transporter control points. Each technological workstation, fixturing workstation and each storage area cell has own transporter control point. The FMS simulation model is used to perform several simulation experiments. The experiments carry out to determine characteristics and the operation of FMS.

Dipl. Eng. Simeon Simeonov, PhD. Technical University of Brno Technicka 2, 61669 Brno Czech Republic tel. (425) 41142446 fax. (425) 41211994 E-Mail: simeon@cspuni12.bitnet

THE VALUATION OF A ROBUSTNESS OF A MULTIPROCESSOR COMPUTING SYSTEM

N.R.Andronaty A.A.Golovnea Yu.V.Rogozhin

Academy of Sciences of Moldova
Stefan cel Mare av., 1,
277612, Kishinau, Moldova
E-mail: prezidium@academy.moldova.su

We consider the model of the multiprocessor computing system on the base of transputers (CT) which permits to resolve the question of valuation of a structural robustness (viability, survivability) of CT.

For research we chose a class of managing computing systems. This class is simulated by a communication network. As a main component of CT for this class its fragment FCT is considered. FCT is a microstructure connecting the output into external environment with data processing unit or one point of entering in the microstructure with other point of entering. The difference FCT from other components of CT is insignificant.

At simulation FCT by a communication network, the transputer is considered as a complex element, consisting from four simple communication elements, possessing one common point, in which concentrates a computing element (data processing unit). An idea of a rank is suggested. The rank is a quantitative characteristic, determining the minimum quantity of faults of simple elements, causing to fault of a communication network between first and second points of entering.

The quantitative expression for calculation of the structural robustness of FCT (microstructure) is suggested:

$$J_{F_k}(N, S, f^{in1}, f^{in2}, R, P^{sr}) = (r + P^{sr}),$$

where F_k - is the chosen FCT, N - is a set of elements of the FCT, S - is a structure of FCT, f^{in1} - is the first point of entering into the FCT, f^{in2} - is the second point of entering into the FCT, R - is a set of the ranks of branches (sections of a way), P^{sr} - is a structural probability of keeping the rank of FCT, r - is the rank of FCT.

The result of calculation under this formula is a real number, the entire part of which is the rank of a microstructure, the fractional part is the probability of keeping this rank at the first single fault. The probability of keeping a rank for a particular microstructure is determined as a fault probability sum of some elements of this microstructure. These elements are selected so that each element fault in of separateness does not lower the rank of the microstruicture.

By use of the given expression, the structural robustness of any structure of CT of a type of FCT can be evaluated, if its structure can be represented as a set of different level microstructures, and if for each of microstructure a way of calculation of this expression is known. The examples of some typical microstructures and ways of account for them of the structural robustness are submitted.

Optimal state save insertion for parallel simulation based on Time Warp approach

Francesco Quaglia e Bruno Ciciani
Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
e.mail ciciani@dis.uniroma1.it
tel. +39 6 49918325
fax +39 6 85300849

The Time Warp is a method of asynchronous parallel simulation with an optimistic approach and is collocated among methods that explain the intrinsic parallelism of real systems through an opportune decomposition of relative models in subsystems which are exclusively interagent by means of messages: this is referred to as object-oriented simulation. Any object can interact at any time with any other object each one of which has three queues: one queue, in which incoming messages from other objects are inserted, is called the input-queue; another queue, in which messages sent to other objects are memorised, is called the output-queue; the third queue, in which a certain number of evolution states passed through by the object during simulation is memorised, is called the state-queue. None of these objects undergoes constraints in order to carry out the computation relative to a new message but can continue to go ahead, unless the input-queue is empty. Therefore, with respect to non optimistic methods, the approach to the problem of the consistency of the simulation changes completely. Rather than synchronisation mechanisms, if a situation of incorrectness occurs, a rollback mechanism is predicted which takes the simulation back to a correct state relative to a previous virtual time from which the computation can start again. It is necessary to memorise a fraction of the states which have been passed through during the simulation just in order to enable the handling of this eventual rollback procedure.

The saving of the state of the objects is a procedure which costs in terms of execution time and therefore saving the state of the object less frequently results in economy of the time spent on this procedure. On the other hand, however, having only a fraction of the states passed through during simulation available increases, on the occurrence of rollback, the probability of restoring a state, which sends the simulation further back than is really necessary for the elaboration of the message which generated the error. This makes the object in rollback go through a series of elaborations which have already previously been carried out correctly. This last phase of rollback is referred to as coasting-forward and its mean cost is proportional to the distance, measured in terms of the number of computations, that separates two consecutive saved states.

Our goal is therefore to determine the optimal distance between two consecutive saved states, for which the increasing cost of the coasting-forward is balanced by the decreasing cost obtained by avoiding saving all the states passed through the simulation.

The Algebraic Programming of Parallel Symbolic Computation for Modeling and Simulation

A.A. Letichevsky, J.V. Kapitonova, V.A. Grebnev and A.E. Doroshenko
Glushkov Institute of Cybernetics,
National Academy of Sciences of Ukraine
Glushkov prosp., 40, Kiev 252187, Ukraine
E-mail: dor@d105.icyb.kiev.ua

Increasing size and complexity of computer simulation systems especially in their parallel implementations evokes a need for their high-level, declarative description and prototyping. Symbolic computation gives good opportunities for research and making knowledge-based decisions. Algebraic programming is a kind of symbolic rewriting based computation that integrates four main paradigms of programming: procedural, functional, algebraic and logical. Its main distinction consists in considering rewriting activity together with a strategy of rewriting usually defined in procedural form. It gives advantages of flexibility, high-level rapid prototyping, low cost simulation and evolutionary development of efficient parallel software from declarative to procedural programs.

The poster displays an approach to symbolic simulation developed in our algebraic programming system APS framework [2] and basic facilities of the APLAN language [3] that provide creating, completioning and interpreting algebraic modules. Simulation algorithms follows event-driven approach in running modules being simulated. Our methodology is demonstrated with an example of parallel program development and simulation for the known computer algebra problem of Gröbner basis construction [1]. We have developed rewriting style parallelization of Buchberger's algorithm. It gives opportunities to construct highly adaptable parallel version of the algorithm that was realised as a multimodule algebraic program with message passing. Experiments on benchmarks for Gröbner basis known from literature show good efficiency of simulated parallelism that can be achieved from our declarative presentation of the algorithm. Current version of the APS system is implemented on IBM PC 486 computer and the experiments on the local net of PCs is now under preparation.

References

[1] B. Buchberger, History and Basic Features of the Critical-Pair/Completion Procedure, *Journal of Symbolic Coputation*, Volume 3 No.1-2, pp.3-38, 1987.

[2] A. A. Letichevsky and J. V. Kapitonova, Algebraic programming in APS system, *Proc. of ISSAC'90*, pp. 68-75, ACM Press, 1990.

[3] A.A.Letichevsky, J.V.Kapitonova, S.V.Konozenko, Computations in APS, *Theoretical Computer Science*, **119**, 1993, pp.145-171.

DISTRIBUTED SIMULATION OF HIGH LEVEL TIME PETRI NETS

Juan Carlos Dueñas Gonzalo León

ETSI Telecomunicación, Universidad Politécnica de Madrid,
Ciudad Universitaria s/n, E-28040 Madrid, Spain
{jcduenas,gleon}@dit.upm.es

High Level Time Petri Nets (HLTPN) are a formalism commonly used for systems specification and development, because they represent both functional and time behaviour. Distributed simulation, where a net is divided into a number of subnets which are executed on different processors is a mean to cope with complexity of large designs, increases the modularization in the specification/design process, can make use of remote resources in the simulation, and makes its animation geographically distributed. We have applied this technique to HLTPN.

The simulation scenario is composed by submodels with shared places, the simulator or logical processor for each submodel, the physical processors and the communication system. **Distributed simulation of HLPTN is event-driven, pessimistic and asynchronous.**

This approach gives maximum freedom to compose subnets in order to create larger models, being the only restriction to join them by places. As a consequence of that, there are some restrictions to the concurrency degree of the simulation, because the marking of a shared place can be modified by the sharing subnets at unpredictable times. The general condition to be met to ensure time correctness is that a transition t can fire at time τ only if no other subnet can insert a token with a timestamp lower or equal to τ in any of its input places.

To ensure this general principle, the algorithm uses several protocols:

- **for transition firing.** To ensure the coherence between local copies of shared places, to fire transitions in an atomic way and to solve effective conflicts found.
- **time management.** To explore the time of next events in each subnet to perform firings in a safe way.
- **deadlock detection.** To detect and break communication deadlocks by chosing the time of the lowest event in the whole system.

A toolset for distributed simulation/animation of HTLPN models has been built in the ESPRIT IPTES project. It has been used for distributed animation of SA/RT models on top of the HLTPN executor.

Parallel Simulation of Large Office Workflow Models

Alois Ferscha
Institut für Angewandte Informatik
Universität Wien
Lenaugasse 2/8, A-1080 Vienna, AUSTRIA
Email: ferscha@ani.univie.ac.at

Abstract

Modeling and analysis of business organizations is required for the purpose of redesigning an enterprise's business process to make the organization more efficient (*Business Process Reengineering*), as well as for the purpose of establishing advanced coordination technology in the organization, i.e. provide automated support for the management of dependencies among the interacting agents (humans, machines) responsible for executing a business process by using e.g. distributed, networked environments. We consider a (business) process to be a set of partially ordered steps to reach a (business) goal. Any component of a process is called a process element, which is an atomic action with no internal substructures. An agent is an actor (human or machine) who performs one or more process elements. A coherent set of process elements to be assigned to an agent as a unit of functional responsibility is called a role, and the a product created or modified by the enactment of a processes element is referred to as "work". A process model now is an abstract description of an actual or proposed process constituted by a set of selected process elements. Typically, these process steps either produce some work, or coordinate the dependencies with other agents involved in the same or a related process. Due to the migration of work from agent to agent we refer to the dynamics of the execution of process models as "workflow models". A workflow model must appropriately describe the input and output relation of work for every process element, and above that, the assignment of process elements to the responsibility of agents. The latter necessitates the preservation of causal (flow) relationships defined in processes, but also the work scheduling and management strategy applied by an agent to execute the assigned process elements. This is essential especially for agents acting in several roles (involved in more than one process).

An important aspect of a real system of business processes is its *temporal behavior*. For a model to be adequate in this respect, it must reflect delays and durations of process element executions. As a consequence, the modeling formalism has to provide sufficient expressive power to characterize the time dynamics of the system. We have developed an abstracted framework of business process systems in the domain of Generalized Stochastic Petri Nets (GSPNs), and have shown how those systems are modeled and analyzed *quantitatively* and *qualitatively* using GSPNs [1]. Preliminarily we have choosen discrete event simulation as the means for the quantitative evaluation of GSPN workflow models, but encountered that with the increasing complexity of business process systems, traditional simulation techniques become practically intractable.

In this work, we present the use of parallel simulation methods to accelerate the workflow model evaluation. Our approach is to automatically decompose a timed Petri net model based on the structural properties obtained from a preanalysis of the model. Submodels are executed by *logical processes* allocated to different nodes on a multiprocessor [2], such that the overall elapsed CPU time can be reduced considerably. We present GSPN model partitioning heuristics as applied to models coming from a real business organization, and the respective parallel execution performance on the CM-5 and a cluster of RS6000 workstations.

References

[1] A. Ferscha. Qualitative and Quantitative Analysis of Business Workflows using Generalized Stochastic Petri Nets. In G. Chroust and A. Benczur, editors, *Connectivity 94: Workflow Management*, pages 222 - 234, Vienna, 1994. Oldenbourg.

[2] A. Ferscha. Parallel and Distributed Simulation of Discrete Event Systems. In A. Y. Zomaya, editor, *Parallel and Distributed Computing Handbook*. McGraw-Hill, 1995.

Sensitivity Analysis of Simulation Time in Time Warp Parallel Simulation

V.Cortellessa , G.Iazeolla

University of Rome at Tor Vergata
Laboratory of Computer Science
Via della Ricerca Scientifica
00133 R O M A (Italy)

iazeolla@info.utovrm.it

Abstract

In the last few years, much efforts have been devoted to obtain efficient implementations of parallel discrete event simulation.

Among the different approaches, time warp is a method based on the rollback mechanism, to perform simulation on a distributed environment. This paradigm states a strategy to synchronize the logical processes involved in the simulation.

Parallel simulation consists of two basic components: the normal forward execution time , and the rollback mechanism time-overhead. It is convenient to parallelize a simulation until the latter component is not predominant with respect to the former one.

The goal of this work is the sensitivity analysis of simulation time versus variables as number of processors, checkpoint interval, rollback rate and lenght.

To this purpose the time warp mechanism is first analyzed by use of an execution graph, whose nodes describe various procedures the mechanism executes to deal with event sequencing and parallelization.

Total simulation time is the by-product of the mean time spent in the execution graph and the average number of events processed by each processor, given a certain simulation length.

For each node in the graph, the time necessary to execute its operation is introduced. Such time depends on the execution environment and the time warp kernel. An expression is also introduced for various branching probabilities among nodes in the graph. A further expression is obtained to evaluate the number of events processed by each processor, for the forward execution and for the rollback overhead.

On the basis of quantities above, the complete expression is obtained for the total simulation time as a function of number of processors, checkpoint interval, rollback rate and length.

Two out of four variables (i.e. number of processors and checkpoint interval) are user-controllable to tune the execution for optimal speedup, while the remaining two (rollback rate and length) are partly model-dependent and partly dependent on the processor number, and strongly affect the simulation overhead.

Various curves are introduced that illustrate the effect of such variables on time warp parallel simulation time.

Parallel simulation of charge transfer in semiconductor devices.

M. Jadzhak

Two-dimensional mathematical model of semiconductor devices represented by two continuity equations and one Poisson equation is considered. Having as the goal determination of concentrating fields of charge carrier and the field of electric potential we use the following new technique: approximation of output system of nonlinear differential equations is realized by the Galorkin form of finite elements method, and the obtained large-dimension system of nonlinear algebraic equations is linearized further by modificated Newton-Rafson method.

Basing on the technique proposed and on the employment of high-level language means for the introducing of parallelism a parallel algorithm for numerical modeling of charge transfer in semiconductor devices is constructed by the following way:

- the sections having explicit inner parallelism are discovered;
- the algorithmic complexity analysis and parallelism degree estimations for every from the sections discovered and for every possible parallel form are effected;
- using the primitives "fork", "join" and program loops of the types AUTON (autonomous), SIM (simultaneous), SYNCH (synchronous) parallel implementation of these sections is prescribed;
- the implementation of program loops is prescribed in the pipeline manner by using type of parallelism PIPE.

The results concerning adequacy of this model, semiconductor devices and parallel algorithms constructed are proved.

Numerical Solutions of Ordinary Differential Equations in a Transputer System

Jiří Kunovský, Karel Mikulášek, Viktor Němec

Technical University of Brno

The paper deals with numerical solutions of ordinary differential equations. The Modern Taylor Series Method is used.

It is known that the Taylor series method is a parallel one and therefore attempts at a parallel interpretation in a transputer network can be expected. Transputers are high performance microprocessors that support parallel processing. They can be connected together in any configuration and they can form a building block for complex parallel processing systems.

The block notations and block diagrams (used in analogue and hybrid computers) represent a very convenient tool for describing parallel tasks in a transputer network.

The integrators, summers, multipliers, dividers, ..., are taken as procedures in OCCAM, the corresponding calculations are viewed as processes. The integrators and summers are interconnected by channels in OCCAM.

As in the OCCAM programs only very simple communication with the operator in the form of a "running column of numbers" is possible and as a debugging a program in OCCAM is complicated, the TKSL/TRANSP has been created.

TKSL/TRANSP created for simulating dynamic systems makes it possible to automatically write a program for a transputer or a network of transputers from a textual description. In fact, the only interconnection of analogue elements must be described in the textual description. Thus, when simulating, it is no longer necessary to debug a program in OCCAM.

Moreover, a repeated start of the calculation is possible in TKSL/TRANSP without having to load a new program into the transputer.

When using a transputer network, any element of the state diagram can be placed practically in any transputer of the network. The problem is, however, the serial communication. The reason is that the serial communication may be almost as time consuming as the calculation itself.

The system elements must then be placed in the network in such a way that the transputers communicate with one another as little as possible.

A TRANSPUTER NETWORK HOST LOAD SIMULATION.

G.K. Mishkoy, I.G. Grama, B.P. Rybakin, O.A. Trofilov

Institute of Mathematics, Academy of Sciences of Moldova

In the organization of parallel computations in transputer networks much attention is paid to load algorithms. Here we are concerned with the algorithms of a transputer system's host work control in the stationary mode. Upon that, as an effective technique of the load simulation and that of the examination of such algorithms use is made of a queueing theoretical model.

It is assumed that the Poisson arrival task stream have been split on any number r of the homogeneous task classes called the priority classes. It is known that the first step in the transputer network load [1] is the network dumping, after that the processors in the network pass into the load mode on one of its network line. Therefore, as it is further believed, on a host, besides the service process, there is another random process, with the help of which the time losses on , so called, "switchings" of the priority classes or their "orientations" will be simulated. The expenditures on a "switching" within a priority class are equal to zero. Following Kendall such a queueing model with the priority and orientation is noted by $M_r/G_r/1$.

The class of host load control algorithms has been built on the basis of the analytic correlations describing the busy period behavior within a network host. Each algorithm of the class depends on the parameters of the arrival streams, time losses for the priority "switchigs", the tasks' service time distributions and other. Each algorithm involves the fast-algorithms for the solution of the recurrent equations and the calculating methods to inverse Laplace and Laplace-Stiltjes transformations, as well as the object-oriented approach to programming.

The operation mode analysis is made by the object-oriented programming technique for common goals of priority systems simulation [2]. By means of the programming language Turbo-Pascal 6.0 the system of the objective types describing a queueing system with priority and orientation have been developed. Also, in order to evaluate the set of the work mode characteristics of a system with r priorities as well as to "evaluate" the possibility or the impossibility of such a system to be functioned in the stationary mode the procedures using the above collection of means and elaborations have been constructed. Constructively, on the each step $k \geq 1$ the characteristic ρ_k is calculated by using the procedures of the packet in force and then the condition of stationarity $\rho_k < 1$ is checked. If the condition is true for each k up to r then it means that the host load can function in the stationary mode under given parameters. Simultaneously with that the characteristics of the stationary mode are evaluated. Numerical results of the simulation under various specific parameters have been obtained.

References

[1] Transputer Applications. Edited by G.Harp, Moscow, 1993 (in Russian)

[2] Grama I.G. and Mishkoy G.K. Object Oriented Programming for Queueing System. Computer Science Journal of Moldova, No 1, 1993.

SCHEDULING AND METASCHEDULING IN MULTIPROCESSORS: EVOLUTIONARY COMPUTATION APPROACH

Franciszek Seredyński
Institute of Computer Science, Polish Academy of Sciences
Ordona 21, 01-237 Warsaw, Poland
e-mail: sered@ipipan.waw.pl

In this paper a variant of the job-shop problem related to the area of parallel processing is considered. The problem concerns scheduling tasks of a parallel program on multiprocessors. An approach to solve the problem based on application of evolutionary computation methodology is proposed.

It is assumed that a parallel program is represented by a directed and weighted graph called a *precedence task graph*. Nodes of the graph represent elementary tasks which are indivisible computational units. Weights of the nodes describe the processing time needed to execute a given task on any processor of a given multiprocessor system. Edges of the precedence task graph describe the communication pattern between the tasks. Weights of the edges describe a communication time between pairs of tasks k and l, when they are located in neighbour processors.

A multiprocessor system is represented by an undirected graph called a *system graph*. Nodes of the system graph represent processors of a parallel computer of MIMD architecture. Edges represent bidirectional links between processors and define a topology of a communication system of the multiprocessor system.

The objective of the scheduling problem studied in this work is to find an assignment of tasks into processors to *minimise the total execution time* of a parallel program. Since this optimization problem is know to be NP-hard, to solve the problem with a realistic size of the precedence task graph some effective heuristic methods should be applied. In this paper, the evolutionary computation methodology based on *genetic algorithms* (GAs) and *genetic programming* (GP) is applied.

A *chromosome representation* for a scheduler describing the scheduling task is proposed. A number of specialized *genetic operators* supporting the chromosome representation has been designed. During a life cycle of each scheduler-chromosome genetic operators are applied to produce new legal solutions of the problem. A population of schedulers evolves in the evolutionary process to search optimal or suboptimal solutions.

The evolutionary system has been implemented on a sequential computer. Results of conducted experiments show that the system is able to discover very effectively a solution, improving sometimes solutions presented in the literature.

While the problem of scheduling tasks it closely related to the area of parallel computing, we face a more general problem deriving a solution of the problem. Our simulation system is described by a set of parametres, and by a set of genetic operators. What are the right values of these parameters, and what is the best order of applying operators to use the most effectively the simulator? To solve these problems we introduce a second level of schedulig, which we call *metascheduling*. We show that introducing the metascheduling implemented also with use of the evolutionary computation methodology, particularly using GP and GAs techniques, increases the efficiency of use of the simulator and quality of obtained solutions.

Parallel simulation programs optimum partitioning

Uljanov Alexander

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: $\rho_u : U \to R^+$ and functions of weights of vertexes $\rho_u : X \to R^+$. We have to calculate a partitioning X' of set X with limitation $\sum_{X'' : x_i \in X''} \rho_x(x_i) \le V^*$ and a criterion $J = \sum_{U_r} \rho_U(x_i, x_j)$ must be minimum,

where

$$U_r = \{(x_i, x_j) / (x_i, x_j) \in U \wedge (\exists X'' \in X')\left[(x_i \in X'' \wedge x_j \notin X'') \vee (x_j \in X'' \wedge x_i \notin X'')\right]\}.$$

To solv this problem the Floyd-matrix of the shortest chains $R^l = \|r_{ij}^l\|$ is used. where

$R^l = \|r_{ij}^l\|, i, j = 1, 2 \ldots, n, r_{ij}^1 = \rho_u(x_i, x_j)$, when $(x_i, x_j) \in U$ and $r_{ij}^1 = \infty$, when $(x_i, x_j) \notin U$.

Let us accept, then vertex subsets $x_{l_{n-2}}^{(1)}$ for the vertex x^l. $d(x^l, x_{l_n}^{(1)}) \le V^*$ and a shortest chain $C_{l_1} = x^l, x_{l1}^{(1)}, x_{l2}^{(1)}, \ldots, x_{lm_1}^{(1)}$ exist. We have a net with the source $x_{lk_1-1}^{(1)} \in X_1^{(1)}$ and flow-off $x_{lk_1}^{(1)}$ (or in the opposition direction). The maximal flow in this net $U_1^{(1)} = \min_{\sigma^1 \subseteq U} \sum_{(x_i, x_j) \in \sigma^1} \rho_u(x_i, x_j)$

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

$x_{lk_1}^{(1)} \in \overline{X}_1^{(1)}, X_1^{(1)} \wedge \overline{X}_1^{(1)} = \{\varnothing\}$. $X_1^{(1)} \cup \overline{X}_1^{(1)} = X$.

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 $x_{l_{n-2}}^{(1)}$ is a source and the vertex $x_{lk_1-1}^{(1)}$ is a flow-off. Now the maximal flow is to get according to formula:

$$U_2^{(1)} = \min_{\sigma^1 \subseteq U} \sum_{(x_i, x_j) \in \sigma^2} \rho_u(x_i, x_j), V_2^{(1)} = \sum_{x_i \in X_2^{(1)}} \rho_x(x_i), x_{lk_1-2}^{(1)} \in \overline{X}_2^{(1)}, X_2^{(1)} \wedge \overline{X}_2^{(1)} = \{\varnothing\}, X_2^{(1)} \wedge \overline{X}_2^{(1)} = X \ldots$$

and so on, until we get the maximal flow between x^l and $x_{l1}^{(1)}$. The k_1 minimal partitionings and k_1 variants $X_1^{(1)}, X_2^{(1)}, \ldots, X_{k_1}^{(1)}, X_i^{(1)} \subset X, i = 1, 2, \ldots, k_1$ will be calculated using the chain c_{l_1}.

Further the following three components will be got for all the chains:

$$V' = \{V_1^{(1)}, V_2^{(1)}, \ldots, V_{k_1}^{(1)}, V_1^{(2)}, \ldots, V_{k_2}^{(2)}, \ldots, V_1^{(n)}, \ldots, V_{k_n}^{(n)}\}$$

$$X' = \{X_1^{(1)}, X_2^{(1)}, \ldots, X_{k_1}^{(1)}, X_1^{(2)}, \ldots, X_{k_2}^{(2)}, \ldots, X_1^{(n)}, \ldots, X_{k_n}^{(n)}\}$$

$$U' = \{U_1^{(1)}, U_2^{(1)}, \ldots, U_{k_1}^{(1)}, U_1^{(2)}, \ldots, U_{k_2}^{(2)}, \ldots, U_1^{(n)}, \ldots, U_{k_n}^{(n)}\}.$$

where

$$|V'| = |X'| = |U'| = \sum_{i=1}^{n} k_i \ (\exists i, j, m, l)\{[V_{ij}^{(i)} = V_{km}^{(l)}] \mathbb{I} [X_{ij}^{(i)} = X_{km}^{(l)}] \mathbb{I} [U_{ij}^{(i)} = U_{km}^{(l)}], U_{ij}^{(i)} \ge \rho_0, X_{ij}^{(i)} \subset X.$$

Then the optimal partitioning $X_r^{(q)} \in X'$ is to calculate with function

$$f = \min_{i=1,n} \min_{j=1,k_i} \{l_1 * [V^* - \sum_{x_k \in X_j^{(i)}} \rho_x(x_k)] + l_2 * \frac{\sum_{(x_P, x_m) \in U_r^{(i)}} \rho_U(x_P, x_m) - P_0}{\sum_{\substack{(x_d, x_a) \in U \\ x_d, x_a \in X_r^{(q)}}} \rho_U(x_P, x_m) - P_0}\}.$$

where minimum is to calculate for all i,j:

$$C_r^{(i)} = \sum_{x_k \in X_r^{(i)}} \rho_x(x_k) \le V^*.$$

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 $|V'| = |X'| = |U'| = \sum_{i=1}^{n} k_i$ is enough great.

Performance Analysis of Prioritized HMPMB Multiprocessor Systems

Hu-Jun Wang Hui-Zhong Wu
Department of Computer Science, NanJing University of Science and Technology
NanJing 210094, P.R.China

Jian Li
SiWei Computer Corporation, ZhuJiang Road 448, NanJing 210018, P.R.China

Abstract

The principal characteristic of a multiprocessor system lies in the capability that all processors share a single main memory. This sharing capability is provided through an interconnected network between the processor and the memory module. A few interconnection networks(INS) have been proposed by many research groups for interconnecting multiple processors, including crossbar and multiple bus, etc.. However, the complexity of these INS is prohibitive for large scale multiprocessor systems.

Recently, hierarchical systems are designed and analyzed to reduced the network complexity by in incorporating hierarchies of interconnection networks. Nevertheless, their cost and complexity are still higher when the number of processors and memory modules is greater. In order to reduce the cost and complexity of the system without significantly reducing the performance, we present an improved m-level hierarchical memory-oriented partial multiple bus(HMPMB) multiprocessor system.

The m-level HMPMB architecture consists of a global interconnection network(GIN), a cluster interconnection network(CIN), or called local level, and a nonlocal interconnection network (NLIN). It is assumed that $N = n \times k_1 \times k_2 \times \cdots \times k_{m-2} \times k$(let N be the number of processors or memory modules). In the CIN, it consists of N/n clusters of processors and memory modules. Each cluster consists of n processors and n memory modules interconnected by means of a CIN, which is a memory-oriented partial multiple bus(MPMB) IN of size $n \times n \times b_c/g_c$. In a cluster, n memory modules are divided into g_c groups with each group of n/g_c outputs fully connected to a set of b_c/g_c buses, whereas all n processors are connected to all buses.

The GIN is a MPMB IN of size $k \times k \times b_g/g_g$. In this network, k outputs are divided into g_g groups with each group of k/g_g outputs fully connected to a set of b_g/g_g buses, whereas all k inputs are connected to all buses. Between the local and global levels there exist m-2 nonlocals(NL). NL_j is a MPMB IN of size $k_j \times (k_j+1) \times b_{nj}/g_{nj}$. In this network, k_j outputs are divided into g_{nj} groups with each group of k_j/g_{nj} outputs fully connected to a set of b_{nj}/g_{nj} buses. Each output of the GIN is connected through a hierarchy of m-1 buses. Each output of the NL_jIN is connected through a hierarchy of j-1 memory bus.

The performance of multiprocessor systems has been studied extensively in a number of papers, but processor priorities were not considered in most of the published literature. In an actual systems, simultaneous requests to the same memory module can be resolved on a priority basis. Thus, we analyze the bandwidth performance of a prioritized HMPMB system in detail. This bandwidth model is useful not only to quantify a set of parameters for a given configuration, but also to investigate the effect of different parameters on system performance.

In order to validate the modeling technique, the system was simulated. Every simulation-run was repeated 10 times with different seed values to determine the 95% confidence interval of the results. Requests are generated randomly by each processor. A blocked processor will reissue the same memory request in the next cycle.

The motivation of the work of this paper is to develop analytical models for prioritized hierarchical multiprocessor systems for the favorite case so that the performance of such systems can be accurately predicted using analytical models.

Parallel simulation of hierarchical digital systems

A. Hunger, S. Werner, F. Müller

Gerhard Mercator Universität Duisburg Gesamthochschule
Fachbereich 9 / Datenverarbeitung
Bismarckstraße 81
D-47057 Duisburg

Topics: Applications (VLSI-CAD), Simulation of VLSI circuits, Distributed Processing

Abstract
With this paper, a strategy is explained which allows to map the hierarchy of a model onto a network of parallel computing nodes for simulation purposes. It is shown that exploiting hierarchy can support the solution of many computational tasks, especially the devide and conquer approach of partitioning a given problem before solving it. Demonstrated results were gained in the field of logic and fault simulation of digital circuits as well as in the test generation for electronoc devices; the parallel computer used is a transputer net with 40 nodes. The developed approach shows general strategies for a broad class of applications in discret event simulation.

1. Hierarchical Modelling

This principal is explained by way of the design of digital circuits. All CAD tools provide the user with some basic primitives, out of which he can construct a first set of subdevices (e.g. adder, multiplexer, and decoder may be constructed using basic gates). In a second step, more complex modules will be constructed, e.g. an arithmetic unit consisting of several adders and multiplexers. The designer takes advantage in the high degree of equality of the instances in the sense that he constructs each module only once, on the next higher level of design he will only refer to the memorized design of subcircuits, i.e. he uses the subcircuit in many instances. Most CAD tools flatten such a hierarchical design and work on a representation which describes all instances in the same detailed way every time the subcircuit is used.

Due to the fact, that such flattened descriptions of large designs are workloads difficult to handle sometimes even for powerful workstations, the simulator TESI (TEst and SImulation) has been developed. Exceptional feature of this simulator is that the hierarchy of the design is not only used for model description but also during run time of the simulator. For this purpuse, a substructure of a circuit which is used in multiple areas of the model is stored only once; only a reference to its unique definiton is stored at the places where the substructure is used. This fact results in very compact internal data structures. As a consequence, the required space for working memory as well as CPU time for loading and storing the net list are dramatically reduced. /1/.

2. Parallel simulation

This technique has been developed for an ordinary sequential simulator. In addition, it turned out that it can be exploited as a basic strategy for parallel simulation. Reasons are:

- The simulation of a large circuit can easily be broken into the simulation of substructures given by design hierarchy. No time consuming cutting algorithms have to be used to create artifical subcircuits.
- Due to the fact that many substructures are identical (i.e. instances of the some basic type), the number of different substructures is significantly smaller than the number of sub-structures of similar sizes gained from a cutting algorithm. Therefore this partitioning will map on a net with fewer computing nodes.
- Due to the very compact internal data structures of these models it is possible to load many different instances (sometimes the entire model) on each and every computing node of the parallel computing net. As a consequence mechanisms can be constructed which allow easy and fast load ballancing when the simulation of some instances requires significantly more or less computing time than the average of all nodes of the net.

Based on these considerations a parallel version of the simulator TESI has been implemented on a transputer net with 40 nodes. The described devide and conquer method revealed to be efficient although frequent communication between nodes is obviously the drawback of this approach. Alternative experiments using different strategies (e.g. partitioning of the fault list) showed that hierarchy can be used as a reasonable way of problem partitioning. Together with the advantages in memory requirement and load time it may be especially worthwhile to use it for the simulation of large circuits /2/.

/1/ Hunger, A., Papathanasiou, A.: Acceleration of simulation time by hierarchical modelling. In: Silvester, P.P.: Software Applications in Elelectrical Engineering, Computational Mechanics Publications, Southampton, 1993

/2/ Müller, F.: Methodische Analyse der digitalen Fehlersimulation als Grundlage für die Entwicklung effizienter Parallelisierungsansätze. Dissertation im Fachbereich Elektrotechnik der Gerhard Mercator Universität Duisburg - GH, 1994.

SDL Integrated Tool Environment - SITE

DESCRIPTION

The Tool Set provides an open environment for SDL'92 development and analysis tools. All tools are single stand-alone components, which communicate via the Common Representation. This allows an independent development of new tools as well as the integration of third-party products.

Currently available are the *Parser* (according to Z.100 with tool dependent ASN.1 extension as well as according to Z.105), the *Semantical Analysis*, the *Simulation* (SIM-Lib, SDL-Lib), the *Petri-Net-Analyser* (dedicated to SDL-Time-Nets), the *Syntax-Oriented Editor for SDL/PR*, the *Pretty Printer for SDL/PR* and the *C++-Generator*. The work on the *Petri-Net-Generator*, the *Graphical Editor*, and the *Result Presentation* is under progress. A *runtime library* to generate prototype software is part of a third party project and not public available.

All tools are developed in C resp. C++ and are portable to different platforms.To keep the expenditure for the development process low, metatools were extensively utelized.The major metatools are

- yacc and lex (resp. Bison and Flex) for the lexical and syntactical analysis
- *Kimwitu* for all CR-operations (CR-construction and semantical analysis)
- Gnu tools like *awk, perl, rcs...*
- own developments, e.g. the *Scanner-Parser Generator* for incremental analysis.

ENVIRONMENT

SITE was developed on SUN work stations with SunOS/Solaris. Graphical applications are X-based. Adaptations to other UNIX platform are possible. For the tool demonstration the preferred environment is:

- Sun (compatible) SPARC station 5,10, or 20, colour monitor
- SunOS 4.1.x with X or Openwin
- Gnu development kit if any problems occurs

CONTACT

Dipl. Inf. Ralf Schröder
Humboldt University Berlin, Germany
Institute of Computer Science; Research domain „System Analysis"
Lindenstr. 54a
10117 Berlin, Germany
Phone: +49 30 20 181 321
Fax: +49 30 20 181 234
e-mail: schroed2@informatik.hu-berlin.de
www: http://www.informatik.hu-berlin.de/Institut/struktur/systemanalyse

AVAILABILITY

Not commercially available, research licence necessary

ESPRIT-III-Project

INSYDE

Integrated Methods for Evolving System Design

Project Objectives:

The main objective of the INSYDE Project is to define, implement, validate and demonstrate a comprehensive methodology for hybrid hardware/software system design. The result of the project will be a prototype methodology and a toolset, exploiting and combining the advantages of object-orientation and formal description techniques (both for hardware and software design), which covers the full development cycle from system requirements analysis to system architectural design and validation. The INSYDE Methodology will be used in selected application domains (telecommunications and microelectronics hardware/software systems).

Project Partners:

- Verilog SA (France)
- Intracom SA (Greece)
- Alcatel Bell Telephone (Belgium)
- Humboldt-Universität zu Berlin (Germany)
- Dublin City University (Ireland)
- Vrije Universiteit Brussel (Belgium)

Duration: March 1994 – March 1996

Project Description – The INSYDE Methodology:

In the INSYDE Methodology a hybrid system is modelled at different levels of abstraction:

1. Conceptual Level:

The Conceptual Level describes the *WHAT*. The Conceptual Model is the result of a rigorous analysis of the functional aspects of a system. Non-functional aspects, such as performance, reliability, cost, etc. are outside the scope of the project. Analysis is done using the object-oriented modelling technique OMT.

2. Architectural Level:

The Architectural Level describes the *HOW*. Although the Conceptual Model provides enough information to describe the required architecture and functionality of a system, it is not formal enough to provide a basis for code generation or to allow simulation. Therefore, a more formal model – the Architectural Model – is constructed using the formal description techniques SDL (for the software components) and VHDL (for the hardware components). The transition from the OMT description to the SDL/VHDL specification is supported by software-tools.

3. Validation Level:

The Validation Model is an executable simulation model (realized by execution of e.g. generated C or C++ code). It is the result of an automatic transformation of the Architectural Model. In the INSYDE Methodology the validation is realized by co-simulation of the hybrid system specification: An SDL simulator and a VHDL simulator are used to simulate the SDL and VHDL part of the system specification; the interaction and synchronization between both parts is realized by a coupling module, which connects the simulators.

Performance Prediction System PEPSY for Parallel Programs *

Roman Blasko
Department of Software Technology and Parallel Systems
University of Vienna, Liechtensteinstrasse 22, A-1092 Vienna, Austria
e-mail: roman@par.univie.ac.at

Abstract

Parallel programs, generated by the supercompiler, VFCS (Vienna Fortran Compilation System), are an application area for the tool, PEPSY, which we have developed recently. The initial sequential version of the program, written in Fortran'77 or Vienna Fortran programming language, is parallelized by VFCS, which translates a Fortran program into an explicit parallel program with message-passing statements, for the parallel computer Intel iPSC/860. Performance prediction should be done before generating the final code for the target compiler, i.e., before a real run is executed on the parallel computer. Performance results should be used for the comparison of several variants of the parallel program, and for recognition of the most critical parts in the program, so as to find the optimal parallelization strategy for VFCS.

PEPSY (PErformance Prediction SYstem) is a tool based on automatic modeling of parallel programs and performance analysis based on discrete-event simulation. The tool consists of two modules, viz. MOGEN and PROGAN. The automatic modeling technique is implemented in the functional module MOGEN (MOdel GENerator), which is integrated with the internal representation of the parallel program in VFCS. The modeling is based on the **Process Graph** (PG), which we have defined for the abstract representation of the sequential and parallel Fortran programs. PG model of the program is analyzed by the performance analyzer PROGAN (PROcess Graph ANalyzer), based on the discrete-event simulation technique evaluating a set of static and dynamic **performance parameters**. *Static parameters* characterize the size and structure of the modeled system, i.e., parallel program. We have defined nine parameters characterizing the size of the model and interconnections among nodes of the model. *Dynamic parameters* characterize the behavior of the model for steady or transient states. PEPSY evaluates about ten basic and several derived dynamic indices, e.g., execution time, computation time, communication time, parallelism degree, utilization, communication volume, etc. Comparing different versions of the parallel program, PEPSY allows to evaluate the *scalability parameters*, e.g., execution signature, speed-up, efficiency, and efficacy.

Monitoring facilities of PEPSY evaluate the performance indices for the whole execution period of the parallel program or for the detailed dynamic behavior, evaluating the values by an optional sampling period. The performance results are provided, in a hierarchy, for the global or *program level*, as an average values for all parallel processors, for the *processor level* comparing all parallel processes, e.g., for the load balancing analysis, and for the intraprocess level, i.e., the *statement level* of the parallel program. Using different monitoring facilities, PEPSY produces a large spectrum of performance characteristics for the parallel program. These information enable the user to find the critical parts and variants of the program, to be modified for the optimal parallelization strategy. A practical performance prediction for parallel programs, using PEPSY, is presented by examples.

*The work described in this paper is being carried out as a part of the research project " Performance Prediction and Expert Adviser for Parallel Programming Environment", funded by the FWF, Vienna, Austria, under the grant number P9205-PHY.

Sensor based Simulation System for Planning and Programming of Manufacturing Processes in One-of-a-kind Manufacturing

Dipl.-Ing. Ulrich Berger, Dr.-tech. Johannes Krauth, Dipl.-Ing. Achim Schmidt
BIBA, Bremen (D)

A major approach to improve quality and cost relation in one-of-a-kind production is the introduction of automation technology with a high degree of flexibility. Existing planning and programming systems (e.g. IGRIP, ROBCAD, GRASP) cover only standard off-line programming features, therefore real workpiece or tool geometries have to be implemented by manual teach-in or similar procedures in the shop-floor area. This reduces the productivity of the manufacturing facilities, and thus the cost/benefit relation is poor. To improve this situation we have developed and realised a sensor based simulation environment for planning and programming of manufacturing processes involving industrial robots.

The new technological approach is the connection of active vision with robot simulation and off-line programming systems. So necessary geometrical data can be acquired from topometric measurement and integrated in the planning and programming task. With this technology not only the described gap in off-line programming can be decreased, but also functions like geometrical inspection during/after manufacturing sequences or calibration of robot cell devices can be demonstrated. To prove the claimed functionality under real manufacturing conditions, an experimental environment has been realised which consists of three main items:

1. Graphical simulation, visualisation and off-line programming system (IGRIP). A new feature is an advanced interface for rapid integration of topometric data, CAD-product data, manufacturing cell models or peripheral devices. The system enables the planning, layout design and simulation of robot based manufacturing of one-of-a-kind products.

2. Sensoric system based on coded light approach and phase shifting techniques. Realised by a sensoric head mounted on a spherical positioning system to cover various product shapes. To modify scenery information into 3D-clusters, preselect, extract and calibrate object data and transfer it to the simulation system, a unique communication process has been developed. As a result, the amount of input data for the simulation is reduced substantially.

3. Flexible manufacturing cell, based on a joint-coordinate industrial robot with rapid tool changing facilities and positioning devices for sensor and workpiece handling.

Field test experiments using arc-welding, spherical flame-cutting and assembly show the overall function of the described technology. Further exploration in view of manufacturing of free from surfaces is under development. The research work has been carried out in the ESPRIT project "NEUROBOT".

Mailing address:	Phone:	Fax:	e-mail address:
Bremen Institute of Industrial Technology and Applied Work Science at the University of Bremen (BIBA) P.O. Box 330560 28335 Bremen	+49/421/218-5552	+49/421/218-5510	ber@biba.uni-bremen.de

Simulation Language TKSL/TRANSP and Partial Differential Equations

Jiří Kunovský, Viktor Němec, František Zbořil

Technical University of Brno

The paper deals with parallel solutions of parabolic, hyperbolic and eliptic partial differential equations (PDE) in a transputer network. The method of lines is applied. Three-point-approximations and five-point-approximations are used for numerical solutions of PDE.

An original numerical method has been worked out based on the use of analogue functional blocks and on the use of analogue schemes. For this method procedures have been created describing the activities of the basic functional blocks (integrators, invertors, multipliers, dividers etc.). The functional blocks are interconnected via channels in OCCAM. The efficiency of the numerical solutions of systems of differential equations in transputer networks is improving with the number of differential equations solved (with the number of functional blocks used), and thus the proposed method is suitable for large systems of differential equations. The paper demonstrates a comparison of computations in a one-transputer and a two-transputer network. To stress the effect of the extreme speed and accuracy of the computation the paper is complemented by an analysis of the effect of the order of numerical integration.

TKSL/TRANSP created for simulating dynamic systems makes it possible to automatically write a program for a transputer or a network of transputers from a textual description. The textual description is based on analogue diagrams.

The paper demonstrates an application of transputer networks to the numerical solution of partial differential equations. The worked out method can be used for arbitrary systems of differential equations. For its illustrative power the method has already been used in teaching Parallel Architectures at the Technical University of Brno.

The worked out method of applying transputer networks has been successfully tested in solving special parabolic, hyperbolic and eliptic partial differential equations (and corresponding large systems of ordinary differential equations - a problem of 3,000 differential equations and a system of 8,000 differential equations has been solved).

Comparison between Ptolemy, SynDEx and TOPModèle shells

Laurent KWIATKOWSKI, Fernand BOERI and Jean-Paul STROMBONI

Laboratoire UNSA d'Informatique Signaux Systèmes (I3S) - URA 1376 du C.N.R.S.
41, Boulevard Napoléon III - F 06041 Nice Cedex - Tél : 93.21.79.61 - Fax : 93.21.20.54

Abstract : (KEYWORDS : Parallel simulation - Modelling - Development of Simulators - Prototyping)

The purpose of this paper is to propose a simulation tool called TOPModèle and to compare it with several development shells (DS) as SynDEx of INRIA Rocquencourt in FRANCE, Ptolemy of the University of California at Berkeley, USA. In many scientific areas, applications involve a large amount of computation suggesting the use of multiprocessors in order to speed up processing. Given an application and a set of architectural restrictions, the goal is the application partitioning so that every part will be processed by a different processor, but the research of the tasks allocation that minimize execution time is not easy since we know that the task allocation problem is a NP-complete one. Nevertheless, a lot of heuristics have been developed and implemented on DS in order to find sub-optimal solutions. Obtaining of optimal performances is a problem where it should be considered at once the potential parallelisms of the studied application, the available parallelisms of the computer and the allocation strategy.

First, study of parallel implementation methods used on DS emphasizes the problem of the granularity tasks application when modelized by means of a graph. The task granularity is defined as the ratio of the mean task execution time to the total application processing time. Exploiting the whole potential parallelism requires fine grain but leads to combinatorial explosion when the size of the application increases. Usually, this explosion could be reduced by increasing granularity, but potential parallelisms are decreased. Ptolemy and SynDEx use a task granularity depending on the application size. Simulations with TOPModèle show that parallel performances depend on this granularity and that DS should take into account the CCR (communication computation ratio) parameter for choosing an assignment strategy (mapping/scheduling, clustering and duplication heuristics) and the granularity of modelling in order to exploit potential parallelisms and raise parallel performances to a maximum.

Secondly, An analysis of some existing multiprocessors shows a large number of differences. Parallel machines contain many processing units of various powers calculating simultaneously. An internal network is used for exchanging intermediate data in parallel. Various natures, topologies and methods for communication have been proposed and achieved yet. An universal model is difficult to find because of the diversity of these architectures. Therefore, DS like Ptolemy or SynDEx modelize a particular machine hardening the research of an adapted architecture to the considered application. The target multiprocessor system considered is often a MIMD (Multiple Intructions Multiple Data) with homogeneous processors connected in a fixed topology. Each processor has a CPU and communication units that share a local memory and work concurrently. On Ptolemy, topology is fully connected or a shared bus with a variable number of homogeneous MIMD processors. Communications are executed through a DPM (Dual Port Memory). There are divided in two similar parts : the transmitter processor writing on DPM and the receiver processor reading on the same DPM. SynDEx is dedicated to T800 transputers or TMS320C40 DSP.

On TOPModèle, the set of processing units and communication links is described by an oriented graph. The vertices are the processors. They include one or more modelized functions : processing, memorizing, routing and sequencing. Then, the machine could be homogeneous or heterogeneous, the memory shared or distributed. The vertices may be partitionned to specify the multiSIMD mode. Therefore, a set of Instruction Registers (IR) is defined. Every class is associated one IR and every processing unit in the class executing the operation registered in that IR. An edge leaving a vertex specifies a physical unidirectional communication link towards a neighbour processor. One wants to stipulate that an input/output port in the machine is shared between several links. Therefore, *links* are grouped in parallel *ways* i.e. set of links able to transfer simultaneously and ways are grouped in communication *channels*, in which may are switched successively by the model during communication. Actual machines propose various parallel modes for processing the delivered informations. They differ by control and synchronization in particular the start, stop and load conditions for processing. Three modes are implemented in TOPModèle. Informations are not actually processed, only the ressource activity is simulated. In SIMD mode, the constraint is that parallel resources are whether idling or running the same operation. The starting condition is the availability of all the processing resources. Already delivered informations are reviewed and if they are associated with a common operation, an occupation delay is deducted for every involved ressource. The occupation indicators are initialized with the calculated load. They will be decremented during the simulation until the availibility of all ressources. The SPMD mode starts with the same condition but simultaneous execution of different operations is allowed. In MIMD mode, synchronization is local. Activating a processor is independant of the other processors activity. The arrival of information in a processor initiates processing with the specified operator as soon as this processor is free. For all these modes, it is possible to specify that transfer and processing overlap or the whole transfer occurs first and then only processing.

After the application and machine modelling, TOPModèle delivers the results of execution timings with the instantaneous utilization of all the machine resources (links, memomies and processing units), the execution time of the application on a target multiprocessor in number of clock cycles, speedup for processing and exchanges, load repartition obtained with a given mapping. TOPModèle is a prototyping assistance for parallel applications design. This prototyper determines some minimum sizes requires for the parallel resources, i.e. memory, registers, FIFOs and I/O ports... Given CCR, this prototyper allows to choose the assignment strategy and task granularity and to change architectural constraint in order to research the best parallel performances.

SIMULATING LANDSCAPE SUSTAINABILITY WITH "MELAS" (MEditerranean LAndscape Sustainability Simulation System)

Fivos Papadimitriou
School of Geography,
University of Oxford.

Abstract

MELAS is an expert system designated to allow the user to simulate the behaviour of Mediterranean landscapes in time with respect to the concepts and principles of sustainability.

MELAS performs five different tasks.

1. Assessment of Mediterranean landscape sustainability on the basis of land use change.
2. Assessment of Mediterranean landscape sustainability on the basis of soil data.
3. Assessment of Mediterranean landscape (land use or soil) sustainability on the basis of land management and disturbance.
4. Sustainable land and soil use optimization.
5. Sensitivity analysis of the solutions.

The data the user imports are the following.

a) Land Use / Land Cover (area or biomass percentage), from time "1" to time "10". (forests , agriculture, shrublands - grazeland, bare ground)

b) Land Use / Land Cover Transitions (16 transition types), which are constant for the time interval from time "1" to time "10".

c) Land Use Sustainability thresholds.

d) Soil features (units as appropriate), for one or more time points, from time "1" to time "10" (soil depth, soil productivity)

e) Soil Sustainability thresholds

Items c and e are constant through time.

The main sustainability condition is that a landscape is sustainable if the soil erosion rate is minimum, the degraded land cover is also minimum and the soil productivity is maximum.

A SIMULATION SYSTEM FOR FORECASTING THE IMPACT OF CLIMATE CHANGE ON MEDITERRANEAN VEGETATION

Fivos Papadimitriou
School of Geography,
University of Oxford.

Abstract

The simulation system "MEDCLIVE" is designed to give for every place (φ,λ) within the boundary of the Mediterranean and time (year from 1990 until 2030) the response of the vegetation type (natural, seminatural and cultivation) to climate change for any climate change scenario. It can also show the sensitivity of the vegetation type to changing climatic parameters for the place and time these parameters are considered.

The grid resolution covering the Mediterranean is 2 by 1 degrees longitude and latitude respectively. Thus, it is possible to calculate the dominant vegetation type corresponding to the given climatic parameters on each one of the resulting grid cells for any year (until 2030).

MEDCLIVE is an simulation system suitable for research, as well as for training purposes. It serves the user by allowing to make rapid estimations of the impact of possible climate change on vegetation in the Mediterranean. In this context, the term "vegetation" describes both agricultural crops and natural vegetation. The most appropriate climatic parameters and the most representative vegetation types were selected for inclusion in the system's database.

C_E_D_L_X a tool for modeling of chemical kinetics

T. Preiß, F. Breitenecker
TU Wien, Dept. Simulation Technique

I. Introduction

In applied chemistry the factors, which determine the run of a chemical reaction, has to be examined. Due to the knowledge of these factors the chemical reaction can be influenced in an optimal way, and it is possible to answer questions about the chemical process without experiments.
The mechanism of a chemical reaction therefore is determined by the kind and number of elementary reactions. The fundamental task of reaction kinetics is to find out the reaction mechanism. Therefore this tool is implemented to support the modeling of chemical systems and the most important experiments applied on the chemical systems.

II. Description of the facilities of the system

The system provides several means of describing a chemical reaction.

- It is possible to write the equations of the chemical reaction with an editor and simulate the system. So a chemical equations' description language is implemented, which uses special keywords as `matter, reaction, constant, step, time, temp` and beside the normal expressions the two special arrow operators `(order, constant_expression>` and `<order, constant_expression/ order, constant_expression>`.
  Each `reaction(input_list; output_list; parameter_list)` can be part of a larger model - So libraries of often used submodels can be built. It is also allowed to nest the model descriptions to gain a fine local structure of the model.
- On the other hand the scientist can use the graphical possibilities of the UNIX X-Window based system and to draw the reaction with the mouse.
  This method uses several basic objects to model the chemical equations: The participating matters are linked together with „arrows" of variable types (reaction in one or two directions (reversible reactions), catalytic reactions).
  The participating matters are the sum of the compounds of one side of the chemical equation. These compounds are linked together by the arrows characterizing the kind of the reaction between the participating matters.
  The attributes of the arrows are the order of the reaction and an expressions characterizing the constant of the speed of the chemical reaction.
- The chemical model results in a mathematical system of differential equations. Special algorithms have to be used because this system is stiff - the algorithm for the solution of differential algebraic systems DASSL also fulfills the necessities of an algorithm to solve stiff systems.
- The experiments are variation of parameters and steady state analysis.
- If further experiments have to be made, the model also can be translated in the most popular simulation-languages like ACSL, MOSIS, SIMUL_R.

III. Steps of Realization

The realization of this project is organized in several phases:

- Implementation of the extended C_E_D_L_X Compiler using LEX and YACC.
- Implementation of a character based User Interface for the solution of the chemical systems.
- Implementation of the Graphical User Interface using the script language TcL/Tk to enable a graphical way to input the model using the basic graphical object types.
- Generation of the code of the description language based on the graphical description.
- Implementation of algorithms for experiments and algorithms for the solution of the differential equations.

An Object-Oriented Graphical Environment for Modelling and Simulation of Complex Process Units

J. Schmuhl, K.-P. Kniess

The modelling of process units in chemical engineering is realized mostly by the generation of complex models describing the basic processes. The complex model usually consists of variety of basic processes like reaction, adsorption, heat exchange, etc. Therefore it is possible to combine their basic models in a complex model.
We present an object-oriented graphical environment under UNIX system for modelling and simulation of complex process units. By using the mouse it is possible to select all required basic processes and to generate the complex model. After that one can simulate the steady state or dynamic condition of this process unit.
The following basic models are available recently:

several types of reactions
tatic and dynamic heat exchange
adsorption, desorption
cooling and heating
mixing
compression and expansion

For simulation purpose we developed interfaces to static and dynamic simulators like AspenPlus and SpeedUp.
Several examples of using this graphical environment in order to generate final complex models for number of columns and reactors are presented in the paper.

A.Schoisswohl, G.Schuster, F.Breitenecker
Technical University Vienna, Wiedner Hauptstr. 8-10, 1040 Wien, AUSTRIA

bgm

Bond Graph Modeler

1 Introduction

The program **bgm** is a project to interpret bond graph descriptions and create output for the **mosis** simulation system which was developed at the Technical University of Vienna.

The current state of the project is a processor that can handle linear bond graphs and output **mosis** model descriptions.

Further developments concern nonlinearities, fitting the model output to make use of the parallelization features of **mosis** and object orientation.

2 Bond Graphs

Bond graphs are a powerful tool for modeling physical systems, especially when they are interdisciplinary. The idea of bond graph modeling is that the power flow in a system is represented by two power variables, namely effort and flow, in any physical discipline.

The bond graph itself is a directed graph, where the different kinds of vertices ("elements") describe the relations between the two power states and the directed edges ("bonds") represent the power flow through the system. The elements can be divided active (sources), passive (consumers and storages), forwarding (transformers and gyrators) and distributing (parallel and serial junctions) elements.

The advantages of bond graph modeling are the fact that the graph still represents the structure of the physical system and that the generation of the corresponding equations from the bond graph can be done automatically. Moreover the elements of bond graphs can be expanded nearly as you like.

3 Equation Generation

As already mentioned each element of the bond graph represents a relation between the power states bound up with the attaching bonds. These relations are acausal, i.e. there is no information about cause and effect being at work at this node. An *acausal* bond graph represents a DAE[2] of the form

$$f(\dot{x}, x, t, u) = 0,$$

where x is a vector of states, t is the independent time and u is a vector of input variables.

The fact, that most simulators[3] need the system equations represented by an ODE of the form

$$\dot{x} = f(x, t, u),$$

necessitates the assignment of causality, which in fact corresponds the transformation of a DAE to an ODE[4].

This step is done automatically by **bgm** based on the state transition table method described in [2].

4 Features

The following is a brief description of the features of **bgm**:

- Simple and succinct C-like bond graph description language with elements for the use of modular development of simulation models - "graphs" and "subgraphs".
- Automatic causality assignment with the possibility of user preferences.
- Full solving of the system equations in order to get pure ODEs.
- Optionally implicit model descriptions.
- Modeling of non-linear systems.
- Definitions of "new" bond graph elements through realization as subgraphs.

References

[1] ATS and ARGESIM, Vienna, Austria. **mosis** *Users Guide*, 1.02. edition, 1994.

[2] S. J. Hood, E. R. Palmer, and P. M. Dantzig. A fast complete method for automatically assigning causality to bond graphs. *Journal of the Franklin Institute*, 326(1):83–92, 1989.

[3] P. J. Gawthrop and L. Smith. Causal augmentation of bond graphs with algebraic loops. *Journal of the Franklin Institute*, 329(2):291–303, 1992.

1. The Modular Simulation System, developed by G. Schuster at the TU Vienna; see [1].
2. Differential Algebraic Equation (DAE), in contrast to Ordinary Differential Equation (ODE).
3. Further releases of **mosis** can also solve DAEs.
4. In the case of the presence of algebraic loops the transformation is not unambiguous. Strategies for causal completion of bond graphs are given in [2] or [3].

APPLICATION OF SIMULATION SYSTEM SOL/PC FOR EDUCATION

MAGARIU G.A., TOFAN T.L., NADVODNIUC N.A., BURTSEVA L.V.

Institute of Mathematics, Moldova

The universality of the simulation method makes it an irreplaceable tool when studying a great number of disciplines connected with projection, maintenance and all-round estimation and reconstruction of complex, hardly formalized systems. Among these are: automated management systems, multiservice theory and operations research. The system for simulation training provides a professor with the following possibilities: adapted to the modern user interface, simulation models description, debugging and running, simulation models development dynamics visualization, editing and the work with the base of models and use of the type models from the base.

Let us consider the capabilities of our system, which are oriented toward the educational use.

1. **Model description language.** Simulation oriented language (SOL) originated by Knuth&McNelly was implemented and supplemented by us. Its simplicity and easy interpretation will allow the user to run fast in the language and to highlight the simulation methods being studied.

2. **Interface.** Systems for computer simulation in general must satisfy the same demands for computer systems. In spite of that they have a number of peculiarities, that can influence interface organization: from simulation process time to the great volume of information received as a result of simulation. Thus, the main problem in interface design is harmonious combination of the simulation tools and results analysis; and of the visualization of the results and simulation process.

3. **Editor and the models base maintenance.** For the creation of new model and the editing those which have been created earlier the integrated texts editor was worked out; it contains a large number of editing commands: create, save and edit models and data files, move cursor; insert, choice, copy and text delete. When editing the user can call up the list of edit options using F1. Contextual help on SOL for any language construction is called up using CtrlF1. There are information files about type models of certain courses in the models base. There are also means for the private user model base creation and addition. All information files and contextual help are written in three languages: Russian, Romanian and English (the languages used in Moldova High Schools).

4. **Display and processing of results.** The simulation results are displayed as statistics of observations for different model objects states changes. Usually they are represented in a standard form used in the statistics theory. The volume of data received as a result of simulation requires further processing to extract information which is sufficient for researcher work. We offer the following types of processing: united tables, graphical representation of the results, multicriteria graphical analysis and diagram of the model dynamic development. Simulation experiment visualization is a valuable teaching tool, allowing the user to find flaws arising between the model and the real world.

Using this system in the education process offers the students with parallelism and simulation logic which prepare them for later application of these methods in real system research. On the other hand, joining together the means of translation, analysis and visualization the system helps to explain quickly and clearly how these methods are applied.

EUROSIM
CONGRSS '95

K. Fischer, V. Hrdliczka
Institute for Industrial Engineering and Management
Swiss Federal Institute of Technology Zurich

POSTER SESSION:

Simulation in Production - new tools enter education of students at the Institute for Industrial Engineering and Management at the Swiss Federal Institute of Technology Zurich

Industrial companies are increasingly faced with new technologies and markets, rapid changes in the environment and significant sociopolitical developments. In this changing environment enterprises are forced to respond to these challenges quickly and in a flexible way.

Under an increasing time pressure it is necessary, to redesign or to adapt the structures of manufacturing systems. Decisions, which may have far-reaching consequences, have to be taken as fast as possible.

Operational procedures can be defined as processes, which are characterized by complexity and dynamics.
Discrete event-driven simulation is a powerful method to analyze and understand these processes and to solve dynamic problems.

For three years, the faculty for Industrial Engineering and Production Management at the Swiss Federal Institute of Technology Zurich has been offering a lecture in "Simulation in Production" to students and persons from industry, who are interested in these problems.
The participants are taught in theory and practice of modelling and in the application of simulation-software packages. The course has a special focus on simulation methodology i.e. project management in simulation or the theory of design, execution und evaluation of experiments.

With the aid of a case study, based on a real project in a large international company, the importance and efficiency of simulation is demonstrated to the students. It Ist shown, that without simulation it was impossible to get control of the inherent dynamic of the system and to fulfil the demands of the management. There was an extrem exceeding of the planned costs, that had could be avoided by using simulation from the very beginning.

The case study shows, that the application of simulation as early as possible may be essential to the success of a project.This will be demonstrated and discussed at the poster-session.

SPICE : The Ultimate Tool for Teaching Electrical Circuits ?

When it was introduced in the 1970's, SPICE ran on minicomputers and was intended as an aid in advanced circuit design. However, when it migrated onto desktop microcomputers such as IBM compatibles and MAC's, it effectively became available to every professional engineer. It is therefore important that, during an undergraduate course in Electronics and Electrical Engineering, students should learn the capabilities of SPICE, and appreciate how they can use it during their subsequent career.

Additionally, in the transfer to the desktop microcomputer, SPICE acquired a usefulness as a teaching tool, which it is probable that the originators of the software never envisaged. The key feature here is the graphical packages such as PROBE, which effectively allow the user to investigate the circuit under study with a software oscilloscope. SPICE with PROBE now provides facilities which make it a superb teaching tool at all levels throughout a student's course of study.

- Student can learn how to analyse a circuit **before** they have the ability to carry out the complex number algebra required for a.c. circuit analysis. The expected behaviour can be predicted from simple approximations - a capacitor blocks d.c. and passes high frequencies - and compared with the output of SPICE.
- **While** they are learning the complex number algebra, SPICE can be used to give practice in evaluating numerical values from algebraic expressions.
- **After** they have acquired the ability to carry out the mathematical work and SPICE simulation for simple circuits, they can progress to more complicated circuits. In most cases, the tedium of wading through a morass of complicated algebra is rendered unnecessary. The vital skill which should be inculcated here is that of continual questioning : does the output of SPICE agree with the simple approximations at e.g. high and low frequencies?

  Once students have thoroughly mastered SPICE and the main algebraic techniques for analysing circuits, it is an invaluable tool in circumstances where there are difficulties either with the theoretical analysis or with the experimental observations.
- SPICE provides numerical solutions where exact algebraic analysis would not normally be attempted e.g. in diode circuits and in transistor circuits.
- The study of power in typical industrial (predominantly R,L) loads is an area where it can be difficult to make experimental measurements. With PROBE it is possible to examine the overall power consumption and compare this with the power dissipated in the resistive components of the load and the energy storage and release from the inductive components.
- The observation of transients (a.c. or d.c.) traditionally requires a digital storage scope, which can be prohibitively expensive. A desktop P.C. with an analog I/O interface provides a cost-effective way to look at transients. SPICE and PROBE on the same PC allow instant comparison between theoretical predictions and experimental measurements.

Martin Macauley, Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8QQ, United Kingdom.

Leonid Novitsky

Riga Technical University & Joint Venture "LMS"

SIMULATION TRAINERS ON THE BASIS OF CAI SYSTEMS

The present poster is devoted to the problem of the development of Computer - Based Simulation Trainers for preparation of the operative personnel of automatic equipment.

Developed Simulation Trainers are based on the combination of calculating and expert models and the possibilities of Computer - Aided Instruction (CAI) systems.

A lot of functions are realized by Simulation Trainers:
- modelling the processes of controlled object;
- control of teaching dialogue;
- developing the individual skills to make a correct decisions for prevention the emergency situations appeared;
- analysis of learning process statistics etc.

The variety of functions requires to use the uniformed model for knowledges representation in Simulation Trainers. Cercone-Rieger semantic networks have been chosen for this aim.

The Simulation Trainers described in the poster realize the following key models:
- static modelling;
- dynamic modelling;
- dialogue based on the trees of situations estimation;
- checking of knowledge and developed skills.

Conclusions.

1. Simulation Trainers have been developed for preparation of operative personnel of atomic and power stations based on CAI systems.
2. The direction of the advanced Trainers evaluation is the automatization of the simulation models development and the creation of the convenient instrumentation for integration teaching and simulation models.

SERDAR BIYIKSIZ

PSU
Portland, Oregon U.S.A.

INVERSE NONLINEAR SYSTEM NOISE COMPENSATION

ABSTRACT:

The objective of this research work deals with resolving the inherent noise factor involved in nonlinear systems. Specifically the motivation is to reduce the noise embedded in signal outputs of nonlinear systems. As what will be termed as the sandwiched Hammerstein nonlinear model is employed so as to initially identify the linear and nonlinear subcomponents of the system. An optimal realizable Wiener filter is utilized to initiate an inverse filter at the input and output. Simulation of the developed algorithms provide a comparison of results using a noise reduction ratio measure.

In this problem only the noisey input and output data are known without aprior knowledge of the noise-free data. Furthermore, the assumption that both noise sources are uncorrelated is not a strict constraint. Within this framework, the nonlinear subsystem is the Hammerstein model represented in polynomial form as:

$y_n = p_1 x_n + p_2 x_n^2 + \ldots + p_m x_n^m$ where x = input, y = output ; n = 1, 2, . . ., N ; m = 1, 2, . . ., M.

Once the nonlinear parameters p and linear input and output subsystems are identified, inverse filtering is utilized by applying an optimal realizable Wiener filter at the input and output linear subsystems.

Simulation of this research is accomplished with the aid of the noise reduction ratio. This ratio is given in general form as:

$\frac{\sigma_y^2}{\sigma_x^2} = \sum_n h_n$, where h is a subsystem impulse response. Results indicate the development in this work show promising achievement of noise reduction.

SOLVING THE INVERSE CAUCHY PROBLEM OF ELECTROCARDIOLOGY

Alina Czerwińska, Marek Doros, Krystyna Kolebska, Zofia Sypniewska,
Institute of Biocybernetics and Biomedical Engineering PAS, Ks.Trojdena 4,
02-109 Warsaw, Poland

In the paper the problems concerning the appointing of epicardial maps (i.e. maps of distribution of potentials on a heart surface) by means of computer simulation have been presented. In the investigated case a source of data are Body Surface Potential Maps (BSPM) taken from human chest. The problem leads to solving the inverse boundary problem of Cauchy type. For the electrical field generated by myocardium the mathematical description is formulated as the partial differential equation of elliptical type with boundary conditions of the first and second kind. To solve this problem the new iterative algorithm based on the unitary distribution with weight parameter has been elaborated.

The numerical experiment has been carried on, in which 3D model has been applied. The examined model contains the region between the heart surface and human torso surface. The parameters concerning the shapes and locations of body organs inside the considered region have been taken from the anatomical atlas obtained by Computer Tomography (CT) method [1]. The respective conductivity parameters have been taken from Rush tables. The model obtained such a way is a subject of computer simulation.

The source of the data to the researched problem are the BSPM measured in non-invasive way on a human chest. To perform these measurements the system HP-7100 of Fukuda Denshi has been used. This system allows to measure the potentials in 87 points of a torso surface at the same time. The distances between the successive moments of measurements of BSPM's are of 1ms during the whole cycle of the heart beat.

As it was mentioned above, to solve the discussed problem the unitary distribution method with weight parameter [2] has been applied. The examined region has been divided into 273 elements. To find, how the electrical heart field changes, the calculations have been made for BSPM measured in 50 most important moments of ECG cycle.

The results of a performed computer simulation are the epicardial maps, i.e. the maps of the distribution of potentials on a heart surface. They help to find e.g. the superconductivity paths on a myocardium - it is an important information for a surgeon before the operation. These epicardial maps are presented numerically as well as graphically. The graphical presentation in a form of respective isopotential lines is being performed according to the rules discussed e.g. in [3].

The analysis of the errors of epicardial potentials, concerning the applied method, is included in [4]. The present elaboration contains the conclusions concerning the quality of construction of human chest model and accuracy of measurements of BSPM. On the basis of the obtained results of carried simulation there has been made the evaluation, how much the usefulness of applied method depends on accuracy of measurements of geometrical and conductivity parameters of chest and body organs.

REFERENCES:

1. R.S.Lendley, H.K.Huang, J.C.Mazziotta, Cross-sectional anatomy - an atlas for computerized tomography, 1988.
2. A.Czerwińska, Inverse problem of the cardiac electrical field, Syst. Anal. Model Simul. no 4, 1987.
3. M.Doros, Interactive method of medical curves compression -evaluation of compression rules, Proc. Int. Conf. Medical Physics and Biomedical Engineering, Nicosia-Cyprus, 1994.
4. A.Czerwińska, The methods of solving of inverse boundary problem of cardiology, Biocybernetics and Biomedical Engineering, v.14, no 1, 1994.

Object–Oriented Modelling and Simulation of Multipurpose Batch Plants

Konrad Wöllhaf, Martin Fritz, Christian Schulz, and Sebastian Engell

Lehrstuhl für Anlagensteuerungstechnik (Process Control Group)
Fachbereich Chemietechnik
Universität Dortmund
D–44221 Dortmund
Germany

Multipurpose batch plants are used for the economic production of chemical products in small quantities. Different products can be made at the same plant, even in parallel, at the same time. Flexible batch production, formerly mainly used in certain specialized branches of the chemical industry, e. g. the production of pharmaceutics or paints, is extended to other products like polymers and replaces continuous production processes as production to order becomes more and more important.

The versatility of such plants imposes special requirements for their control, because at the same time optimal usage of resources, safe operation of the plant and high product quality are desired. Recipe–driven production makes it possible to master these complex systems by describing the chemical and physical operations for the production of a specific product in a basic recipe, independent of any particular plant. For the actual production, the equiment of the plant has to be assigned to the operations described in the recipe and the control sequence for the execution of the production has to be generated.

This task is hardly achievable without computer support. A need exists for tools to simulate the production process and to support production planning.

From a system–theoretic point of view, batch plants have elements of both discrete–event and continuous–time systems (such systems are called hybrid systems). Simulation of flexible batch plants with (potentially) multifunctional equipment is a task which is different from — and in a sense more demanding than — the simulation of continuous–flow plants, and thus specific solutions have to be developed.

Such a tool, called BaSiP, is currently developed at the Process Control Group of the University of Dortmund. It enables the simulation of recipe-driven processes. The simulation models contain all relevant aspects of the plant and the recipe, i. e. the continuous dynamics of the equipment and the material as well as the sequential and parallel execution of operations in the recipe.

Plant and recipes can be modeled by the user with full graphics–oriented editors. By using an object–oriented approach in modelling, an extensible and configurable library of model blocks can be provided.

The tool supports various tasks in the operation and design of batch plants:

- Finding errors in recipes by simulating them on a given plant,
- checking for resource bottlenecks by simulating several recipes in parallel,
- supporting production planning by determining the execution time of operations and the usage of resources, and
- calculating the necessary capacities when designing or resizing plants.

A New Method of Multivariate Orthogonal Random Process Simulation

J. Figwer *

The problem of simulating multivariate orthogonal random processes (MORP) defined by its power spectral density matrix has been effectively solved for rational power spectral densities. Presently, multivariate time-series are simulated as outputs of discrete-time linear filters excited by white noise. This filter has to be identified using one of many spectral factorization approaches [3]. However, very often the power spectral density matrix of the time-series to be simulated is given only by a nonparametric representation as tables or diagrams.

The main idea of the proposed approach is to simulate the MORP by a multivariate multisine random series (MMRS) with a periodogram matrix coinciding with the power spectral density matrix of the MORP for a given number of equally spaced frequencies from the range $[0, 2\pi)$. The periodogram matrix is next factorized to get the Fourier transform matrix, which is via any FFT algorithm transformed into the time-domain, resulting in the simulated time-series.

The MMRS is defined in the time-domain by the p-dimensional multivariate series $\mathbf{u}^N(i)$ where the r-th ($r = 1, 2, \ldots, p$) MMRS element is given by:

$$u_r^N(i) = \sum_{\Omega n \in N_r} A_n \sin(\Omega n i + \phi_n),$$

where: $\Omega = \frac{2\pi}{N}$ denotes the fundamental relative frequency, $n = 0, 1, \ldots, \frac{N}{2}$ denotes consecutive harmonics of this frequency in the range $[0, \pi]$, i denotes consecutive discrete time instants, ϕ_n are the phase shifts of which ϕ_0 is deterministic and the remaining phase shifts are random, independent, uniformly distributed on $[0, 2\pi)$ for $n = 1, 2, \ldots, \frac{N}{2} - 1$ and Bernoulli distributed with parameter $\frac{1}{2} \in \{\alpha, \pi + \alpha\}$ for $n = \frac{N}{2}$. N_r is the set of all frequencies Ωn present in the r-th MMRS component $u_r(i)$. The sets N_r for $r = 1, 2, \ldots, p$ are pairwise disjoint and their sum gives all frequency lines up to the Nyquist frequency $\left\{0, \Omega, \ldots, \Omega\frac{N}{2}\right\}$.

When this approach is applied to the power spectral density matrix of true white noise, the corresponding MMRS is turned into a white MMRS. For $p = 1, 2$ constant bin spacings can be kept throughout the entire frequency range $[0, 2\pi)$ and whiteness holds for finite N-sample series [1], [2]. This property can't be unfortunately extended for MMRS having dimension larger than 2 or non-white MORP.

References

[1] J. Figwer A. and Niederliński. On the Generation of High Quality Scalar White Noise Series. *Applied Stochastic Models and Data Analysis*, vol. 8, pp. 311-326, 1992.

[2] Niederliński A. and J. Figwer. Using the DTF to Synthesize Bivariate Orthogonal White Noise Series. *IEEE Transactions on Signal Processing*, vol. 43, no 3, 1995.

[3] S.U Pillai and T.I.Shim. Spectrum Estimation and System Identification. *Springer-Verlag*, New York, 1993.

*Instytut Automatyki, Politechnika Śląska, Akademicka 16, PL-44-100 Gliwice, Poland, e-mail: jfigwer@appia.polsl.gliwice.pl

IMPACT MODELLING USING DELTAFUNCTION

R. Hohmann
Otto-von-Guericke University Magdeburg
Department of Simulation and Graphics

We consider the direct central impact of two masses m_1 und m_2, hitting each other with speeds v_1, v_2 at the collision time point $t = t_C$. For a short period these bodies exert big forces on each other. This results in time-dependent deformations in the environment of the contact area, which complicates dealing with the matter. Changing the state of motion during the collision can be described by means of the following idealisations:

a) The collision time Δt is so short that the change of position of the bodies involved during Δt can be neglected.
b) The forces being exerted at the area of contact are so big that all other forces can be neglected during Δt.
c) The deformations of the bodies are so small that they can be neglected with respect to the motion of the bodies as a whole (i.e. the bodies are considered as rigid due to the laws of motion).

After passing the limit $\Delta t \rightarrow 0$ the impact can be simulated as a state event, happening at the very moment. A computer-oriented, alternative approach is presented and compared with the conventional method, according to the classical solution in physics.

Conventional, piecewise modelling

The contact forces $F(t)$ do not appear in the two equations of motion. The periods of time are connected by two fitting conditions for the speeds at $t_C - 0$ and $t_C + 0$:

$$\begin{aligned} v_1' &= v_1'(v_1, v_2, m_1, m_2, e), \\ v_2' &= v_2'(v_1, v_2, m_1, m_2, e). \end{aligned} \tag{1}$$

v_1, v_2 are the speeds just before and v_1',v_2' the speeds just after the impact. The coefficient of restitution e can have the values $0 \le e \le 1$ ($e = 0$ plastic, $e = 1$ elastic). For multiple collisions these fitting conditions are to be repeatedly applied.

Computer-oriented, closed description

The impact forces $F(t)$ are involved in the equations of motion by means of Deltafunctions $\delta(t)$. During the collision time the impulse of force

$$\hat{F} = \hat{F}(v_1, v_2, m_1, m_2, e) \tag{2}$$

is exerted on both masses in opposite directions.
Finite impact forces, acting after $t = t_C$ for a short period $\Delta t = \varepsilon$, can be described by the δ_ε-Function:

$$F(t) = \hat{F} \cdot \delta_\varepsilon (t - t_c), \tag{3}$$

compression and restitution phases being differenciated, dependent on the restitution coefficient e.
Passing the limit $\varepsilon \rightarrow 0$, resulting in the abstraction of the immediate impact, can be carried out after integrating the equations of motion. Here, the δ-Function (more precisely, δ-Distribution) is replaced by the discontinuous Thetafunction $\theta(t)$ with values of 0 and 1, which can be presented without any problems. For multiple collisions, the impulses of forces are to be added.
Using the SCHEDULE-Operator and a DISCRETE-Section, the modelling of an impact is given in the all-purpose simulation system ACSL.

Symbolic Computation in Industrial Simulation Applications

Esko K. Juuso
Control Engineering Laboratory, Department of Process Engineering
University of Oulu, FIN-90570 Oulu, Finland
Esko.Juuso@oulu.fi

Computer simulation is a growing discipline which is widely applicable to fairly complex systems. However, potential application area is much wider, and futher extensions can be achieved by integrating simulation with other related fields. Understanding of real-world systems becomes more and more important when the size of applications grows. Since restrictions of simulation tools become apparent, conjoint numeric, symbolic and neural computing will be very important. Symbolic computing provides methods for connecting applications and making them easier to access for non-specialist users. There is a wide selection of these methodologies which should be adapted to appropriate levels in simulation applications.

Rule-based programming. Rule-based programming is commonly used in the development of expert systems but leads to serious maintenance and testing problems in practical applications where rule bases become really massive. Using declarative languages on relation level can reduce amount of rules to some extent. However, these simulations are still too slow for complex systems, and linking the rule-based systems to more efficient modelling methods is essential for operability of the practical systems.

Fuzzy set systems. Fuzzy set systems provide a practical way of handling qualitative information in simulation and integrating different applications on an appropriate level. Nonlinearities can easily taken into account. Fuzzy logic together with other AI methods can be used in diagnostical applications which are very valuable in improving the resemblance between the simulation models and the real-world systems. Although building small systems is quite easy, acquiring the knowledge from human experts is a tedious and time-consuming task for complex systems.

Neural networks. Neural networks (NN) have been used as a behavioural model to map a systems input to its output regardless of the nature of the system. NN models are difficult to connect to other models of the system. Another drawback is that extremely long training periods are required for complex systems. Connecting NNs to other modelling techniques is vitally important: a solution might be neurofuzzy approach which makes the model more understandable. In this framework, neural computing provides an suitable identification method for working point adaptation.

Genetic algorithms. Genetic algorithms *(GA)* can be considered as an experimenting tool which produces a satisfactory solution which is not necessarily optimal for complex systems where all the models are not known, search space is very large, and data is noisy. The algorithm is a reasonable way for processing populations of alternatives. The crucial point is that fitness functions measure the right thing, i.e. they should be application dependent. Massive parallel processing is required in real applications.

Linguistic equations. Linguistic equation approach is a new method for developing and tuning adaptive expert systems by combining different approaches and modelling methods. A wide variety of applications is based on this methodology, although some of these applications are implemented by conventional techniques. A set of linguistic relations can be changed into a compact equation, $AX = 0$, where linguistic values are replaced by integer numbers, e.g. *very_low*, *low*, *normal*, *high*, and *very_high* by numbers -2, -1, 0, 1 and 2. The direction of the interaction is represented by coefficients $A_{ij} \in \{-1, 0, 1\}$. This presentation is easily generalized for finer fuzzy partitions and transfered between the programming systems. Adaptive systems can be built if the structure of the system and the data specifying the working point conditions are separated, i.e. for fuzzy simulation rules belong to the knowledge base and membership functions to the database.

Description language. For small systems, a structural description language for symbolic computing could be based on rules since they are widely used in present systems. For complex systems, better standards for component model exchange are linguistic relations which can be used either as a list of facts corresponding linguistic relations for a Prolog program or as a database of alternatives for simulation programs with embedded SQL commands. Finally, linguistic equations provide an even more universal gateway since this framework can be implemented easily on any programming language.

Mathematical Modelling and Simulation of The Singularities in Contact Mechanics.

G.Lvov, E.Kostenko
Kharkov State Polytechnical University,
Kharkov, 310002, Ukraine

The investigation of contact interaction of beams, plates and shells with rigid bodies is very important for technical applications.Analysis of exact solutions for such problems in the simplest cases testifies that the contact reaction may include concentrated forces and bending moments along the boundary.

The kind of the singularity of reaction depends on the used theory of shells. For classical theory the solutions is possible only in class of the generalized functions and the structure of contact pressure includes δ- function .

At the employment of the shell theories with allowance for the deformations of shear, the exact solution can obtain in class of the functions having discontinuity of the first type. More complicated shell theories which describe deformation of normal to the median surface permit to realize physical smoothing and to obtain the correct mathematical problem.However such theories have not wide propagation in practice.

The contact problems for real construction are solved mainly by numerical methods. This paper deals with the investigation of numerical results in the domains of the singularity of contact reaction. Discretezation of the continuous contact problems has been carried out by the quadratic-differential approximation. The problem of non linear programming has been solved by relaxation method with projection.

Manifestation of the singularity and discontinuity in discrete models of contact problems have been determined as a result of numerical experiments.

K-CHARAKTER DIFFERENTIAL EQUATIONS IN PRIVATE DERIVATIVES AND THEIR USE FOR SIMULATION DIGITAL DEVICES

Korsunov N.I., Dmitrienko V.D., Leonov S.Yu.

Kharkov Politechnikal University
Frunze str., 21, 310002, Kharkov, Ukraine

E-mail: korsunov@polyt.kharkov.ua

Simulation with the help of K-character ordinary differential equations gives more total information about transitional processes in devices in comparison with boolean simulation and at the same time does not require the excessive volume of calculations. The description device of computer facilities with the help of ordinary K-character differential equations enables to conduct its detailed researches, however, when the studied object contains the large number of the same type components of computer facilities, such close mathematical model hinders investigations. Especially sharply this question stands at simulation of various systolical structures for processing of digital and logic data. It forces to improve known and to develop the new approaches to exact and simplified simulation.

The repeated repeatability of separate groups elements of known digital technics devices, as well as in one- , two- and threedimensional systolical matrixes and analogy in distribution information in these computing structures with distribution of heat or diffusion accordingly to uniform rods, plates or volumes induce on idea that the considered objects can be described by discrete differential equations in private derivatives of parabolic type.

The methodology of models construction and serviceability of numerical methods are checked during the research of one- and twodimension periodic computing structures, which were described by K-character analogues of heatconductivity and diffusion equations. So new theory of K-character differential equations in private derivatives which help to receive in the main new mathematical models were worked out.

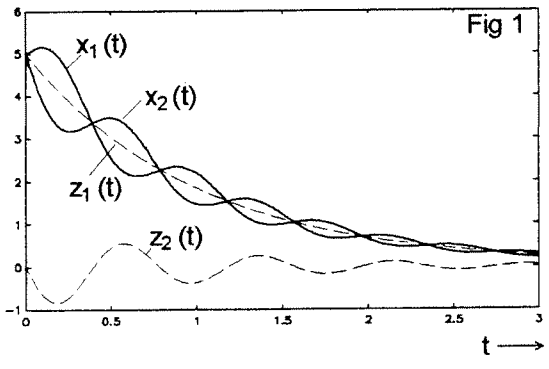
Aspects of the Dominance Analysis for Model Simplification

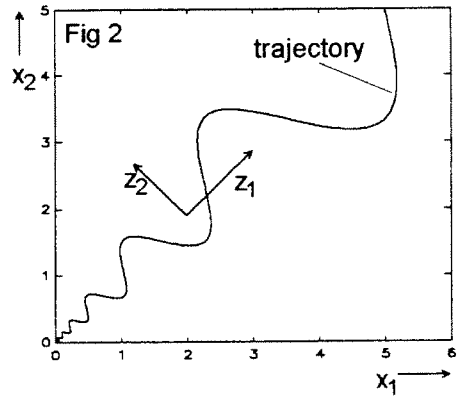
Boris Lohmann , Höhrenbergstr. 10 , D - 78476 Allensbach

A simplification of mathematical system models can be helpful for the system analysis, simulation and controller design. Different approaches are known which all aim to find a model that approximates the behaviour of the most important (dominant) variables of the original complex model. Depending on the simplification approach these dominant variables may be identified intuitively or by technical considerations. In many cases, however, a ***systematic dominance analysis*** *is required by which important and unimportant system variables can be distinguished and judged by dominance size values. This contribution presents a general approach. It is useful in different fields of system modelling and has been applied successfully to technical systems in the context of model order reduction [2-4].*

Different approaches to the simplification of dynamic systems are known and used frequently: (1) linearisation, (2) neglecting unimportant factors, terms, or part systems, (3) approximation of distributed parameter systems by ordinary differential equations, (4) approximation of dead time systems by ordinary differential equations, (5) model order reduction. Whatever method is applied, the designer should be in a position to decide which system variables are dominant and which are "contained" in the behaviour of the other variables. The required dominance analysis of system variables can be done based on a time history analysis by a singular value decomposition (which has proven to be helpful in linear model simplification also, [1]).

Figure 1 illustrates the main idea: The time paths $x_1(t)$ and $x_2(t)$ represent the behaviour of two system variables, gained by measurement or simulation. Neither x_1 nor x_2 can be classified to be dominant. For a rough description the approximation $z_1(t) \approx x_1(t) = x_2(t)$ may be sufficient. With a second function $z_2(t)$ it is possible to describe $x_1(t)$ and $x_2(t)$ precisely: in the example we have $x_2 = z_1 + z_2$ and $x_1 = z_1 - z_2$. Figure 2 illustrates the transformed coordinates and the trajectory. In the description of the dynamic system in terms of $z_1$and z_2, the dominant variable is identified easily: z_1 is dominant, since the original variables x_1 and x_2 both can be approximated from z_1. z_2 is non-dominant. A systematic approach to finding the transformation, the dominance size values and the dominant system variables is presented in the poster session and is also described in [2-4] together with application results.





[1] Moore, B.C.: Principal Component Analysis in Linear Systems: Controllability, Observability and Model Reduction. IEEE Trans. Autom. Control, AC-26 (1981), pp.17-32.

[2] Lohmann, B., Ordnungsreduktion und Dominanzanalyse nichtlinearer Systeme. Habilitationsschrift, VDI-Verlag Düsseldorf 1994.

[3] Lohmann, B., Order Reduction and Determination of Dominant State Variables of Nonlinear Systems. Mathematical Modelling of Systems, 1 (1995).

[4] Lohmann, B., Application of Model Order Reduction to a Hydropneumatic Spring Suspension. IEEE Transactions on Control Systems Technology, Vol.3, (1995) pp.102-109.

IDENTIFYING AN OBSERVABLE SYSTEM WITH ONE OF SEVERAL SIMULATION MODELS

Nicholas A. Nechval

Aviation University of Riga
Department of Applied Mathematics
Lomonosov Street 1, LV-1019 Riga, Latvia

One of the most important steps in the development of a simulation model is recognition of the simulation model which is an accurate representation of the system being studied. It is natural for model users to devise rules so as to identify an observable system with one of several distinct models, collected for simulation, which accurately represents the system, especially when decisions involving expensive resources are made on the basis of the results of the model.

This paper deals with the problem of how to identify a sample of independent observations obtained from a real system with one of several samples of observations obtained from distinct models collected for simulation, respectively. The problem considered represents, mathematically, the problem of classifying a system data sample as coming from one of several populations. Some of the information about the alternative distributions of populations has been obtained from model data samples collected for simulation. Each sample is declared to be realization of a specific stochastic process. By this step each sample is attached to just one out of a set of possible models with distinct characteristics. We are dealing with the case when the alternative distributions of populations are multivariate normal with different mean vectors and covariance matrices. It is assumed that all parameters are unknown. If it is known explicitly that a system data sample belongs to one of populations from a set of possible populations of data of models, but to which of them it belongs is unknown, the problem is to identify this sample with the proper model data population. The decision rule should be in the form of associating the system data sample with one of the model data samples and declaring that this sample came from the same population as the model data sample with which it is associated. When there is the possibility that the system data sample does not belong to one of the specified model data populations, it is desirable to determine if the assumption that this sample belongs to one of the above populations is feasible.

In the present paper, for solving the above problem, we propose to use the tests of homogeneity or normality of several independent samples in order to construct separation rules for distinguishing between model data populations into which a system data sample may be classified. This idea is new in the literature and it is due to the author. It is shown that certain tests of homogeneity or normality of several independent samples of data can be used to transform a set of model data samples into some statistic that measures distance from homogeneity or normality of these samples, respectively. Its distribution will be parameter-free or distribution-free. This statistic is then used to construct a sample based discriminant rule maximizing a distance from homogeneity or minimizing a distance from normality of several model data samples, respectively, with respect to sample data of a real, observable system.

The sample based discriminant rules, mentioned above, are applied to obtain the new procedures of identifying an observable system with one of several simulation models. These procedures are relatively simple to carry out and can be recommended in those situations when we deal with small samples of the data.

Illustration of the new procedures suggested herein is given by using a numerical example.

HYBRID MODELLING OF COMPLEX SYSTEMS AND ITS APPLICATIONS

F.Stanciulescu and V.Popa

Research Institute for Informatics
Averescu Avenue 8-10
71316 Bucharest-Romania

Abstract Poster Session

The paper describes a system designated for handling compound models based.on various types of formalisms. The architecture of the system is also presented. Different submodels sharing some state variables or parameters are coupled. More precisely, the paper addresses issues related with continuous time/discrete time submodels,discrete event submodels,fuzzy submodels,knowledge-based submodels. To describe a complex process, several types of formalisms are nedeed, as for instance developing applications based on expert systems and numerical simulation, fuzzy systems and numerical simulation , continuous time systems and discrete event systems. Given the difference between the models, coupling formalisms in implementation proves rather difficult.The hybrid model describing the behaviour of a complex system, useful in both simulation and predictive control, is presented. To make such a model operational, the following problems have been solved:

. Standard formalisms of the four mentioned models, i.e. a discrete time nonlinear equations set including interval constraints for state variables, a Boolean differential equations set, a set of fuzzy functions and a set of heuristic rules (behaviour, control and decision rules, derived from expert knowledge).

. Solving of each model problem, by using adequate computer programs: Mathcad 3.0, Matlab 4.0 for the two first models, a special program for solving fuzzy models, and an expert system shell (Clips 6.0) for solving knowledge-based models.

. Solving of the problem of interactions between various kinds of models, including their synchronization, using a special blackboard module and a hybrid algorithm of computation.

To account for the efficiency and portability of our systems, i.e. hybrid modelling, simulation , and predictive control of complex systems, an application has been proposed:a complex hydrological system of the Danube Delta, more precisely a complex network of 10 great lakes , interconnected by canals and channels.

Such problems as simulation and predictive control of (some) state variables (e.g. inflow and outflow of water, the level and the depth of water, the accumulated volume of water in the lakes, the water refreshment factor, the biomass of some species) are being solved.For example, the water refreshment factor (ecologically the most important factor) is to be found in the so-called suboptimality intervals, established by an expert A simulation experiment offers the solution to the problem of matching water refreshment factors (of all lakes) with suboptimality intervals, by predictive control. Ecologists are to pronounce as regards the results and to make their own decisions.

Finally, we emphasize that the paper develops a new approach to modelling of complex systems, using different kinds of interconnected models, and an application based on simulation experiments and predictive control.

AN INTELLIGENT MODELLING AND SIMULATION ENVIRONMENT FOR DECISION SUPPORT

Z. Strezova
MICROTEL Foundation
Sofia, Bulgaria

The development of decision support (DS) technologies is related to the integrated use of approaches, methods and tools of modelling, simulation and artificial intelligence field, aiming at the design of intelligent DS systems. In the paper are summarized the results of the development and the implementation of an intelligent modelling & simulation environment for complex problem solving support.

A framework of procedures for development of DS environment is suggested. The main phases of the DS cycle are as follows: *A. Problem Definition Phase. B. Model Development Phase. C. Model Assessment Phase. D. Decision Support Phase.*

The *Model Development Phase* requires the design of a general model of the problem stated and identification of sets of model variables.

The *Model Assessment Phase* consists in multimeasure cross-validation, assessment and selection of the set of modelled alternatives. The stages of this phase are as follows: a) simulation of the set of modelled alternatives; b) 1st validation: screening procedure; c) 2nd validation: sensitivity analysis I; d) 3rd validation: sensitivity analysis II (metamodelling).

The *Decision Support Phase* consists in analysis of a set of feasible alternatives, via hypothesing future decision states of the problem modelled, and a choice of the "best feasible" alternative for implementation.

For supporting all modelling, simulation and assessment functions required by the framework for DS a Generalized Software System (GSS) has been developed. The architecture of the GSS embodies *Data Base, Model Base, Knowledge Base* and interfacing modules.

Following the GSS architecture a problem-dependent software system SIMOS PAM has been developed.

The concepts, procedures and systems developed have been implemented in a number of policy analysis applications, in particular, in the design of DS environments for resource assignment and allocation problems.

Duplication of Mental Representation for Visional Illusion in the Rotating Space

Takeshi SETA and Ryoichi TAKAHASHI

This paper proposes the human like vision model which reconstructs three-dimensional shape from impossible figures (i.e. M.C.Escher's nonexisting world, Penrose's triangle and so on). The intuitive notion of an impossible figure seems to be summarized as follows: An impossible figure is a flat drawing that gives an impression of three-dimensional object such that the object suggested by our spatial interpretation of the drawing can not exist in the Euclidean space, that an attempt to construct it leads to geometical contradictions clearly visible to the observer. Paying attention to that human beings perceive three-dimensional objects from the impossible figure, his mental representation for visional illusion should be mathematically described. This paper indicates that three-dimensional shape of the impossible figure can exist in three-dimensional rotating space with periodic axis of depth. The rotating space is three-dimensional distance space and does not contain the time axis, from the viewpoint that human mental representation possesses whole parts of the figure concurrently. The shape can be reconstructed from the impossible figure, in spite of the constraints that one vertex on the two-dimensional image uniquely corresponds to one vertex in the three-dimensional object, and that every edges connecting vertexes is recontstructed as a geodesic representing the shortest distance in the rotating space. In the rotating space value in depth of one point on the figure is smaller than that of another point and the former is larger than the latter at the same time. Though the description "One value in depth is smaller and larger than another" is logically incorrect, it accurately expresses property of the rotating space and is mathematically correct. This property enables human mental representation for visonal illusion to be duplicated. The curvature of this space is defined as a function of difference between two values in depth found from one line by recovering the shape.

Shape recovery method is as follows. First, three-dimensional-shape of edges is reconstructed. The three-dimensional direction of an edge is formulated as a function of the inclination of the edge in the two-dimensional images. A norm is defined to evaluate the error from the correct value of the direction of edges. An assumed energy function is utilized to evaluate the orthogonality of the two edges. This function represents the deviation from right angles of the above-mentioned picture. Minimization of the weighted summation of the energy function and the norm enables us to reconstruct the three-dimensional shape of edges from two-dimensional drawings by means of the modified Powell method. Second, three-dimensional shape of faces is reconstructed with considering influence of shading on image. Authors deal with Poisson equation representing the membrane in the three-dimensional object to reconstruct the shape of faces. Furthermore, in order to allow for the influence of shade on the faces, the image-irradiate equation will be introduced. A measurement of image brightness restricts the possible face orientations. This constraint is expressed by the image-irradiate equation. The shape of the faces is recovered to minimize summation of potential energy of the Poisson equation and error norm of image-irradiate equation simultaneously by means of the Gauss-Seidel method. Boundary condition of faces is specified by position in depth determined by the reconstructed edges. Several experimental results demonstrate that the present method yields human like interpretation.

The method and language of simulation of systems for management, designing and training

Ph.D. Mikhail G. Tiagounov, Ph.D. Nicolay A. Sobolenko

(Moscow Power Engineering Institute- University of Technology,

Moscow, Russia)

Diversity of approaches to formations of structure and contents of tasks of management by technical objects, set of proposals of special packs of programs of acceptance of decisions, including on the basis of expert systems, make necessary the development of method of reliable simulation of control systems.

The necessity of maintenance of reliability of management has caused the construction of control systems, as man-machine systems. The new conditions of activity of staff of power-generating objects have required development of special systems of training and retraining of staff.

Thus was generated the wide circle of problems, on first sight poorly connected among themselves, however incorporated units of increase of reliability, controlliability and survivability of control systems is extraordinary.

The purpose of work to develop the method of structural-functional simulation of technical objects for uses of it at designing of control systems, designing and training.

For achievement of delivered purpose in work the following tasks are resolved:

- the way of coordination of various methods - from algorithm to heuristic - optimization for various levels of management in frameworks of technology of expert systems is found;
- the unified description of control system is created;
- the way of representation of knowledge about technical objects, enabling to describe the object with of given by degree detailing is found;
- the technique of automated training of experts, ensuring formal substantation of decisions on each step of their acceptance is developed.
- the software of automation structural- functional simulation of technical systems and designing information and software of management in technology of expert systems are developed.

CHAOTIC SIMULATION - A NEW KIND OF SIMULATION

WEI Qingchao, ZENG Xuegui, ZHANG Qinghang
(Civil Engineering Dept. of Northern Jiaotong University
P. O. Box 100044, Beijing, China)

We have known that the three greatest revolutions in the world-view of science in the twentieth century are Quantum Mechanics, Relativity, and Chaos. For physicists and philosophers, Quantum Mechanics and Relativity must rank above Chaos for their impact on the way we view the world, but they have less appreciable effect whatever on medicine, biology, or geology. Yet Chaos is having an important impact in all of these fields, as well as many others, including chemistry, physics, mathematics, engineering, social science, even philosophy.

In recent two decades, the scientific community has witnessed the birth and initial development of Chaos for understanding complicated and seemingly unpredictable behavior. During the same period, as computers had permeated society, computer simulation has achieved greatest development. So it is the time for us to combine Chaos and Computer Simulation together and create a new research field: Chaotic Simulation.

1. Chaos and Chaotic System

We often say observations are chaotic when there is no discernible regularity or order. A system exhibiting chaotic behaviors evolves in a deterministic way, but measurements made on the system do not allow the prediction of the state of the system even moderately far into the future.

A chaotic system possesses following distinguished features. First, it is sensitive dependence on initial conditions on invariant set. Thus, in a very real sense, chaotic systems are unpredictable. Second, it is topologic transition on invariant set. As a result, a chaotic system can not be resolved into two invariant and uncrossed open sets. Third, periodic orbits are dense on invariant set. Thus, there is a regularity in the seemingly irregularly chaotic map. The regularity is that it has the densely periodic solution.

There are many chaotic systems, such as turbulence, thermal convection in fluids, weather system, etc.

2. Simulations about Chaos

There are two kinds of simulations about chaos. The first one is Chaotic Simulation and the other one is Chaotic System Simulation.

2.1 Chaotic System Simulation

Chaotic system simulation pays more attentions to simulating natural phenomena or system's behavior. Generally speaking, we should decide whether a system is chaotic first, then study what kind of simulation model can be built up and how to simulate system's behavior. Chaotic system simulation's purpose is on application. Some chaotic system simulations are non-linear dynamic system simulations, fractal simulation, dissipation system simulation, and other chaotic system simulations.

2.2 Chaotic Simulation

Chaotic simulation focuses on simulation itself, i.e., on common properties shared by chaotic systems, on suitable simulating methods or tools, etc. Chaotic simulation also deals with integration and iteration that are common problems in simulations.

There are some methods to measure and characterize chaos, such as strange attractors, bifurcation diagrams, Poincare maps, manifolds, Lyapunov exponents, fractal dimension, power spectrum analysis, etc. They are powerful tools and useful methods for the chaotic simulations.

3. New Classification of Simulations

There are many kinds of classifications about system simulations. Now we proposed a new kind of classification in which simulations are classified from the viewpoint of mathematics.

The new classification of simulations includes deterministic simulation, probability and statistic simulation, fuzzy simulation, chaotic simulation, fractal simulation, gray simulation, etc.

Object oriented approach for modelizing discrete event systems

Liliane Muhlenbach and Serge Wendling

Laboratoire d'Informatique de Sévenans (LIS)

Jeune équipe DRED (DGRT) JE 399

Institut polytechnique de Sévenans (IPSé)

90010 Belfort Cédex - France

Tél : (033) 84.58.30.05, Fax : (033) 84.58.30.30

Simulation provides a significant aid for the design of new systems or the analysis of already existing ones. However, before running any simulation, it is necessary to work out an abstraction of the system being studied. This modelization is all the more difficult as there is no improved methodology to assist the designer in his job.

After having briefly recalled the main steps of any simulation process, we expose a method based on the object oriented approach for modelizing discrete event systems. The main objectives of our method are : (1) a logical description of the real-world, (2) a description adapted to the simulation goals, (3) a high-level of abstraction and independence of effective simulation tool, (4) a description which leads to an almost automatic elaboration of the "simulable" model.

The specifications which describe the system and clearly identify the goals of the simulation must be available before starting the modelization process. The later is performed in four steps : (1) elaboration of a structural object oriented model which is a static, structured and hierarchical description of the system, (2) working out of a dynamic model which expresses the global temporal behavior of the system, the temporal behavior of each class being a state machine (statechart diagrams), (3) construction of a communication model synthesizing the interactions between its different components and (4) elaboration of a "simulable model" which corresponds to a description from which object code can be generated and executed; here we recommend VHDL (Very high speed integrated circuits Hardware Description Language) which allows easy implementation of the concepts described above (classes, hierarchy, concurrency).

Using this approach, we describe and simulate an asynchronous numerical clock having two functions : time counting and time setting. With this example, we exhibit : (1) a top-down **analysis** process which determines the components, structure and behavior of the system in accordance with the simulation objectives and (2) a buttom-up **design** process leading to the definition of generic behaviors.

The method exposed integrates the results obtained in software engineering in order to achieve reliable, coherent and possible advanced models.

Generic Features of an Artificial Neural Network Pultrusion Process Model Developed Using Object Oriented Qualitative Analysis

David T.Wright and David J.Williams,
Department of Manufacturing Engineering,
Loughborough University of Technology, LE11 3TU, UK.

ABSTRACT

Previous extensive laboratory pultrusion trials using Derakane 440/40 thermosetting resin system and materials analysis yielded a rich data set. This poster shows the application of previous pultrusion Artificial Neural Networks (ANN's) structures with a new very different set of pultrusion data using a polyester thermoplastic resin system to create new ANN's which perform well. Within this context the authors examine the generic features and effectiveness of both sets of ANN process models. Further, a methodology linking Object Oriented and Qualitative Analysis (OOQA) techniques in relation to the collection of appropriate data and development of successful ANN structures is discussed. Further insight into process genericness was gained through mathematical deconstruction of the ANN's.

CONCLUSIONS

The main conclusions include:

* OOQA proved useful for determining data to be collected, and in structuring ANN's.
* The thermoplastic and thermosetting data was very different, but could be used to successfully create ANN models.
* The best ANN models were similar in structure for both thermoplastic and thermosetting data.
* Comparison was made in a deconstructed ANN math model and existing pultrusion math models.
* Some elements of genericness are present, and development of further ANN models for similar manufacturing processes is underway.

ACKNOWLEDGEMENTS

This research was supported by CASE award 92P/07035 from the CDP group of EPSRC in collaboration with Pera, Melton Mowbray. Thanks to Detlef Jürss of IKV, Aachen, Germany for pultrusion data and materials samples.

Modelling of combustion of coal particle in gas flow

V.I. Bykov, T.I. Vishnevskaya, N.M. Tsirul'nichenko

Computing Center of RAS, Krasnoyarsk, 660036, Russia
NIFTI, Krasnoyarsk State University, Krasnoyarsk, 660062, Russia
E-mail:bykov@cckr.krasnoyarsk.su

A modelling of dynamics of particles of dust solid fuel is an actual problem. In general case a combustion process has three basic steps: drying, pyrolysis and homogeneous heterogeneous combustion reactions and gasification reactions of flying substances and carbon mass. A mathematical modell and calculation results of dynamics of combustion process as a totality of named steps are given in the article.

Physic model. A particles of solid fuel of different dimensions (up to 5 fractions has been considered) and humidity are brought into the moving higt temperature gas stream. During acceleration of particles the following processes occur: a heating of solid phase, a cooling of gas phase, reactions of combution and gasification, drying, output of flying. Besides, a possibility of particle destruction is accounted. Mathematical model. A model of processes of heat- and mass exchange inside the porous particle hase diffusion character. Mathematical description includes a change of temperature and density of water steam and flying substances in time and on particle radius. For the gas phase the kinetic model includes an equations of chemical kinetics of conservation of basic components: $CO, O_2, CO_2, CH_4, H_2, NH_3, NO, H_2O$, HCN. For the numerical solution of diffusion-kinetic model an integration methods for stiff systems and integro-interpolation method were used.

Calculation results. The original parameters and physical and chemical conditions of process corresponds to real characteristics of installations of fuel preparing. For the various variants of poly-fractional composition of coal and its humidity a periods of drying and pyrolysis are calculated. The calculation results of reaction kinetics in gas phase allow to estimate an ecological danger of products of pyrolysis and combustion.

Modelling of bifurcations for some basic combustion theory models

V.I. Bykov, T.P. Pushkareva

Computing Center of RAS, Krasnoyarsk, 660036, Russia
E-mail:bykov@cckr.krasnoyarsk.su

A number of basic models of combustion theory, corresponding to oxidation reactions in CSTR are considered. In classic case one the models (Zeldovich-Semenov's model) is of the form:

$$\begin{aligned} \dot{x} &= f(x)e^{y/(1+\beta y)} - x/Da \\ \gamma\dot{y} &= f(x)e^{y/(1+\beta y)} - y/Se, \end{aligned} \tag{1}$$

where x, y are concentration and temperature; $f(x) = 1 - x, Da, \gamma, Se, \beta$ are dimensionless parameters. The model corresponds to one reaction $A \to B$. The models of the type (1) corresponding to different reactions

$$nA \to B;\ A + O_2 \to B;\ A_1 \to B, A_2 \to B$$

are considered in the study.

A parametric analysis of dynamic model (1) with dimensionless and real parameters and other forms of dimensionless models (Amundson-Aris, Volter-Salnikov) is carried out.

The parametric analysis of dynamic model (1) includes: analysis of steady state number; construction of dependencies on parameters; stability analysis: construction of bifurcation curves (the steady state multiplicity and neutrality curves), phase portraits and time dependencies. On the basis of specificity of the model the bifurcation curves are constructed in explicit form for the various planes of dimensionless parameters. A parametric portrait allows to estimate conditions of ignition where the system pass from one steady state to another one with high temperature. On the basis of known parametric portrait of dimensionless model a procedure for construction of ignition diagram for real conditions of process is proposed.

Modelling of utilization processes of rocket fuel components

V.I. Bykov, V.N. Podvezennyi*, V.G. Veselov*

Computing Center of RAS, Krasnoyarsk, 660036, Russia
*NPK "Kraskhimmash" ATN RF,
Krasnoyarskii rabochii, 29, Krasnoyarsk, 660123, Russia

E-mail:bykov@cckr.krasnoyarsk.su

Complex solution of the problem of the rocket-cosmos technique utilization, including fluid rocket fuel components, is an actual problem [1]. It has an important value as ecological as economic. A storage and tendency of rocket fuel accumulation require a development of new effective multi-tonnage and ecologically pure technologies of utilization. Technological lines of heptile, amile and melanges processing are projected and installated in present. A researches carried out allow to suggest methods for obtaining from the fluid rocket fuel components a wide series of products of industrial-technique purpose: thinners, extractors, inhibitors, plasticizers, hardeners and so on.

A problems of ecological control and fire-and-explosion hazard of named components utilization technology, which are toxic substances, are solved at the same time.

The mathematical modelling in rocket fuel conversion include following problems:
— mathematical modelling of chemical-technological processes of utilization;
— disign of chemical reactors for utilization;
— optimization and optimal control;
— automatization of technology for rocket fuel conversion;
— disign of complex chemical-technological schemes;
— computer monitoring of ecological restrictions and control of fire-and-explosion hazard.

References

1. V.N. Podvezennyi, Y.G. Veselov, V.I. Bykov // Iut. Aerospace Congress. Moscow: MGU, 1994, p. 78.

Dr. Tomislav Grladinović
Zagreb University, Faculty of Forestry
Production organisation
in wood industry
Svetosimunska 25, Zagreb, Croatia

Continuous Computer Simulation - A Support in Managing and Decision-Making in the Production and Business of Furniture Industry of Furniture Industry

The paper is a research on the method of predicting and optimisation of business regime and furniture production management. The method used has been the System Dynamics, a relatively young scientific discipline with its own scientific methodology of investigating the behaviour dynamics, modelling, simulation and optimisation of primarily the most complex dynamic systems that have been scientifically studied and determined by real continued models, i.e. by a group of linear and non-linear differential equations. It is also an actual application of the "System Thinking" to the processes of management of complex, dynamic, natural, technical and organization systems. This paper demonstrates the Philosophy of the System Dynamics Continuos Computer Simulation of the behaviour dynamics in the production and business management of furniture manufactures. The System-dynamic models are not based on any mass data processing but on the least number data that yield the most information on the studied laws of behaviour dynamics in the organizational business systems.
Having made a model of a furniture company, we have identified, by simulating its performance, the factors influencing decision-making and management/production processes. The simulation model has been validated through twenty different operation regimes and varying demands for the company's products. We have compared results of the simulations with the original data, showing the dynamics of a real system in the past. This comparison has shown that the model is valid, because the differences between the simulation results and the original data were small. A simulation of future management and performance of the company has shown that furniture industry would pass through the following three periods: 1. A period with high oscillations in material flows, causing a high oscillations in financial flows; 2. A period of a decreased demand for the company's products, causing a drop in the values of other elements in the material and financial flows subsystems, and 3. A period of stabilisation, with decreased oscillations in material flows and, consequently, positive effect on financial flows. Through the simulation a number of factors significantly influencing decision-making and management processes have been determined. The obtained results enabled a determination of the factors that will advance the processes of predicting and optimisations if the business regime in furniture production. The model has been successfully applied to decision-making and management processes in furniture industry. The simulation language was DYNAMO. The programming and computer simulation was made in BASIC version using the SYSDYNS software package of system dynamics, the DYNAMO language compiler in the BASIC language, enabling simulation on personal computers. The processing was carried out on the computer. The presented conclusions are an attempt of a scientific approach to prediction and a system-dynamic optimization of business regimes for improvement of this important field in furniture production and other both industrial and nonindustrial activities.

Modelling by Thermal Resistances of Asynchronous Machine Heating

M. Iordache, D.-I. Iordache, E. Demeter, N. Voicu
"Politehnica" University of Bucharest, Dept. of Electrical Engineering
Spl. Independentei no. 313, Sector 6, Cod 77 206, Bucharest, Romania
Phone/Fax: +401 312 11 90, E-mail: andrei@faraday.elth.pub.ro

The paper presents an original method for the determination of the thermal field in the asynchronous machines. The method is based on the modelling through linear or/and non-linear thermal resistances of the heating of the asynchronous machines which are outside ventilated.

The analysis of the equivalent non-linear resistive circuit associated to the asynchronous machine heating is done by using **the modified nodal analysis**. This method is associated to the original algorithm based on the **node tearing**, in which the subcircuits are torn apart along appropriate nodes. Node tearing method leads to a bordered block diagonal structure of the circuit matrix. The development of parallel processing systems for solving a number of subcircuits simultaneously leads to a reduction computation time and to an increase of the accuracy.

Corresponding to the ways of the heat transfer, the thermal resistances corresponding to the heat transfer by conduction are linear thermal resistances and the ones corresponding to the heat transfer by convection and by radiation are non-linear thermal resistances.

For computing of the thermal resistances corresponding to the heat transfer by conduction we are considered that the thermal conductivities of the materials are constants and the thermal conductivity of the stator winding presents the different values for the three directions corresponding to the three dimensional space, because the isolation and the copper have different structures in these directions.

For computing of the thermal resistances corresponding to the heat transfer by convection we are considered the following formula for the computing of the coefficient of heat transfer by radiation and convection:

$$\alpha = \alpha_0\left(1+1,3\sqrt{v}\right), \qquad (1)$$

where: $\alpha_0 = 3,49 + 0,093\,\Delta T$ in $Wm^{-2}K^{-1}$, v - is the velocity of air in ms^{-1}, and $\Delta T = T - T_a$ is the difference of the temperatures between the connective wall and the coolant fluid.

Numerical results obtained on the mathematical model are compared with the experimental results which were obtained on the physical model. The good agreement between the numerical values by the proposed method and the experimental ones confirm the validity of the proposed method.

The our method allows the testing of the great number of alternatives being useful instrument in the design and testing of the asynchronous machines.

Modified nodal analysis associated to the original node tearing, used for the analysis of the equivalent non-linear resistive circuit corresponding to the asynchronous machine heating has the advantage that the elements of the equation matrix which depend only on linear circuit element parameters remain unchangeable in the time of the iteration procedure.

Novák, Vladimír [+], Schmidt, Jan [*]

The Model of Stochastic Activity of a Neuron at Some Types of Gaussian´s Entrance Processes.

+ Department of Biokybernetics, Institute of Physiology, 1st Medical Faculty, Charles University Prague, 128 00 Albertov 5, Prague 2, ČR

* Department of Automatic Control and Informatics, Technical University of Prague, 160 00 Prague 6, ČR

Contemporary approach to the modelling of neuron dynamic properties, following both aspects of informatics and physiological aspects, have above all concentrated on the mathematical problem formulation and its corresponding solution (1,2). This approach has a series of limitng assumptions more or less removing the problem formulation, and the solution resulting from it, from experimentally found and by physiology presented behaviour of a living nervous cell. Methods of experimental physiology make possible a relatively acceptable study of the dynamic behaviour at the output of a living nervous cell enabling only a very limited study of corresponding input processes. Consequently, the principle question arises, when monitoring a dynamic behaviour of the neuron, consisting in the description of the input stochastic processes transformation, coming in the form of the input action potentials at the input neuron synapsis on the output stochastic process.

It is possible to assume that these transformation to a great degree depend not only on the properties of input stochastic processes but on the values of physiological parameters, too, examples of which are the threshold level, absolute refracter-phase duration, input synapses, weights and types and number of synapses.

In principle proposed simulation methodics is supported by already once realized stochastic neuron activity model implemented on a hybrid computer EAI 680. Applicability of at that time obtained results (3,4) leads to the forming of a new model realizead on PC and bringing a much broader spectrum of the possible experimentation. This model in contrary to a previous one has increased freedom degrees number, enabling **the study of stochastic transformations** between input and output point stochastic processes in dependence on the following physiologically significant parameters: types and stochastic properties of input stochastic processes, input excitation and inhibition synapses number, their weights, threshold level and absolute refracter-phase duration. These properties to a great degree draw nearer the model behaviour to that of a living neuron.

Flexible Robots Modelling

Joško Petrić
University of Zagreb
Faculty of Mechanical Engineering and Naval Architecture
Ivana Lučića 5
HR-41000 Zagreb

Abstract: For the purpose of better efficiency and better productivity, the efforts are made to make robot construction lighter and to increase its speed. Consequences are significant vibration of robot links. Due to that, important performance of precise positioning, repeatability and precise trajectory tracking of the robot tip are deteriorated. When robot links elasticity have significant effects on the robot characteristics and performances, then also mathematical model of robot dynamics has to include robot links elasticity. I.e. the elasticity of robot links can not be neglected any more, and the links can not be assumed as absolute rigid bodies without serious consequences on the robot performances.

This paper will present a finite element - Lagrange approach to mathematical modelling of dynamics of lightweight flexible robots. Obtained mathematical model can be used for numerical simulation of flexible robot motion, or it can be base for robot control algorithm design as well. Differently of their rigid counterparts, flexible robots are, mathematically speaking, systems with parameters distributed in space, and consequently they are described with partial differential equations (PDE) with infinite model order. For the majority of engineering purposes, mathematical model order reduction is required, i.e. model should be of finite order. Moreover it should be of as small order as possible and yet it has to describe process dynamics good enough. Therefore, finite element method (FEM) for process discretization will be used in this paper. Flexibility and facility to manipulate with various constructions are main advantages of the FEM compared with other methods which are used frequently in flexible robots modelling (for instance assumed-modes method). So, different discontinuance of the robot link's shape or material discontinuity are solved easier with FEM.

MSD A FUZZY MODEL FOR MEDICAL SIMULATION AND DIAGNOSIS

math. NICOLAE VASILESCU [1], math. M.BADICA [1], Dr. M.MUNTEANU [2]

1) Research Institute for Informatics Bucharest,
8-10 Averescu Avenu, 71316, Bucharest 1,
Fax:+40-1-3128539
ROMANIA,

2) " Marie Sklodowska Curie " Children Clinic Bucharest.

MSD - presents an original manner of treating a fuzzy model for simulation and medical diagnosis . The modelling is realized by means of a new conceived fuzzy technique and practicioner - physician reasoning, namely by fuzzy modelling the influence over them and over the final diagnosis.

MSD is a general-purpose software based on fuzzy techniques, for medical simulation and diagnosis hint.

MSD comes to assist general practitioners in offering them medical simulation possibilities and diagnosis hints.

The **MSD** system architecture covers:

Hardware : IBM PC compatible personal computers
Operating system: DOS 6.0

MSD is a software product developed by the **Research Institute for Informatics, Bucharest.**

MSD is under testing at Children's Hospital in Bucharest. It helps in simulating and diagnosing syndroms in children's pathology.

The system is available at Research Institute for Informatics - Bucharest, ROMANIA

Evaluation of queuing allocation strategies by simulation

J. Weinroth, A. Mehrez, and F. Dull
Graduate School of Management
Kent State University, Kent OH 44242, USA

In this research a simulation model was developed in GPSS/H and used to test the impact of a variety of allocation-location strategies on system throughput and service levels, without the constraints which we need to impose in mathematical analysis of this problem. The model simulates arrival and service process for any system in which two or more centers supply services to a larger number of demand points and where significant time is consumed in travel from any one demand point to any one center.

The type of system represented has a structure in which demands from some points have limited choices, between two adjacent service centers, while others have no choice. The model tests the comparative effectiveness of five control strategies with respect to assigned behavior for demands.Theses strategies are:

1. random assignment for demands which have a choice
2. selection of the shorter queue for demands which have a choice
3. cycling for demands which have a choice
4. selection of shortest queue in entire system for all demands
5. selection of shortest combined travel and estimated delay time in entire system for all demands.

System characteristics which were varied in 1600 random configurations of the model were:

1. interarrival time distribution -- exponential, normal, deterministic
2. service time distribution -- same alternatives
3. number of servers -- 1,2,3
4. rate of utilization -- 50%, 70%, 80%
5. ratio between travel time an service mean -- 50%, 100%, 200%
6. number of service centers (with corresponding number of demand points) -- 2,4,9,16

What is particulary exciting about results is that a regression analysis of the comparative effectiveness of the five strategies yielded a very high correlation invariant of the configuration of system characteristics, implying a high level of generalizability of the results. Strategy 5, which is sometimes considered to benefit the invidual but to penalize the overall sytem, was indeed more effective. However, the regression analysis shows this effectiveness to be only a 4.14% improvement over strategy 2, which would presumably be an inexpensive strategy to implement. Therefore, if the cost of implementing the more complex strategy 5 would exceed by more than this percent that of strategy 2, it would not be recommended.

Telecommunications systems constitute one example of area of application, although we would need to add the event of the signal for a service request being part of the volume of service provided in order to accomodate this application within the model.

Extending the Use of Simulation: A Manufacturing Case Study

Máire Kerrin and Andrew Greasley
School of Business, University of Derby

This paper aims to provide a practically based insight into a case study manufacturing organisation in the UK which used simulation as part of an overall change programme. The organisation manufactures a range of bogies - the supporting frame and wheel sets for rail vehicles. The approach taken is in response to calls for case study material which, rather than focusing on the tool of simulation, focuses on the problem, solutions and business benefits related to the use of the technique (Simulation Study Group 1991).

The first part of the paper will focus on the change process brought about by the need to improve performance in response to changing market conditions. Low demand in the UK market had led the company to enter new markets in Europe and the Far East. This need to compete on a global basis led the company to re-evaluate its manufacturing facility with particular emphasis on the need to increase output, reduce lead times and increase flexibility. To meet these demands, management had identified areas where substantial investment in new machinery was required. Simulation was used as part of the planning process in order to maximise the potential of the investment before implementation. The focus of the simulation study was on the technique of line-balancing which would help to ensure this was achieved. A 'balanced' system would provide one of the necessary steps in an implementation strategy for the proposed Just-In-Time (JIT) system (Schniederjans 1993, p.96). The results of the simulation study will be presented, highlighting the business benefits of using this tool within the above context.

The second part of the paper will examine how the simulation technique can be developed within the organization beyond its use as a 'one-off' tool. In addition to its traditional role as a top-down management aid, the paper will explore ways of using simulation to facilitate shop-floor involvement in the overall change process. Areas highlighted in which simulation could be utilised include training, communication and learning. Recommendations will be put forward for the future use of simulation within the UK manufacturing industry. They will be considered within the context of the move towards the 'Japanese model' of manufacturing (Schonberger 1986), which requires a greater understanding of the overall business from shop-floor workers (Womack et al. 1990). The aim of the paper is to provide practitioners with ways in which they can adapt and use simulation to better effect within their organisation.

References:

Schniederjans, M.J. 1993. Just-in-Time Management, Allyn and Bacon

Schonberger, R. 1986. World Class Manufacturing. New York, The Free Press

Simulation Study Group, 1991. Simulation in UK Manufacturing Industry, Management Consulting Group

Womack, J.P., Jones, D.T. and Roos, D. 1990. The Machine that Changed the World: The Triumph of Lean Production, New York, Rawson Macmillan

Furnace Distributed Control System

Mihaela Cosma, Florin Hartescu
Research Institute for Informatics (ICI)
8-10 Averescu Blvd. 71316 Bucharest - ROMANIA,
Fax 40-1-3128539

The paper presents a distributed software system -"Furnace"- , written on a UNIX machine, for blast-furnace charging supervision and control.

This system offers the opportunity for various kinds of equipments such as programmable logic controllers (which can be linked in a PLC-network), microprocessor controlled weighing/dosage equipments, a UNIX machine and others used in blast-furnace charging supervision and control to work together in an integrated system, taking into account the general and specific demands of the application.

The distributed software system is composed of a configuration module, a set of modules for PLC controlling - "AP-MASTER" -, a set of modules for controlling microprocessor weighing/dosage equipments DD-Master, a real-time database, a set of modules for operator-system interface, based on windows (OSF/Motif) (there are many sorts of windows: windows for alarms, for graphical representation of the evolution of the process values, for synoptical representation, etc), a report generator for reports on any measurements or important data of the process, a security module, that based on a hierarchy of passwords ensures the access to windows or database elements.

A PLC-simulator and a blast-furnace simulator have been used for testing the Furnace system off-line.

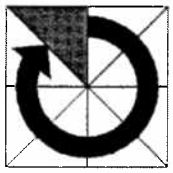
Object-oriented "fill-in-the-blank" tables permit blast-furnace charging applications to be configured without any programming and without the limitations imposed by predefined menus.

The Furnace system has a library of graphical symbols specific to blast-furnace charging process. Based on them a series of synoptical representations can be presented. The system has also a set of routines for on-line updating of synoptical representations from user program.

The Furnace system optimizes blast-furnace charging, raw-materiales use, reduces equipments non-functioning period and offers higher reliability of technological devices operation. It will be implemented in several metallurgical platforms in Romania: Calarasi, Hunedoara, Resita.

The Furnace system is an open system, other functions can be easily added, application programs can be expanded, enhanced, replaced, duplicated without affecting other programs or current application.

Author Index



Unseld + Partner

Business- and Marketing Consulting Simulation!

What we can offer our customers !

Unseld + Partner :

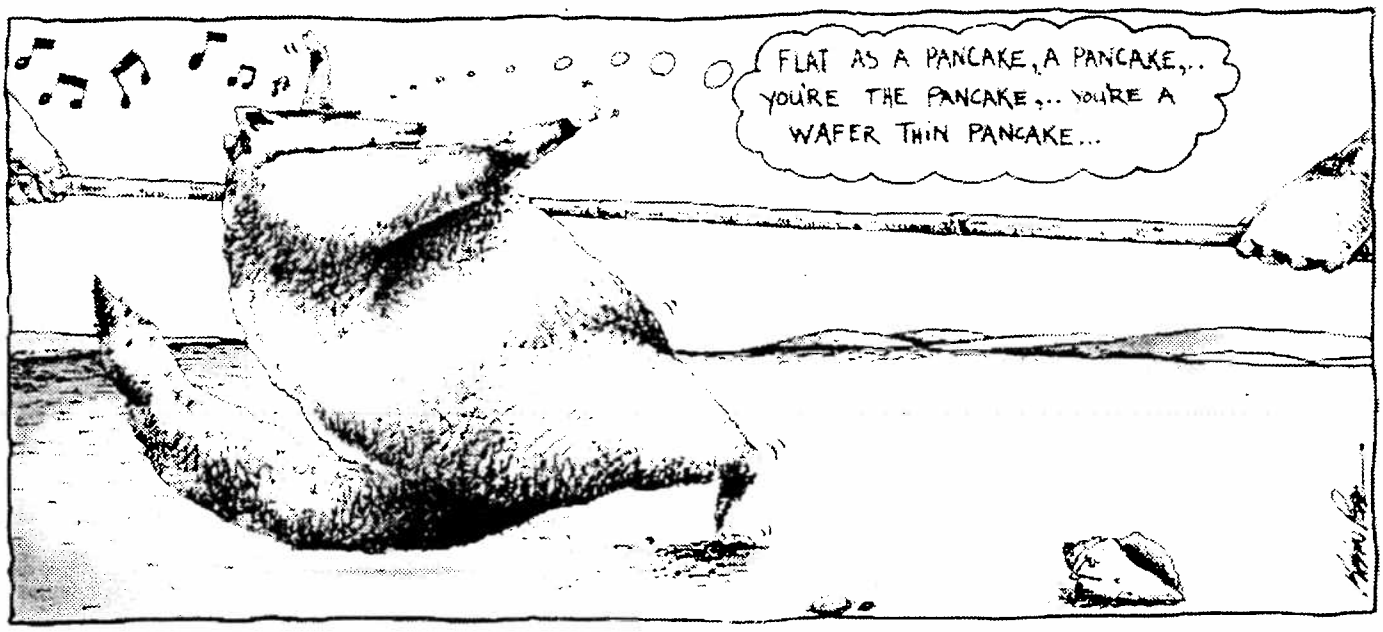
- focuses on simulation topics only and provides efficient industrial simulation experience, gained in many industrial projects
- consists of a group of highly educated simulation experts representing the highest concentration of such know-how in Austria
- offers simulation expertise in industrial projects, while being receptive for sophisticated new strategies in freight centres and intercompany logistics, tackling especially multimodal aspects.
- provides advice on EU-programs (concerning railways, IT and Telematics) to leading members of Austrian industrial organisations and government bodies
- is a completely independent Austrian company
- has co-operation contracts and contacts with many prominent international research and university institutes

Of course state-of-the-art simulation software technology for instance SIMPLE++ and VISIO are used and professional project management skills are provided.

➔ Without Simulation no Innovation ←

Unseld + Partner Business- and Marketing Consulting Simulation!
A-1080 Vienna Lerchenfelderstraße 44/9 phone +43-1-4030371-0* fax +43-1-4030371-90

THE WOLVERINE'S FLEXIBILITY CONTINUES TO AMAZE USERS.



Our discrete-event simulation and animation software handle even the most complex models

New users of **GPSS/H**™ – in industry, education and government – quickly discover how GPSS/H's superior flexibility makes simulation model development easier. Compared to GPSS/H, other packages and languages often fall short. GPSS/H won't let you down whether you're building a large, complex model or just learning about simulation.

Our **Proof Animation**™ software can bring your simulations to life. It's the next generation in PC simulation animation – faster, smoother, and more capable than the competition.

Call us today for more information or a free Proof Animation demo disk.

WOLVERINE™

Wolverine Software Corporation
7617 Little River Turnpike, Suite 900
Annandale, VA 22003-2603 USA
(800) 456-5671 (USA)
Tel: (703) 750-3910
FAX: (703) 642-9634

Proof Animation and GPSS/H are trademarks of Wolverine Software Corporation.

Readers in Germany, Austria, Switzerland (German speaking part) and Benelux contact our distributor:

scientific COMPUTERS

Scientific Computers GmbH
Franzstraße 107, 52064 Aachen
Postfach 18 65, 52020 Aachen
Germany
Tel: (0241) 26041/42
FAX: (0241) 44983