



Proceedings

Late Paper Volume

F. Breitenecker
I. Husinsky
editors

ARGESIM Report No. 1

ISBN 3-901608-01-X

© 1995 ARGESIM

ISBN 3-901608-01-X

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/](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, 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 (Proceedings *EUROSIM'95* - Session "Software Tools and Products", ISBN 3-901608-02-8).

The reviewed Poster Session completes the scientific programme. The abstracts of the 111 posters are published as ARGESIM Report (*EUROSIM'95* - Poster Book, ISBN 3-901608-03-6).

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 *Unsel + Partner* and *CA (Creditanstalt)* for sponsoring the printing of this report.

Felix Breitenacker

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. Javor (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. Merkurjev (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. Stanculescu (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. Breiteneker, I. Husinsky	3-901608-01-X
# 2	Congress EUROSIM'95 - Session Software Products and Tools	F. Breiteneker, I. Husinsky	3-901608-02-8
# 3	EUROSIM'95 - Poster Book	F. Breiteneker, I. Husinsky	3-901608-03-6
# 4	Seminar Modellbildung und Simulation - Simulation in der Didaktik	F. Breiteneker, 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. Breiteneker	3-901608-05-2
# 6	Seminar Modellbildung und Simulation -COMETT - Course "Object-Oriented Discrete Simulation"	N. Kraus, F. Breiteneker	3-901608-06-0
# 7	EUROSIM Comparison 1 - Solutions and Results	F. Breiteneker, I. Husinsky	3-901608-07-9
# 8	EUROSIM Comparison 2 - Solutions and Results	F. Breiteneker, I. Husinsky	3-901608-08-7

For information contact: ARGESIM, c/o Dept. Simulation Techniques,
 attn. F. Breiteneker, 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

Foreword	iii
Committees	v
About ARGESIM	vi
Special Lectures	
Simulation for the Future: Progress of the ESPRIT Basic Research Working Group 8467 <i>Kerckhoffs E.J.H.</i>	1
Simulation Methodology	
Choosing Simulation Software for Flexible Job Shop Environment <i>Niwinski J.</i>	3
Results of the EUROSIM Comparison "Lithium Cluster Dynamics" - Trends in Continuous Simulation Software <i>Breitenecker F., Husinsky I.</i>	9
Results and Experiences derived from a Comparison between Different Simulation Systems <i>Klußmann J., Krauth J., Splanemann R.</i>	15
Mathematical and Statistical Methods	
Systems Simulation with Nonlinear Supervisor <i>Popescu D., Cheruy A., Doboga F.</i>	21
Genetic Algorithms in Simulation	
Conservative Agents in GA-deceptive Games <i>Dawid H., Mehlmann A.</i>	27
Parallel Simulation	
The Strings that Tie Simulators together: the Message Passing Paradigm as Instantiated by PVM <i>Weisz W.</i>	33
The Real-Time Simulation of Multiframe System on Multiprocessors <i>Du T., Huang K., Zha Y.</i>	37

Distributed Interactive Simulation

- Efficiency of Parallel Strategies in Simulation - Results of the
EUROSIM Parallel Comparison
Breitenecker F., Husinsky I., Schuster G. 43

Simulation in Mechatronics and Computational Mechanics

- Real-Time Simulation and Control of an Uncertain Railway Bogie
Gajdár T., Kolumbán G. 49

- 3D-Finite Elements for the Analysis of Progressive Failure
in Laminated Composite Beams
Luciano R., Zinno R. 55

- Finite Element Analysis of Failure in Laminates
Bruno D., Porco G., Zinno R. 61

Simulation of Thermodynamic Processes

- Design Model for Boiler Heat Transfer
Lampenius B. 67

Simulation in Physics and Chemistry

- Modelling of RPSA Air Separation
Verelst H., Zhu L.-Q., Baron G.V. 73

- The Model of Radionuclides Spreading on a Vast Territory
Katkov V.L., Bakulin A.V., Kuznetsov G.P., Nichiporovich I.A. 79

Simulation in Process Engineering

- Comparison of Theoretical and Identified Models for Control of a
Pilot Plant Distillation Column
Lamanna R., Aponte F., Campos M.I. 81

Simulation of Environmental Systems

- Analysis of Dynamical and Structural Properties of a Microbial Landfill Ecosystem
by Means of Computer Simulation
Böttner R., Schleusener B. 87

Simulation in Biology and Medicine

- An Application of the Bifurcation Analysis for Determining the Conditions
for the Bursting Emergence in the Stomatogastric Ganglion Neuron Model
Dick O.E., Bedrov Y.A., Akoev G.N., Kniffki K.-D. 93

Neural Nets in Modelling and Simulation

Neural Nets: The Brain and the Computer
Rattay F., Pfützner H.

99

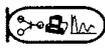
Simulation and Artificial Intelligence

The Knowledge Modelling for Human Skeleton Animation:
towards a Simulation Prototype
Fourar H., Koukam A., Deschizeaux P.

105

Author Index

111



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 Group 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 if 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:

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

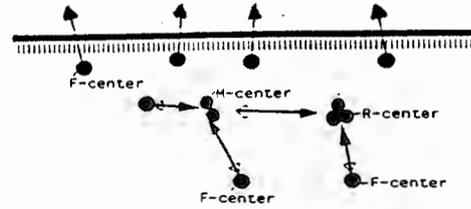


Fig. 1: Comparison1. physical background

- 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.

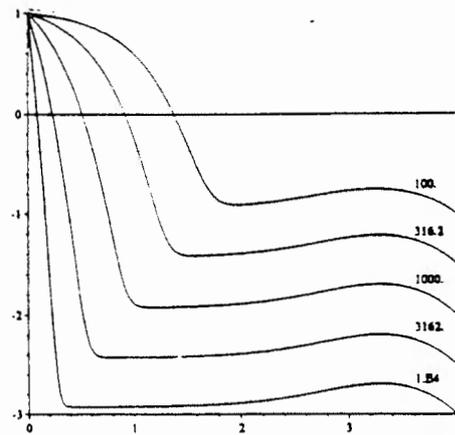


Fig.2: Results $f(t)$, variation of l_f

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 1.E-6 ae, 1.E-5 iss	0.506 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	1 (8 sec)
			Trapezoidal	mss=0.01	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	STEP SIZE 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 descr.(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. Klußmann¹, J. Krauth¹ and R. Splanemann²

Bremen Institute of Industrial Technology and Applied Work Science at the University of Bremen (BIBA)¹, Degussa AG²

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 "Eurosime 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) A_x , a buffer conveyor $B2$ of variable length in front of the station, a one-place buffer behind it, a bypass conveyor $B1$, and two connecting elements S_x and S_y . Here $B1$ is part of the inner rectangle. A pallet coming from the left can either be shifted to $B2$ in S_x 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 A_x ,
- there is enough space on the conveyor $B2$ in front of station A_x .

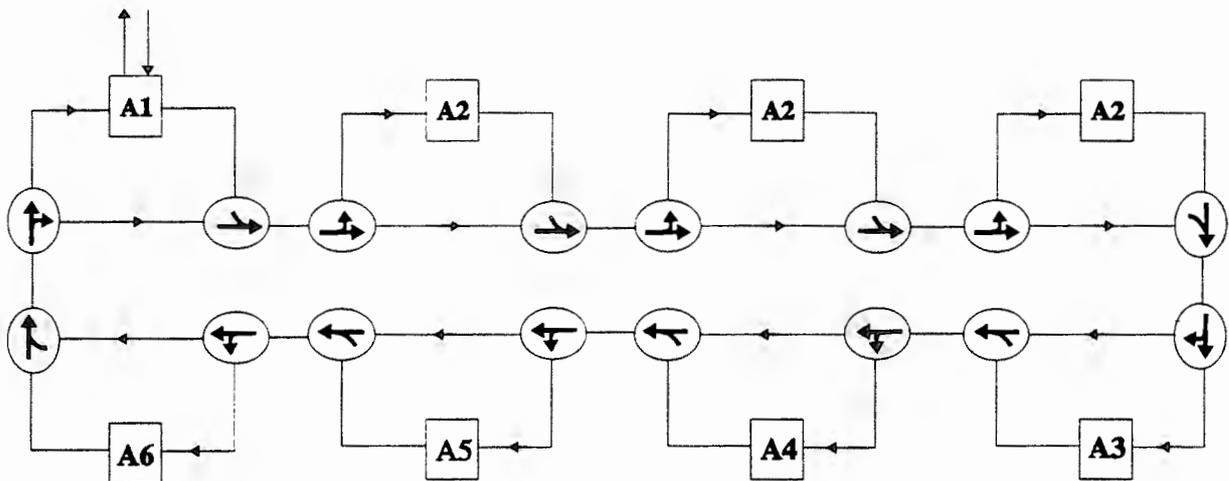


Figure 1: Flexible assembly line as test system

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 $A6$.

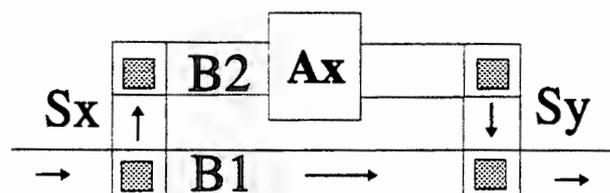


Figure 2: Subsystem of the flexible line

The processing times and the buffer sizes in front of all stations as well as the length 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 whether 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)	Unselde&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 feedback 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 predefined, 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 S_x and S_y . 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 building 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 ISTEEL 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 Simulatorenentwicklungsumgebung. In: Tavangarian, D. (Hrsg.): Fortschritte in der Simulationstechnik, Band 4. Simulationstechnik, 7. Symposium, Hagen, 1991, pp. 10-14

Systems Simulation with Nonlinear Supervisor

D. Popescu¹, A. Cheruy² and F. Doboga^{1,2}

¹Faculty of Control and Computer Science, U.P.B., Splaiul Independentei 313, Bucuresti 6, Romania

²Laboratoire d'Automatique du Grenoble, I.N.P.G., BP 46, 38402 St.Martin d'Herès, 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 represented 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 admissible 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 considered by a nonlinear model:

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

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

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

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

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

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

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

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

$$d^2(y, y_k) = (y - y_k)^T A (y - y_k) \quad (8)$$

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

$$I(y) \cong I(y_k) + (\Delta y)^T GI(y_k) \quad (9)$$

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

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

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

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

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

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

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

With (12) it can be written

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

and from this

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

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

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

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

$$\|y_{k+1} - y_k\| < \varepsilon_1 \quad (17)$$

or

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

or

$$\|GI(y_{k+1})\| \leq \varepsilon_3 \quad (19)$$

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

$$\lim_{k \rightarrow +\infty} \frac{\|y_{k+1} - y^*\|}{\|y_k - y^*\|^p} = c \quad (20)$$

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

$$c \in (0, 1), \quad p = 1 \implies \text{linear convergence}$$

$$c \in (0, 1), \quad p > 1 \implies \text{superlinear convergence} \quad (21)$$

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 \mathbb{R}^n then a possible research strategy is given by

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

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

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

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

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

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

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

with A constant matrix and $b(\omega)$ a random variable, or

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

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

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

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:

$$m^T = [E\{\theta_i\}]^T \quad (28)$$

and the covariance matrix is

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

The problem

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

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

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

where

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

is the distribution function for z.

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

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

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

Imposed Risk Condition

This condition is defined by

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

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

$$\begin{aligned} \alpha_0 = p\{I(y) < I'\} &= p\left\{ \frac{\hat{\Theta}^T(\omega)f(y) - m^T f(y)}{\sqrt{y^T V y}} < \frac{I' - m^T f(y)}{\sqrt{y^T V y}} \right\} = \\ &= \Phi\left\{ \frac{I' - m^T f(y)}{\sqrt{y^T V y}} \right\} \end{aligned} \quad (35)$$

$$\frac{I' - m^T f(y)}{\sqrt{y^T V y}} = d \implies I' = m^T f(y) + d\sqrt{y^T V y} \quad (36)$$

Now the problem (34) is written

$$\max_{y \in D_{adm}(\omega)} \left\{ I' = m^T f(y) + d\sqrt{y^T V y} \right\} \quad (37)$$

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 $r_1=1400 \text{ m}^3/\text{h}$, $r_2=450 \text{ m}^3/\text{h}$, $r_3=4 \text{ bar}$ and $r_4=800 \text{ }^\circ\text{C}$. 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

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

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

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

The quality condition is given by the ethylene production an hour

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

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

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

The result is

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

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

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

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

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 $s : \Omega \rightarrow \Delta^m$ 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_k^i is the relative frequency of the string $k \in \Omega$ in P^i). We call the strategy $s^i \in \Delta^m$ with

$$\bar{s}^i = \sum_{k \in \Omega} \phi_k^i s(k) \quad (1)$$

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 :

$$f_k^i(\phi^j) = s(k)' A s^j \quad k \in \Omega, \quad i, j \in \{1, 2\}, \quad i \neq j. \quad (2)$$

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 assumptions 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

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

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

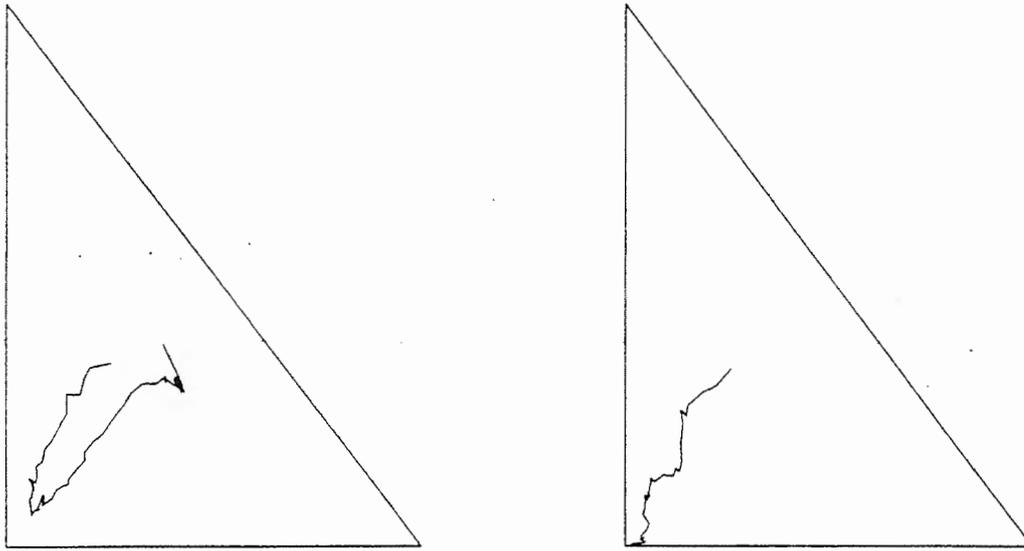


Figure 1. The population strategies s^1 and s^2 in a simulation with a GA with parameter values $n = 100$, $l = 12$, $\chi = 1$ and $\mu = 0$.

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 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

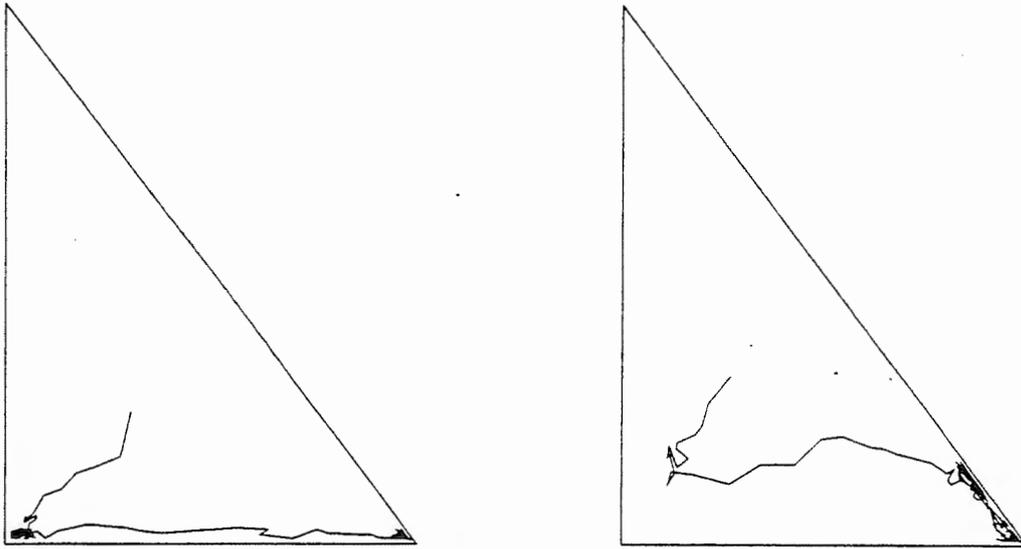


Figure 2. The population strategies \bar{s}^1 and \bar{s}^2 in a simulation with a GA with parameter values $n = 100$, $l = 12$, $\lambda = 1$, $\mu = 0$ and one conservative agent for each pure strategy in both populations.

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

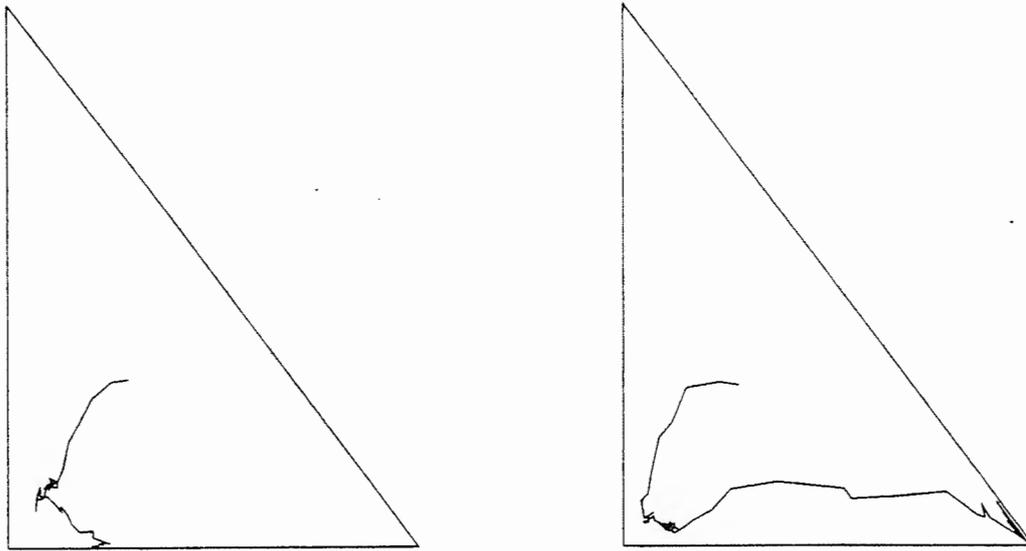


Figure 3. The population strategies s^1 and s^2 in a simulation with a GA with parameter values $n = 100$, $l = 12$, $\chi = 1$, $\mu = 0$ and one conservative agent for each pure strategy in the second population.

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 *pvm*s) 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 its 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 *receiving* 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 *pvm*.

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 too tightly coupled. If there is too much message exchange the communication will dominate over the computing time, and it is several orders of magnitude 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 *tids* 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 whereabouts 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 *pvm* 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 its 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

^aDept. of Computer Science, ^bDept. of Automatic Control
National University of Defense Technology
Changsha, Hunan, P. R. of China (410073)

Abstract

The principles of processor allocation for the parallel 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, $X = \{x_1, \dots, x_n\}$ is a set of state variables, $f = \{f_1, \dots, f_n\}$ 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 $X_F = \{x_1, \dots, x_i\}$ ($1 < i < n$), the state equations are

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

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

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

Therefore, M can also be denoted by

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

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

```

/* t0 is the starting time of simulation, tmax is the finish time,
   X(0) is the initial conditions of X at time t0,
   X(N) is the values of X at time tN, tN=t0+N*tS (N=0,1,2, ...) */
for t=t0 to tmax step tS
  begin
    calculat-XS(X(N), XS(N+1), tS) /* calculate XS(N+1) via X(N) */
    calculat-XF(X(N), XF(N+1/R), tF) /* calculat XF(N+1/R) via X(N) */
    for i=1 to R-1 step 1
      begin
        interpolation(XS(N), XS(N+1), i, X̄S(N+i/R))
        /* interpolation between XS(N) and XS(N+1) to obtain the
           approximate value X̄S(N+i/R) of XS at time tN+i/R */
        calculat-XF(X̄S(N+i/R) ∪ XF(N+i/R), XF(N+(i+1)/R), tF)
      end
    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 $X_F(N+i/R)$, $i=1, 2, \dots, R$. 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:

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

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, computed in 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

$$C_p = k_1(m_1-1) + k_2(m-m_1-1)$$

$$C_s = k_1(m-1) + k_2(m-1)$$

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

$$\frac{C_s}{C_p} = \frac{(k_1+k_2)(m-1)}{k_1(m_1-1) + k_2(m-m_1-1)}$$

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

$$\frac{C_s}{C_p} = \frac{2(m-1)}{m-2} > 2$$

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

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

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, $X_F(N+i/R)$ ($i=1, 2, \dots, R$) 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 $X_S(N+i/R)$ ($i=1, 2, \dots, R-1$) 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=t0 to tmax step tS
  par-begin /* solve the two subsystem concurrently */
  begin
    calculat-XS(X(N), XS(N+1), tS) /* computer XS(N+1) via X(N) */
  end
  begin
    for i=0 to R-1 step 1
      begin
        extrapolation(X'S(N), XS(N), i, XS(N+i/R))
        /* extrapolation based on X'S(N) and XS(N) to obtain the
           approximate value XS(N+i/R) */
        calculat-XF(XS(N+i/R) ∪ XF(N+i/R), XF(N+(i+1)/R), tF)
      end
    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:

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

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, and the 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.

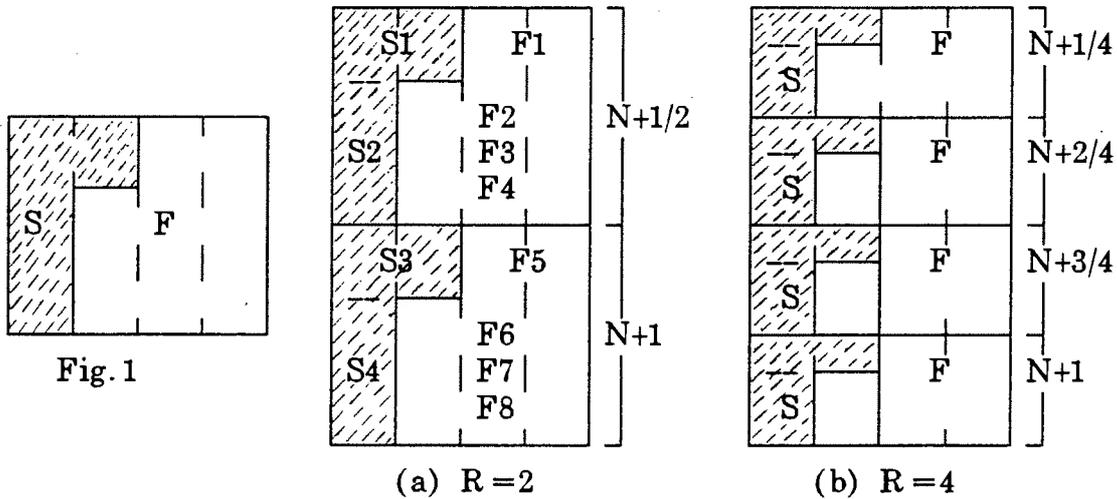


Fig. 2

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

S is divided equally into 4 parts (S_1, S_2, S_3, S_4) according to its execution order, and F into 8 parts (F_1, F_2, \dots, F_8). The codes of F_5, F_6, F_7, F_8 are identical with those for F_1, F_2, F_3, F_4 respectively. The execution flow of S is: S_1 is executed on P_1 and P_2 (P_i is the i th processor) concurrently at first, secondly, S_2 is executed on P_1 , and then S_3 is executed on P_1 and P_2 , at last S_4 is executed on P_1 .

The execution flow of F is: F_1 is executed on P_3 and P_4 concurrently at first, secondly, f_2, f_3, f_4 are successively executed on P_2, P_3, P_4 simultaneously, and then F_5 is executed on P_3 and P_4 , at last F_6, F_7, F_8 are executed on P_2, P_3, P_4 .

Both some code of S and some code of F are executed on P_2 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 become much more complicated which will bring 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" is not 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
     $m_S = \text{INT}(m_S + 0.5)$ ,  $m_F = m - m_S$ 
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 $Q_S \ll Q_F$ or $Q_F \ll Q_S$, 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 parallel 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 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 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 | • frequency domain | • statistical processors |
| • event handling | • plot features | • control strategies |
| • submodel features | • parameter sweep | • optimization |
| • numerical integration | • postprocessing | • random numbers |
| • steady-state calculation | • statistical features | • 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

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

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).

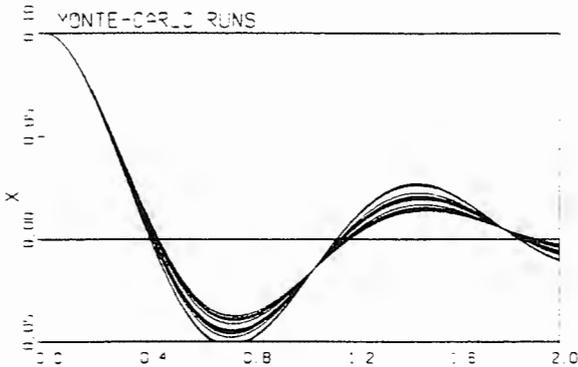


Fig.1: TE-1: Results of Monte Carlo Study

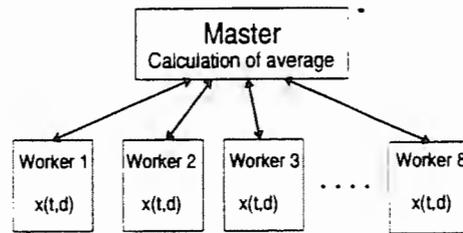


Fig.2: TE-1: Hierarchical structure of tasks

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):

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

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.

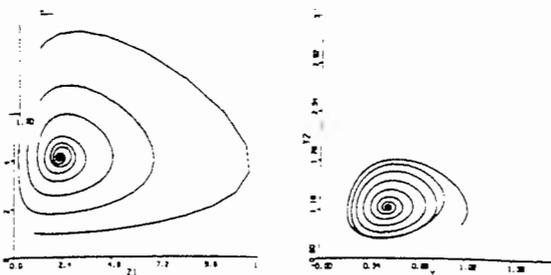


Fig.3: TE-2: Some simulation results

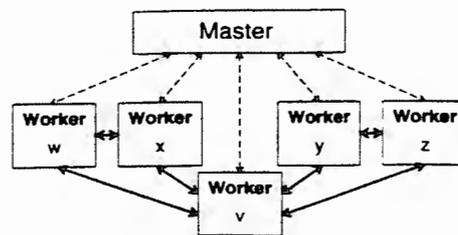


Fig.4: TE-2: Strongly coupled task structure

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.

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

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

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

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

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 structure of parallel tasks, if M different processors calculate N/M subsequent lines.

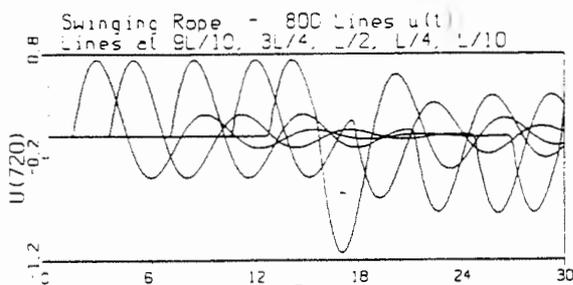


Fig.5: TE-3: Results for some lines

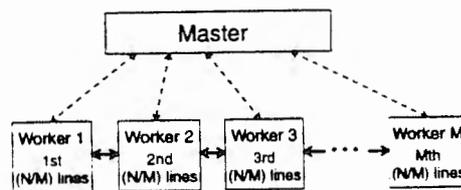


Fig.6: TE-3: Weakly coupled task structure

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.

SOLUTION	MONTE-CARLO-STUDY (TE-1)	PREDATOR-PREY SYSTEM (TE-2)	PARTIAL DIFFERENTIAL EQUATION (TE-3)																														
Workstation Cluster / PVM with FORTAN, C)	$P = 8 (9), S=2, f = 4,9$	<table border="1"> <tr><th>P=5 (6)</th><th>h</th><th>5h</th><th>10h</th><th>20h</th></tr> <tr><td>S=2</td><td>0,05</td><td>0,21</td><td>0,39</td><td>0,77</td></tr> </table>	P=5 (6)	h	5h	10h	20h	S=2	0,05	0,21	0,39	0,77	<table border="1"> <tr><th>P = 8 (9)</th><th>h</th><th>2h</th><th>4h</th><th>8h</th></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>	P = 8 (9)	h	2h	4h	8h	N=600,S=125	0,59	1,10	1,9	3,1	N=800,S=200	0,72	1,37	2,05	3,82	N=1000,S=250	0,93	2,05	3,10	4,30
P=5 (6)	h	5h	10h	20h																													
S=2	0,05	0,21	0,39	0,77																													
P = 8 (9)	h	2h	4h	8h																													
N=600,S=125	0,59	1,10	1,9	3,1																													
N=800,S=200	0,72	1,37	2,05	3,82																													
N=1000,S=250	0,93	2,05	3,10	4,30																													
Pasytec Cluster / SLIM	<table border="1"> <tr><th>P</th><th>2 (3)</th><th>8 (9)</th><th>16 (17)</th></tr> <tr><td>f</td><td>1,99</td><td>7,71</td><td>14,57</td></tr> </table>	P	2 (3)	8 (9)	16 (17)	f	1,99	7,71	14,57																								
P	2 (3)	8 (9)	16 (17)																														
f	1,99	7,71	14,57																														
Cogent XTM / mosis	$P = 8 (9), S=2, f = 4,4 (6,2)$	<table border="1"> <tr><th>P=5 (6)</th><th>h</th><th>5h</th><th>10h</th><th>20h</th></tr> <tr><td>S=2</td><td>0,06</td><td>0,19</td><td>0,30</td><td>0,61</td></tr> </table>	P=5 (6)	h	5h	10h	20h	S=2	0,06	0,19	0,30	0,61	<table border="1"> <tr><th>P = 8 (9)</th><th>h</th><th>2h</th><th>4h</th><th>8h</th></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>	P = 8 (9)	h	2h	4h	8h	N=800,S=200	4,33	4,56	5,64	6,04										
P=5 (6)	h	5h	10h	20h																													
S=2	0,06	0,19	0,30	0,61																													
P = 8 (9)	h	2h	4h	8h																													
N=800,S=200	4,33	4,56	5,64	6,04																													
Heterogenous MP-System / FSIMUL_P	$P = 4 (5), S = 2, f = 3,67$	<table border="1"> <tr><th>P</th><th>S</th><th>h</th><th>2h</th><th>5h</th><th>10h</th></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>	P	S	h	2h	5h	10h	5 (6)	2	0,38	0,38	0,83	1,32	3 (4)	2/4	0,14	0,27	0,62	1,06	$P = 4 (5), N = 800, f = 3,85$												
P	S	h	2h	5h	10h																												
5 (6)	2	0,38	0,38	0,83	1,32																												
3 (4)	2/4	0,14	0,27	0,62	1,06																												
Cogent XTM / Linda („C“)	$P = 8 (9), S=2, f = 7,8$	<table border="1"> <tr><th>P=5 (6)</th><th>h</th><th>5h</th><th>10h</th><th>20h</th></tr> <tr><td>S=2</td><td>0,08</td><td>0,29</td><td>0,60</td><td>1,20</td></tr> </table>	P=5 (6)	h	5h	10h	20h	S=2	0,08	0,29	0,60	1,20	<table border="1"> <tr><th>P = 8 (9)</th><th>h</th><th>2h</th><th>4h</th><th>8h</th></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>	P = 8 (9)	h	2h	4h	8h	N=600,S=125	6,78	7,12	7,54	7,61	N=800,S=200	6,88	7,21	7,62	7,70					
P=5 (6)	h	5h	10h	20h																													
S=2	0,08	0,29	0,60	1,20																													
P = 8 (9)	h	2h	4h	8h																													
N=600,S=125	6,78	7,12	7,54	7,61																													
N=800,S=200	6,88	7,21	7,62	7,70																													
Cogent XTM / SIMUL_R PARALLEL	<table border="1"> <tr><th>P</th><th>2 (3)</th><th>8 (9)</th><th>16 (17)</th></tr> <tr><td>f</td><td>1,93</td><td>3,37</td><td>9,99</td></tr> </table>	P	2 (3)	8 (9)	16 (17)	f	1,93	3,37	9,99	$P = 5 (6), S = 2, f = 0,04$	<table border="1"> <tr><th>P</th><th>1 (2)</th><th>2 (3)</th><th>4 (5)</th><th>8 (9)</th></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>	P	1 (2)	2 (3)	4 (5)	8 (9)	N=800,S=200	1	1,79	2,75	2,35												
P	2 (3)	8 (9)	16 (17)																														
f	1,93	3,37	9,99																														
P	1 (2)	2 (3)	4 (5)	8 (9)																													
N=800,S=200	1	1,79	2,75	2,35																													
Workstation Cluster / MATLAB - PSI	<table border="1"> <tr><th>P</th><th>2 (3)</th><th>5 (6)</th><th>10 (11)</th></tr> <tr><td>f</td><td>2,00</td><td>4,99</td><td>9,92</td></tr> </table>	P	2 (3)	5 (6)	10 (11)	f	2,00	4,99	9,92	<table border="1"> <tr><th>P=5 (6)</th><th>h</th><th>2h</th><th>5h</th><th>10h</th></tr> <tr><td>S=2</td><td>0,70</td><td>0,98</td><td>1,90</td><td>2,77</td></tr> </table>	P=5 (6)	h	2h	5h	10h	S=2	0,70	0,98	1,90	2,77	<table border="1"> <tr><th>P = 8 (9)</th><th>h</th><th>2h</th><th>4h</th><th>8h</th></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>	P = 8 (9)	h	2h	4h	8h	N=800,S=200	5,01	5,71	6,24	6,54		
P	2 (3)	5 (6)	10 (11)																														
f	2,00	4,99	9,92																														
P=5 (6)	h	2h	5h	10h																													
S=2	0,70	0,98	1,90	2,77																													
P = 8 (9)	h	2h	4h	8h																													
N=800,S=200	5,01	5,71	6,24	6,54																													

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 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://eurosimsim.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.

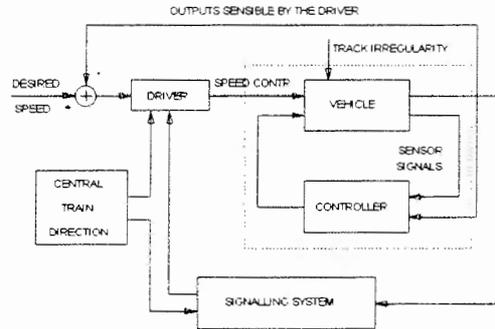


Figure 1. Complex model of vehicle-controller-driver and its environment

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 generalized equations of motion of a linear wheelset and bogie model without to going into details can be found in literature [4,6].

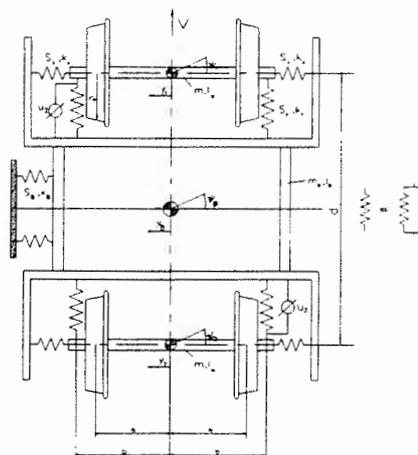


Figure 2. Linear railroad bogie model

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

$$M\ddot{\mathbf{x}}(t) = -K\dot{\mathbf{x}}(t) - S\mathbf{x}(t) + D_{\delta_1}\delta_1(t) + D_{\delta_2}\delta_2(t - \frac{d}{V}) + C\mathbf{u}_2, \quad (1)$$

The state vector, \mathbf{x} , of the bogie model can be expressed as follows:

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

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

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B_1\mathbf{u}_1(t) + B_2\mathbf{u}_2(t). \quad (2)$$

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:

$$y_1(t) = C_1\mathbf{x}(t) + D_{11}\mathbf{u}_1(t) + D_{12}\mathbf{u}_2(t). \quad (3)$$

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:

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

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:

$$\dot{\mathbf{x}} = A_o\mathbf{x} + L\Delta N^T\mathbf{x} + B_1\mathbf{u}_1 + B_2\mathbf{u}_2, \quad (6)$$

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:

$$J(\mathbf{u}_1^*, \mathbf{d}^*, \mathbf{u}_2^*, \lambda, \gamma) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [\mathbf{x}^T (\mathbf{Q}_o + \lambda \mathbf{N}\mathbf{N}^T) \mathbf{x} + \mathbf{u}_2^T \mathbf{R}_o \mathbf{u}_2 - \gamma^2 \mathbf{u}_1^T \mathbf{u}_1 - \lambda \mathbf{d}^T \mathbf{d}] dt \quad (7)$$

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

$$\mathbf{u}_2(t, \gamma, \lambda) = -R_o^{-1} B_2^T X_{H_\infty/RLQR}(\gamma, \lambda) \mathbf{x}(t), \quad (8)$$

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

$$\begin{aligned} & X_{H_\infty/RLQR} A_o + A_o^T X_{H_\infty/RLQR} + (\mathbf{Q}_o + \lambda \mathbf{N}\mathbf{N}^T) + \\ & + X_{H_\infty/RLQR} \left(\frac{1}{\gamma^2} B_1 B_1^T + \frac{1}{\lambda} L L^T - B_2 R_o^{-1} B_2^T \right) X_{H_\infty/RLQR} = 0. \end{aligned} \quad (9)$$

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,

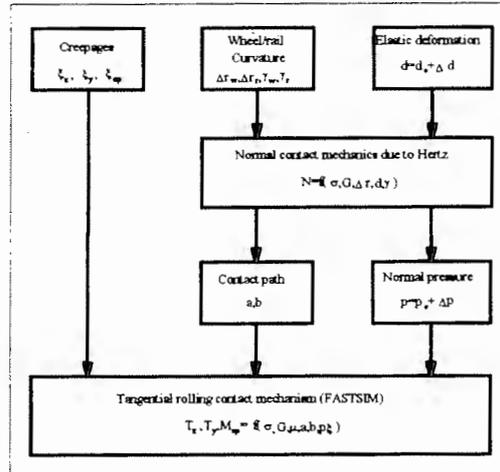


Figure 4. Nonlinear contact mechanics

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 ($\gamma=50$ and $\lambda=850$). 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).

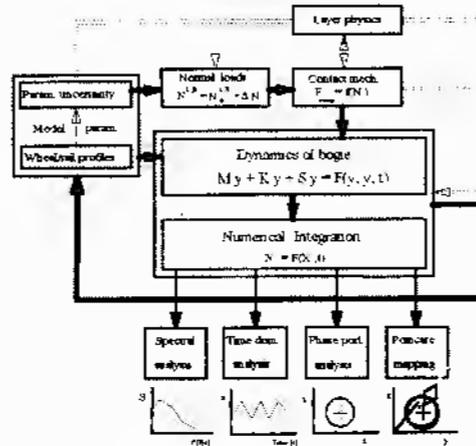


Figure 5. Method of dynamic analysis

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 much smaller and no wheel/rail contact occur.

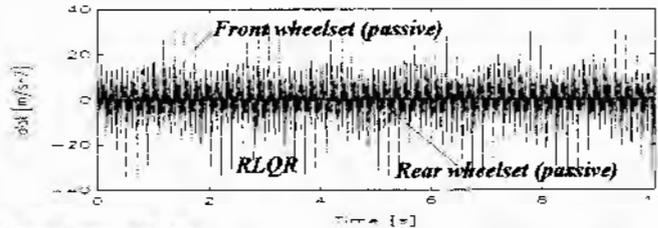


Figure 6. Lateral accelerations of the wheelsets of bogie

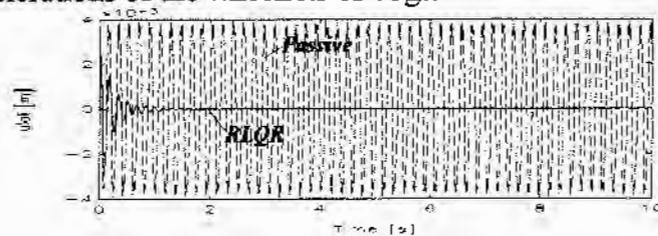


Figure 7. Lateral displacements of front wheelset of bogie

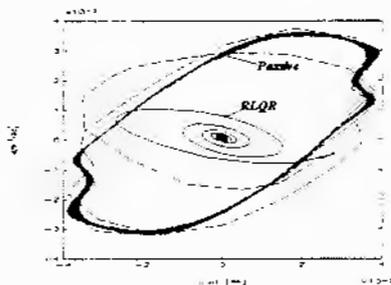


Figure 8. Phase plane plot of front wheelset at $V=190$ km/h

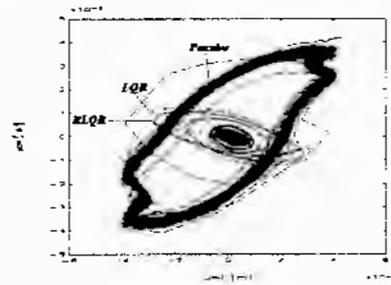


Figure 9. Phase plane plot of front wheelset at $V=300$ km/h

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.

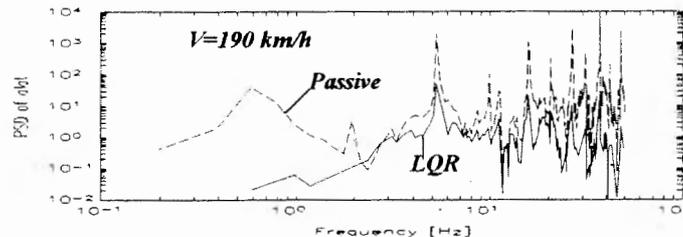


Figure 10. Lateral acceleration spectrum of first wheelset

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 occurrence 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:

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

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 , $i = 1, \dots, 6$) as follows:

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

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 $\sigma_4 (= \sigma_{23})$, $\sigma_5 (= \sigma_{13})$ and $\sigma_6 (= \sigma_{12})$ 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° 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, constituent 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:

$$\{\varepsilon\} = [S]\{\sigma\}; \quad (3)$$

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:

$$\Pi = \frac{1}{2} \int_V \{\sigma\}\{\varepsilon\} - \int_{S_b} \{t\}\{u\} dS_b = -\frac{1}{2} V \{\bar{\sigma}\}[S]\{\bar{\sigma}\}; \quad (4)$$

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:

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

where $[\mathbf{S}(\bar{\sigma})]$ is the effective final compliance tensor of the cracked solid and it depends on the value of the actual stress field $\{\bar{\sigma}\}$. 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:

$$\Pi_c - \Pi = E_r = \frac{1}{2}f_s\{\bar{\sigma}\}\Gamma^s\{\bar{\sigma}\}. \quad (6)$$

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

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

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

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

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

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

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

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

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

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:

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

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

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

It is worth to note that, in eqns. (7) ÷ (12), the matrix $[\mathbf{S}]$ 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:

$$\{\mathbf{u}\} = [\mathbf{N}]\{\mathbf{u}_i^e\}; \quad \{\mathbf{x}\} = [\mathbf{N}]\{\mathbf{x}_i^e\}; \quad (13)$$

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:

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

where $[\mathbf{B}]$ is the linear strain-displacement matrix which connects the strains $\{\epsilon\}$ 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:

$$\{\bar{\sigma}_{123}\} = [C_{123}(\bar{\sigma})]\{\epsilon_{123}\}; \quad \{\bar{\sigma}_{xyz}\} = [C_{xyz}(\bar{\sigma})]\{\epsilon_{xyz}\}; \quad (15a, b)$$

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].

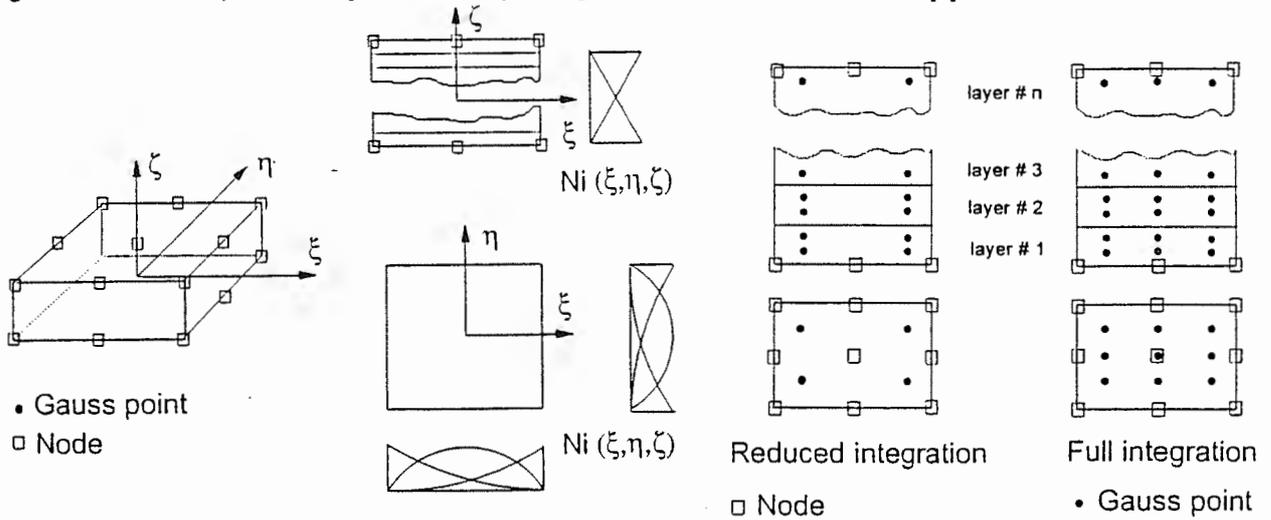


Fig. 1 - 3DLCS element and position of the Gauss points in a cluster of laminae.

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

$$[\mathbf{K}^*] = \int_{V_e} [\mathbf{B}]^T [C_{xyz}(\bar{\sigma})] [\mathbf{B}] dV_e. \quad (16)$$

By using one element for each layer (or cluster of laminae) we will omit the statement that normals remain straight 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]:

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

where $[\bar{\mathbf{K}}^e(\bar{\sigma})]$ and $\{\bar{F}^e(\bar{\sigma})\}$ 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 transform 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 tolerance (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

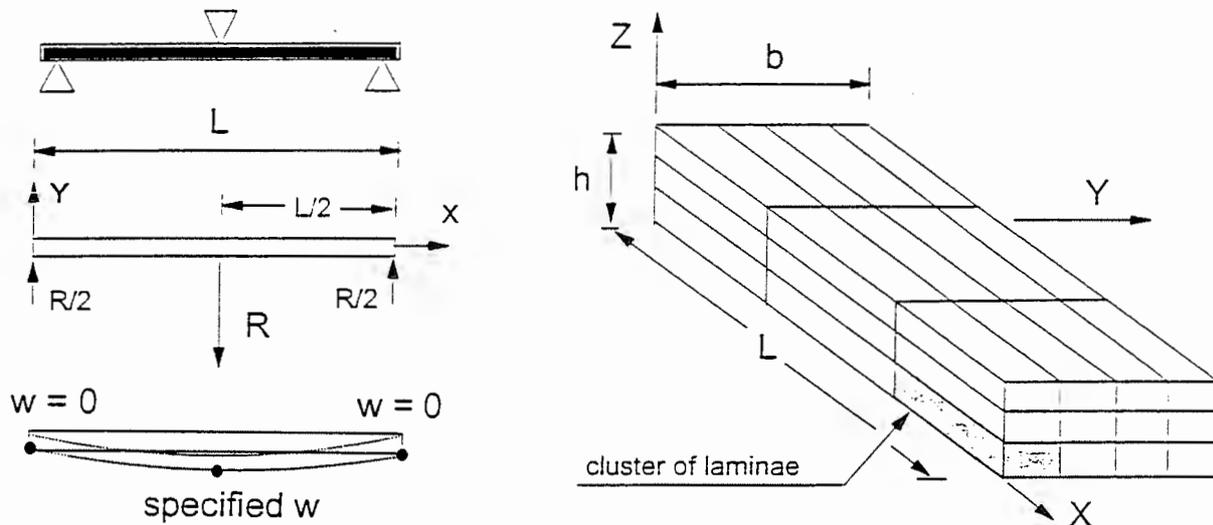


Fig. 2: Three-point bending scheme and typical 3D-finite element mesh

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:

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

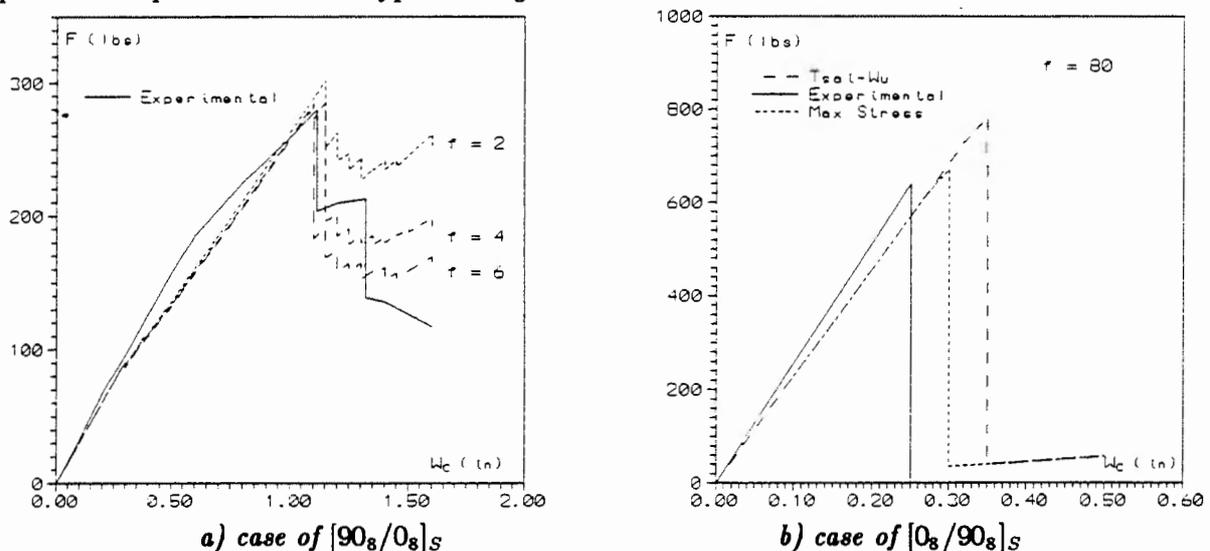
First a specimen with span of 6 inches, width of 1.01 inches and total thickness of the laminate equal to 0.179 inches, with a $[90_8/0_8]_S$ layup was analyzed. The load configuration and the boundary condition adopted are schematized in Fig. 2. In particular the beam was discretized by a $5 \times 2 \times 4$ 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° layers are broken the internal 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 occurrence 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.



Figg. 3a, 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]:

$$\begin{aligned} u &= u^0 + z(\gamma_x - w_{,z}^0) + \sum_{k=1}^{n-1} \psi_k(z - z_k)Y(z - z_k); \\ v &= v^0 + z(\gamma_y - w_{,y}^0) + \sum_{k=1}^{n-1} \chi_k(z - z_k)Y(z - z_k); \quad w = w^0; \end{aligned} \quad (1)$$

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:* $u(z_k^-) = u(z_k^+); \quad v(z_k^-) = v(z_k^+); \quad w(z_k^-) = w(z_k^+); \quad (2)$

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

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

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:

$$\{\sigma_{123}\} = [C_{123}]\{\varepsilon_{123}\}; \quad \{\sigma_{xyz}\} = [C_{xyz}]\{\varepsilon_{xyz}\}. \quad (5)$$

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:

$$\psi_k = a_k \gamma_x + c_k \gamma_y; \quad \chi_k = d_k \gamma_x + b_k \gamma_y; \quad (6)$$

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, consequently 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:

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

where $\{\delta\}^T = \{u^0_{,x}; v^0_{,x}; u^0_{,y}; v^0_{,y}; \gamma_x; \gamma_y; \gamma_{x,z}; \gamma_{y,z}; \gamma_{x,y}; \gamma_{y,y}; -w^0_{,xz}; -w^0_{,yy}; -2w^0_{,xy}\}$ 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:

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

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 $m, n = 1, 3, 5, \dots$. The boundary conditions are the following:

$$\begin{aligned} u^0(x, 0) = u^0(x, b) = v^0(0, y) = v^0(a, y) = w^0(x, 0) = w^0(x, b) = w^0(0, y) = w^0(a, y) = 0; \\ \psi_x(x, 0) = \psi_x(x, b) = \psi_y(0, y) = \psi_y(a, y) = 0; \\ N_y(x, 0) = N_y(x, b) = N_x(0, y) = N_x(a, y) = M_2(x, 0) = M_2(x, b) = M_1(0, y) = M_1(a, y) = 0; \end{aligned} \quad (9)$$

where a, b are the plate's dimensions in the x and y direction, respectively. The displacement field solution can be expressed by:

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

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

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

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

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:

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

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:

$$\{\mathbf{u}\} = [\mathbf{N}]\{\mathbf{u}^e\}; \quad (13)$$

where $\{\mathbf{u}^e\}$ is the collection of the nodal degrees of freedom $\{u_i^e\}$ which is in this formulation $\{\mathbf{u}_i^e\}^T = \{u_i^0, v_i^0, \gamma_{xi}, \gamma_{yi}, w_{xi}^0, w_{yi}^0, w_{xyi}^0\}$. 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:

$$\{\delta\}_{13} = [\mathbf{R}]_{13 \times 32} \{\mathbf{u}^e\}_{32}. \quad (14)$$

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:

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

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 $t/T \ll 1$.

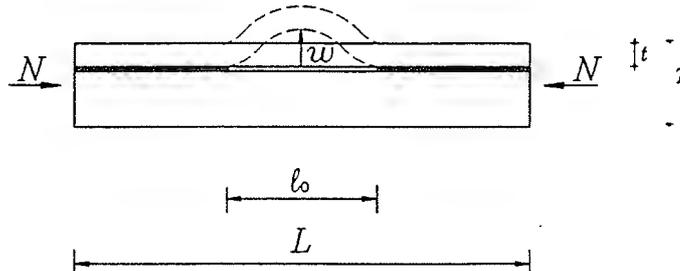


Fig. 1: Buckled narrow plate

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

$$k_1 = E_1 B t; \quad k_2 = E_2 B (T - t); \quad (16)$$

in which $k_1/k_2 \ll 1$ 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]:

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

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:

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

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

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

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:

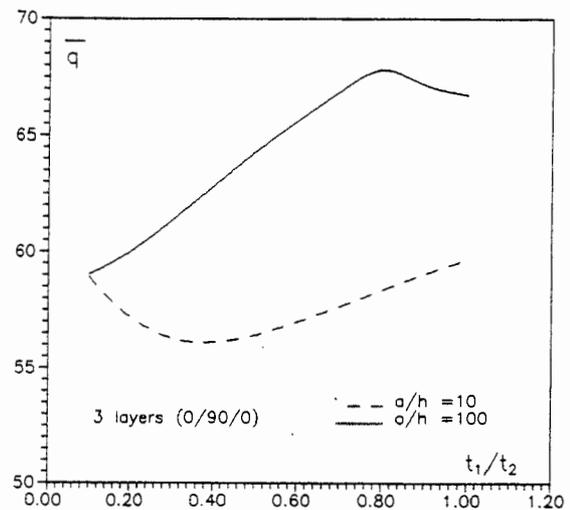
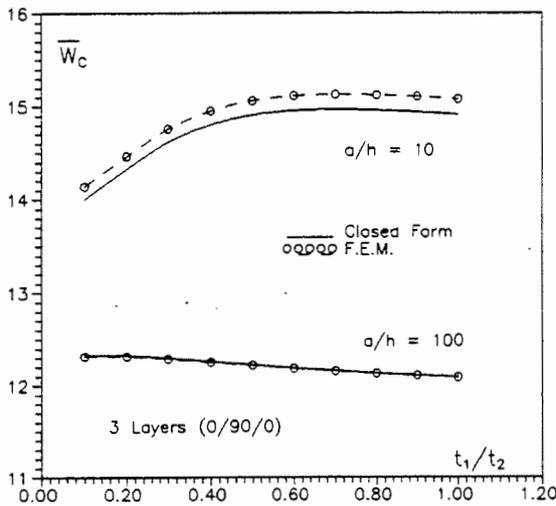
$$G_{IC} = \Gamma B; \quad G_{IIC} = \gamma \Gamma B; \quad m = n = 1; \quad \text{and} \quad \gamma \geq 1; \quad (22)$$

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 ($0^\circ/90^\circ/0^\circ$) made in Graphite-Epoxy T300/5208. The material has the following mechanical properties:

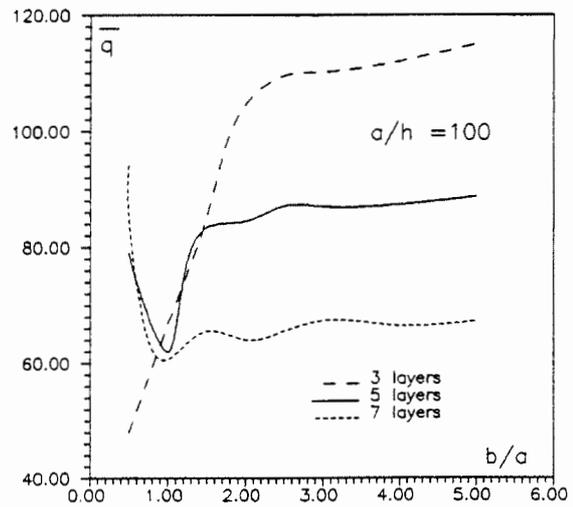
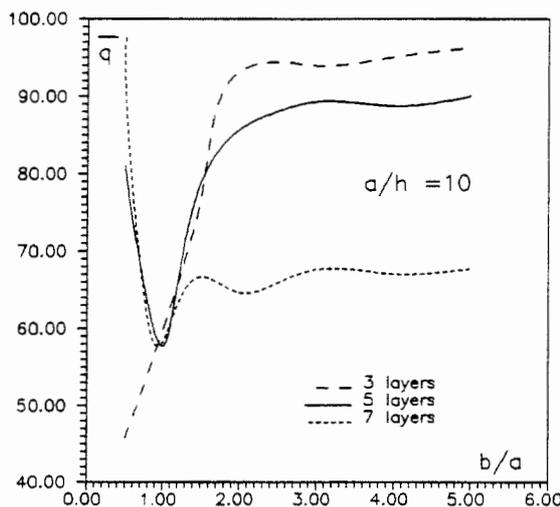
$$E_1 = 19.2 \cdot 10^6 \text{ psi}; \quad E_2 = 1.56 \cdot 10^6 \text{ psi}; \quad G_{12} = G_{13} = 0.82 \cdot 10^6 \text{ psi}; \quad G_{23} = 0.49 \cdot 10^6 \text{ psi};$$

$$\nu_{12} = 0.24; \quad X_T = 219.5 \cdot 10^3 \text{ psi}; \quad X_C = 246.0 \cdot 10^3 \text{ psi}; \quad Y_T = Z_T = 6.35 \cdot 10^4 \text{ psi};$$

$$Y_C = Z_C = 6.35 \cdot 10^6 \text{ psi}; \quad R = 9.8 \cdot 10^3 \text{ psi}; \quad S = T = 12.6 \cdot 10^3 \text{ psi}.$$

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 5×5 mesh of the proposed two-dimensional finite elements, because of the symmetry both in load and in geometry.

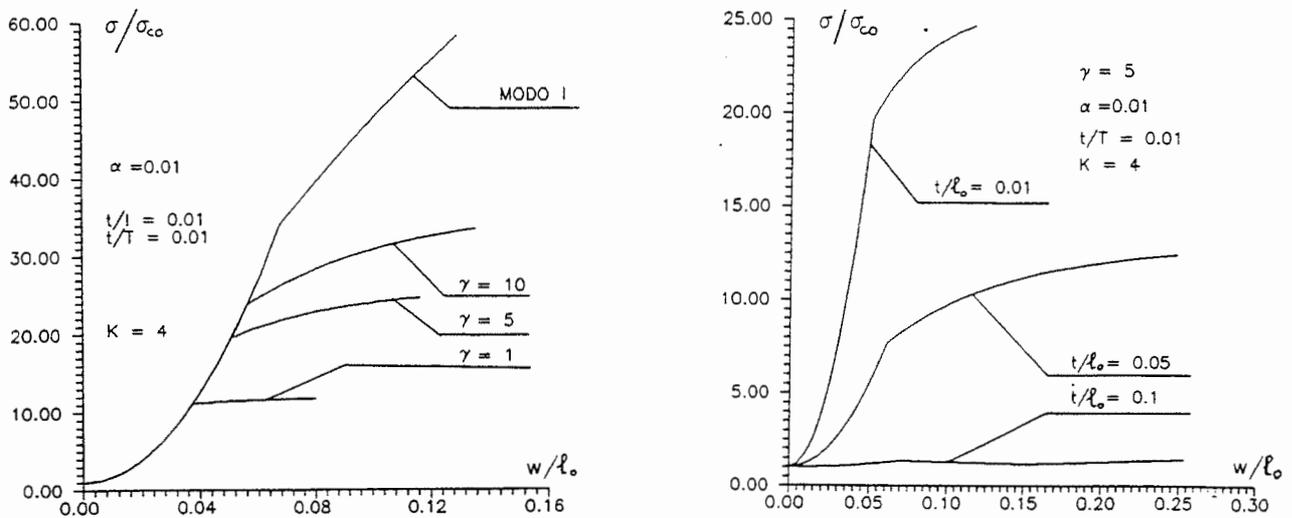
In Fig. 2a the finite element and the series solution results of the adimensionalised central transverse displacement $\bar{W}_c = ((W_c E_2 h^3)/(q_0 a^4)) \cdot 10^3$ v/s the t_1/t_2 ratio, t_1 being the outer layers thicknesses and t_2 the inner layer thickness, are compared. The results show that 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 $\bar{q} = ((q_0/E_2) \cdot (a/h)^2) \cdot 10^3$ is very clear which will be overestimated if the shear effects are neglected.



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-NO_x 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:

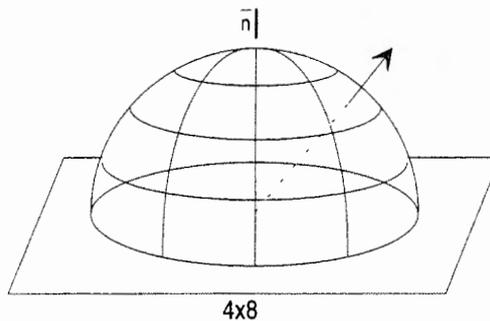
- reasonable calculation time (10 of seconds)
- good accuracy
- possibility to model different kind of geometry, including the radiation heat exchangers in the furnace.
- references in many applications

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 possible 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].

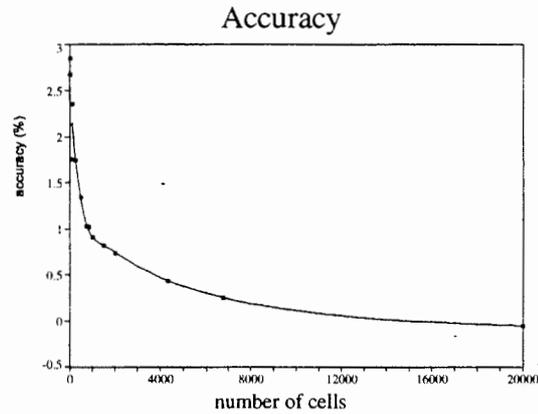


Picture 1: The picture shows the sectors from which centre the rays of the surface cell are being sent out. The normal angle is divided in 4 parts and the azimuthal angle in 8 parts.

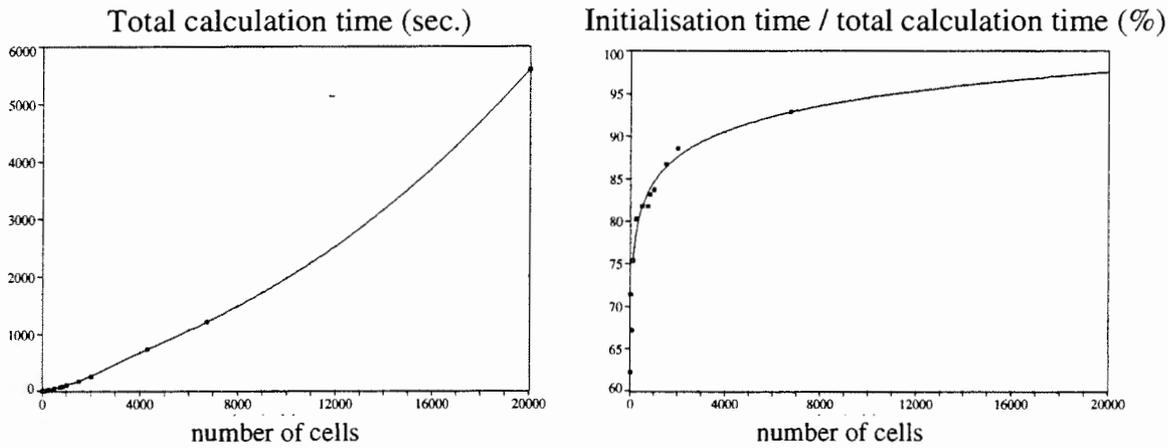
The values of temperature at outlet and total radiative heat transfer from the furnace are usually predicted satisfactorily by using constant absorption and scattering coefficients. However the radiative properties can be changed in the different parts of the boiler by the user. If desired, it is also possible to let the model calculate the emissivity of the flue gas, from the gas components of H₂O and CO₂ taking into account the soot distribution given by the user.

The calculation time increases and the accuracy improves rapidly with an 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

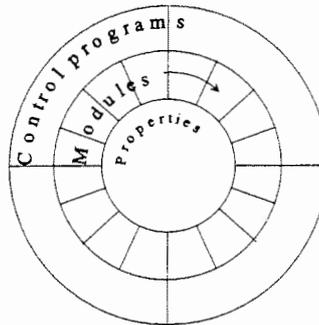
Conduction between the water cooled inner side and the heated outer side of a heat exchange wall, raises the outside temperature of the wall, thus decreasing the radiation heat transfer to it.

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].

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

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 properties.

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.

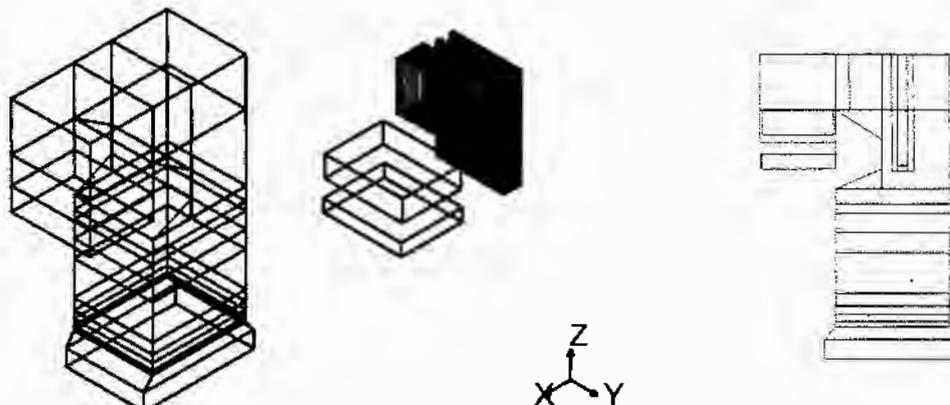
4. MODEL DESCRIPTION

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 commission 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 Rauhalahhti 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 %	
	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 Rauhalahhti 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 cau 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., Säteilämmönsiirtoyhtälön 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 O₂ with a productivity of 240Nm³/h/m³ 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.

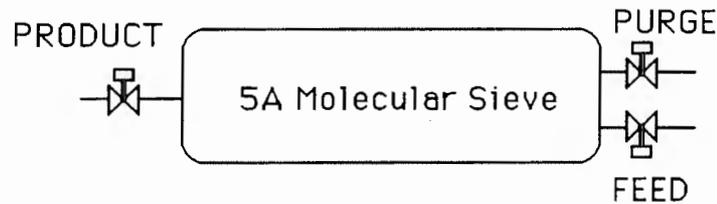


Figure 1. RPSA single column adsorber

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 adsorbent 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 Forchheimer 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).

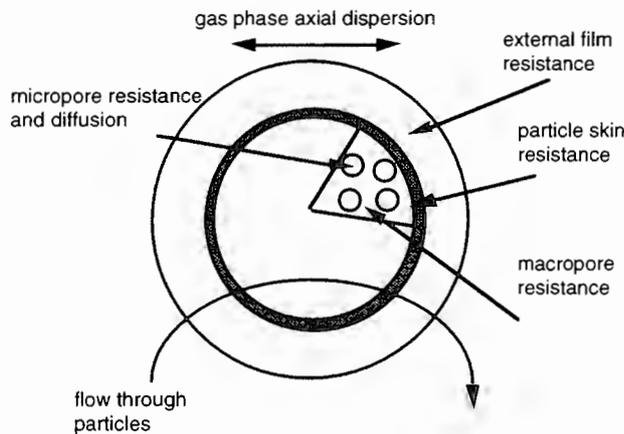


Figure 2. Transport resistances in adsorbent beds

The mass balance for component i in fluid phase is:

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

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

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

and the adsorption equilibrium:

$$q_i = \epsilon_{\text{micr}} P_i + (1 - \epsilon_{\text{micr}}) \frac{K_d P_i}{1 + \sum L_i P_i}$$

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

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:

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

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

$$V_n = K_v \sqrt{\frac{P_1 (P_1 - P_2)}{\gamma T}} \quad \text{if } p_1/p_2 \leq 2$$

$$= K_v \frac{P_1}{2\sqrt{\gamma T}} \quad \text{if } p_1/p_2 \geq 2$$

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

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

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

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:

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

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

And at $Z=L$:

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

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

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

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_{N_2} (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]

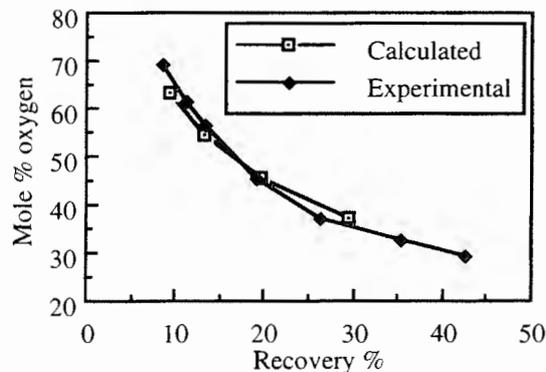


Figure 3. RPSA with 2.5 bar feed in a 1 m column with 5A (250°C), cycle timing .5/1/4 s. Experiment and simulation

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.

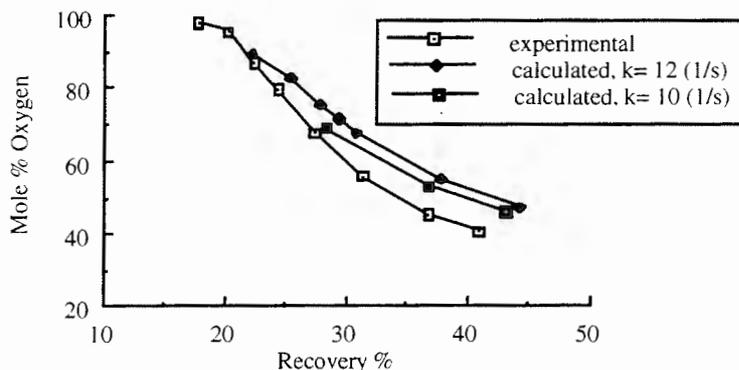


Figure 4. RPSA with 2.5 bar feed in a 1 m column with binderless 13X (350°C), cycle timing .5/1/4 s. Experimental data and model prediction.

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 meteooobservations 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 meteooobservations, 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 meteooobservations 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 Bolívar.
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-stationary 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 strategy 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 .

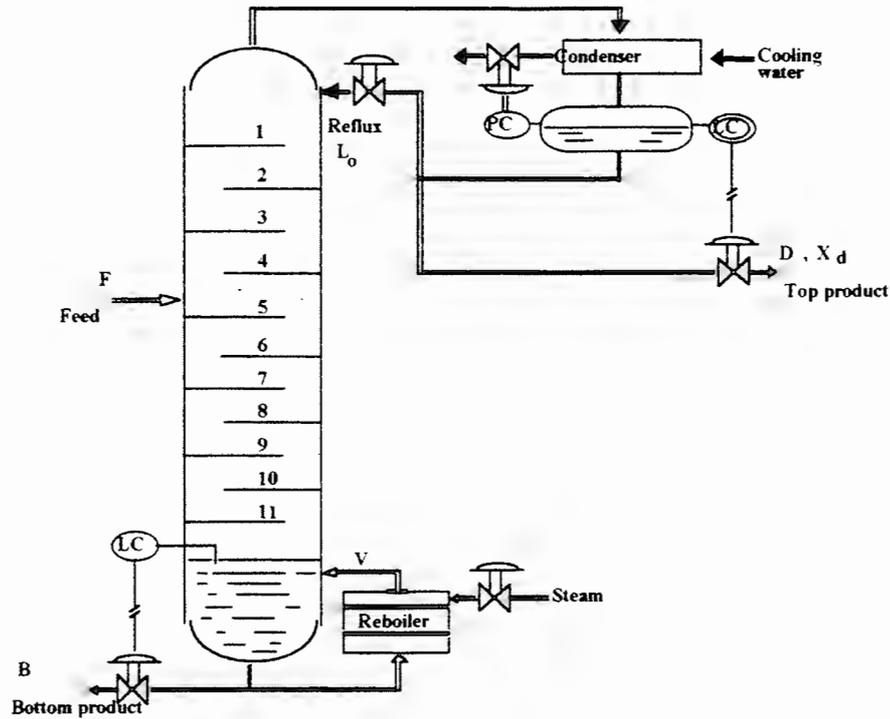


Figure 1. Schematic of pilot plant distillation column.

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:

for the n-th tray:

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

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

for the feed-tray:

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

$$\frac{dM_n X_n}{dt} = V Y_{n+1} + L_{n-1} X_{n-1} - V Y_n - L_n X_n + F X_f \quad (4)$$

for the reflux drum:

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

$$\frac{dM_d X_d}{dt} = V Y_0 - L_0 X_d - D X_d \quad (6)$$

for the reboiler:

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

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

where:

M_n : 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_0 .

The mathematical model is solved via numerical integration to simulate the time responses of the plant, under perturbations in F (the feed-stream flow-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 (continuous 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.

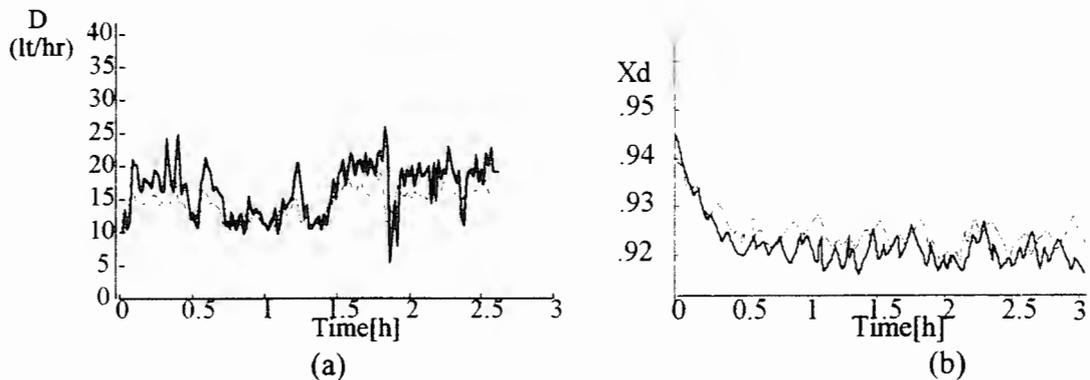


Figure 2. Plant and model outputs D and X_D under pseudo-random input signals on Q .

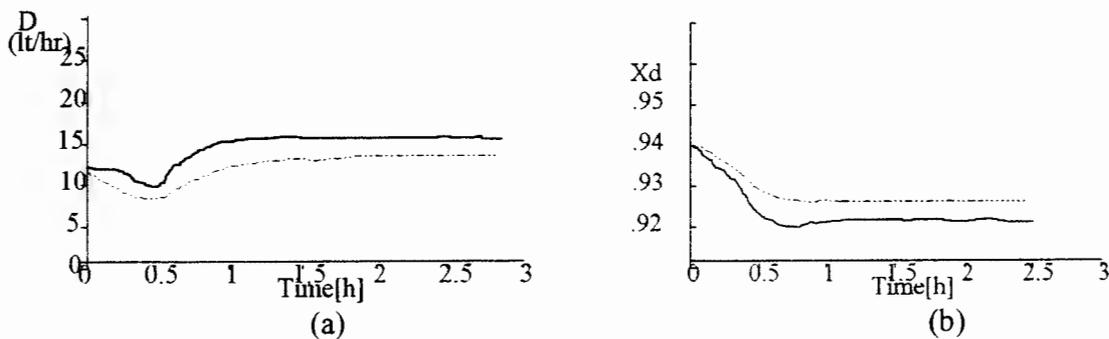


Figure 3. Plant and model outputs D and X_D under a step change in the feed flow-rate F .

4. RESULTS OF CLASSICAL RECURSIVE IDENTIFICATION

An alternative for the distillation plant modelling is the system identification, which relies 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 least 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 operating 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 simultaneously to collect 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:

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

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 $[2 \times 1]$.

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 finally 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 (continuous 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.

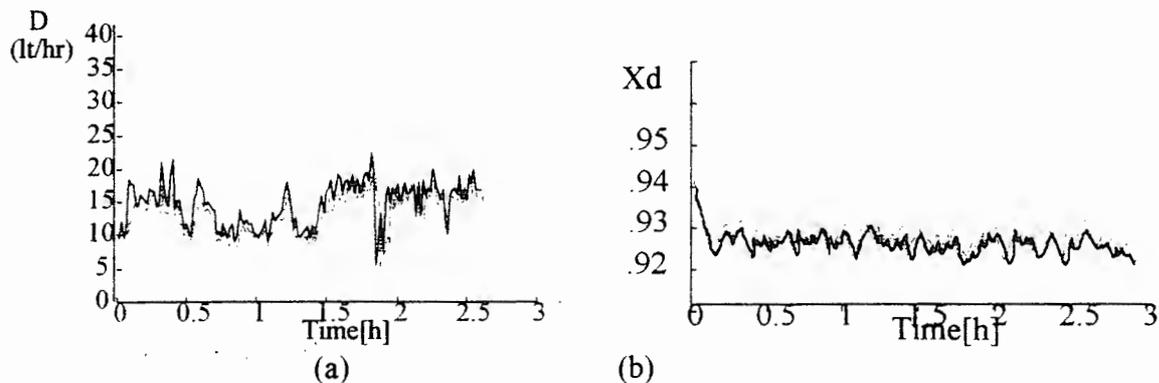


Figure 4. Plant and ARX model outputs D and X_d under pseudo-random input signals on Q .

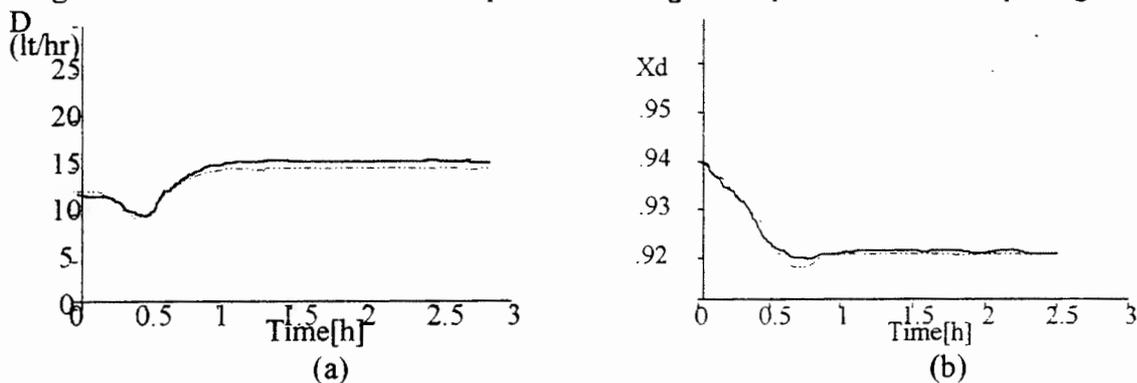


Figure 5. Plant and ARX model outputs D and X_d under a step change in the feed flow-rate F .

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 acquisition 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 perhaps non-linear identified models (of the neural networks type, for instance).

REFERENCES

1. Åstrom, 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 stoichiometric 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 referring 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.

Microorganisms 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 (CO₂) and methane (CH₄). Figure 1 shows the most important interactions between the participating bacterial groups, the substrates and intermediate products.

The solids (c₁) consisting of carbohydrates, lipids, and nitrogenous material are hydrolyzed by extracellular enzymes (e₀), produced by a group of bacteria (b₁^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 (c₃), CO₂ (c₄), H₂ (c₅) and acetic acid (c₆).

Acetogenic bacteria (b₂^a) consume the alcohols and carboxylic acids (C>2) of the hydrolyzers and acid formers (b₁^f) and transform them to acetic acid (c₆), CO₂ (c₄) and H₂ (c₅). The only way to perform this transformation is to have hydrogen-uptaking bacteria close by. The hydrogenic methanogens (b₃) are doing this work and produce methane (c₇) from CO₂ and H₂. 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 (b₄^{ma}) convert acetic acid (c₆) to methane (c₇) and carbon dioxide (c₄).

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 stoichiometric yield coefficient.

The substrate utilization rate amounts to :

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

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

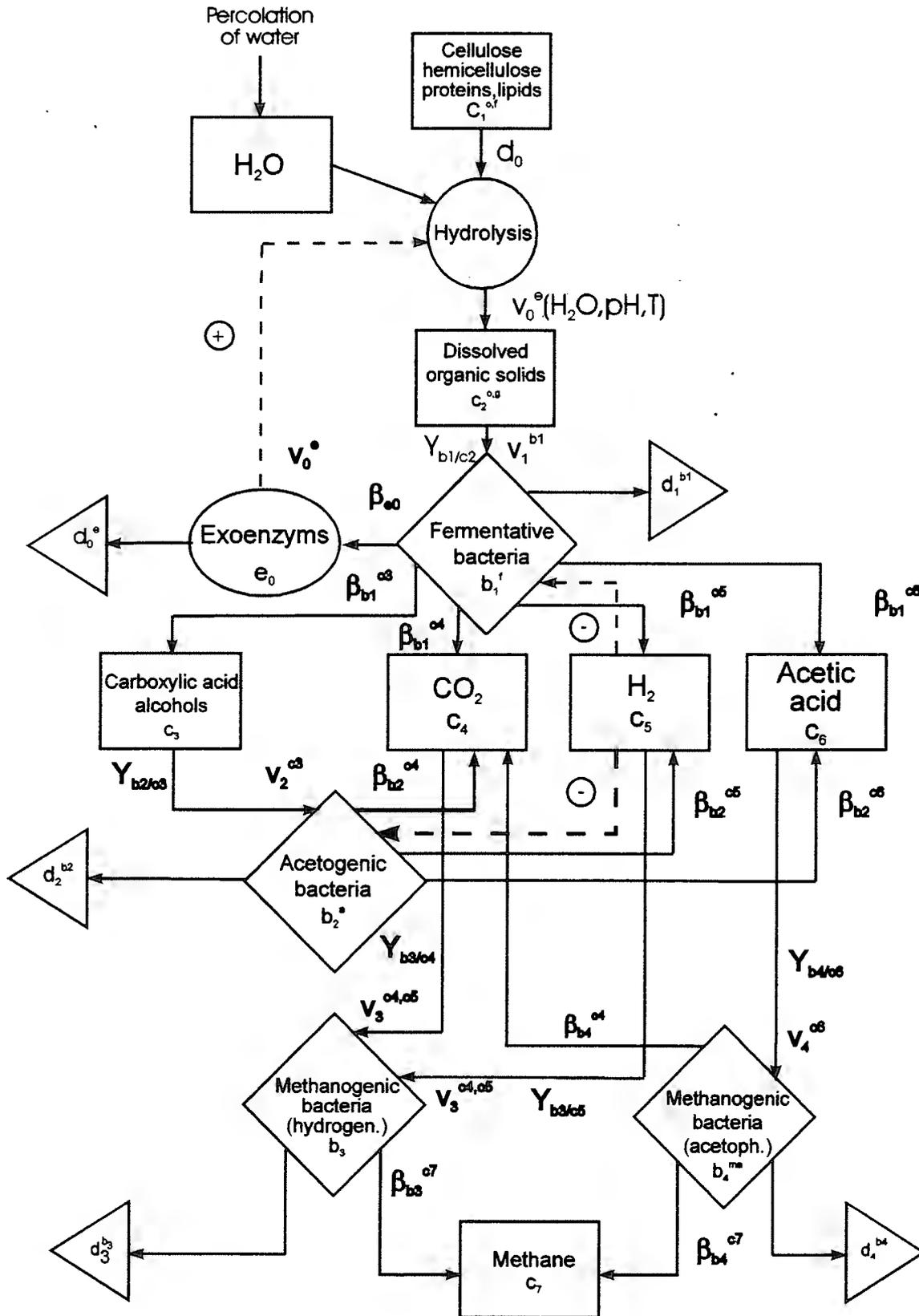


FIG. 1 FOODWEB OF THE LANDFILL ECOSYSTEM

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

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 :

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

$n=0, m>0$ yields an inhibitory control characteristic, whereas $n \geq 1, m > n$ 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).

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

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

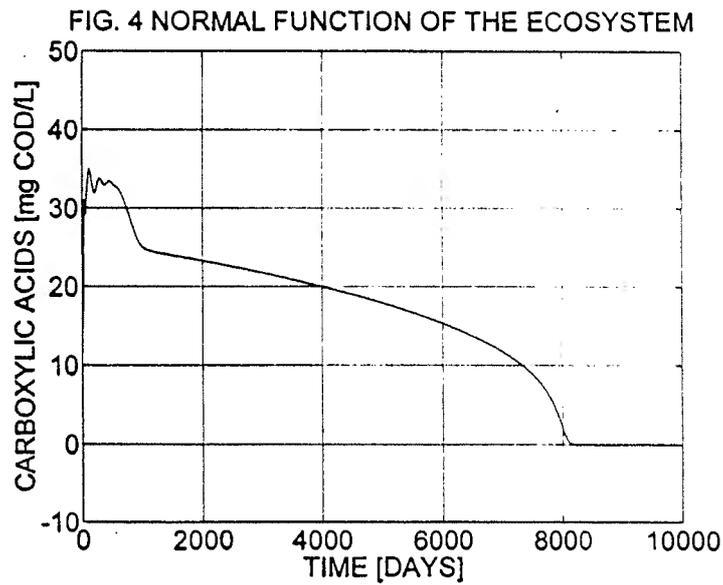
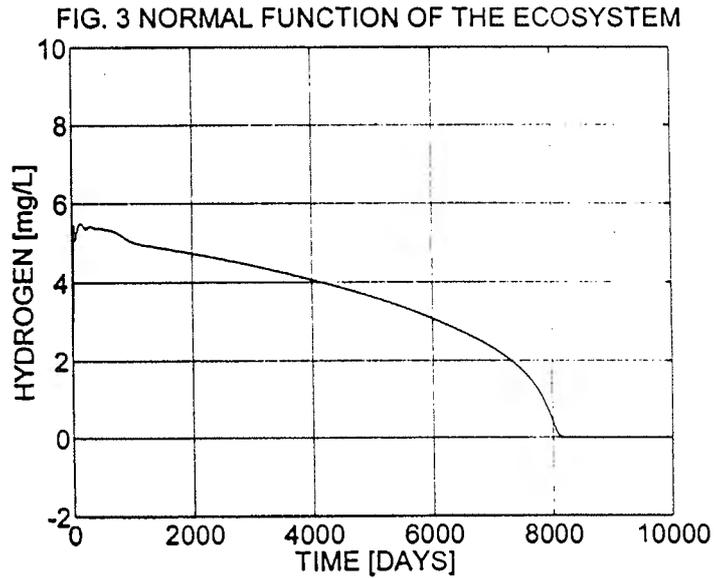
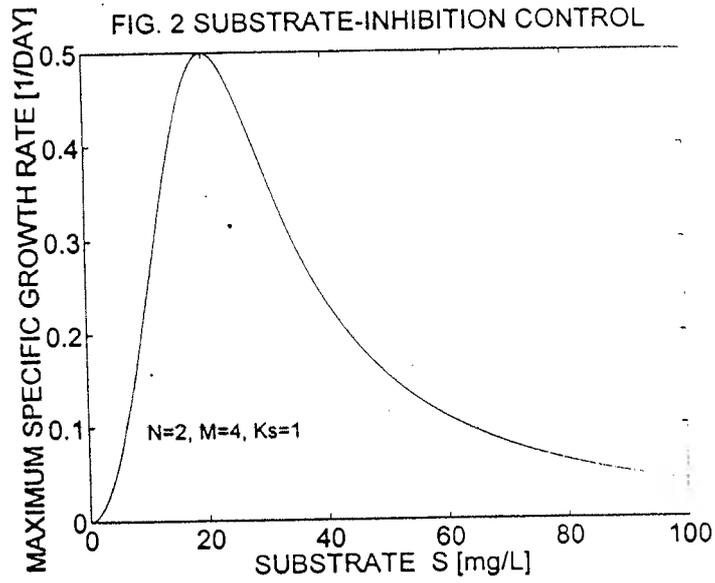
4. SIMULATION RESULTS

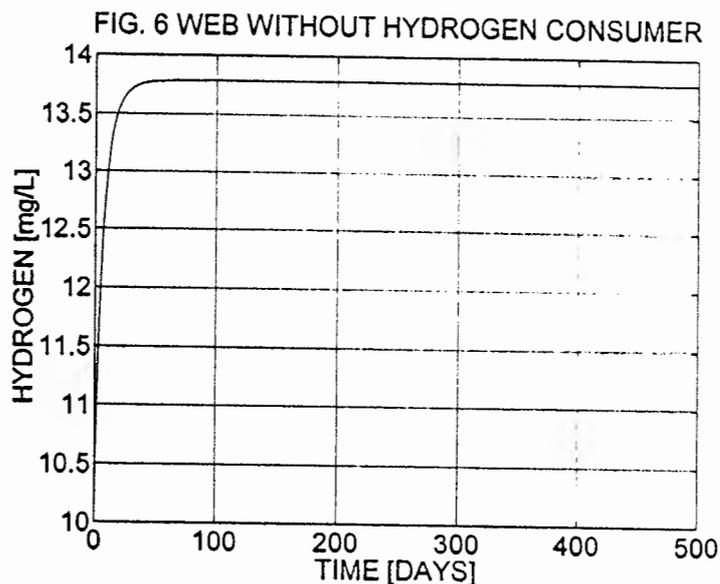
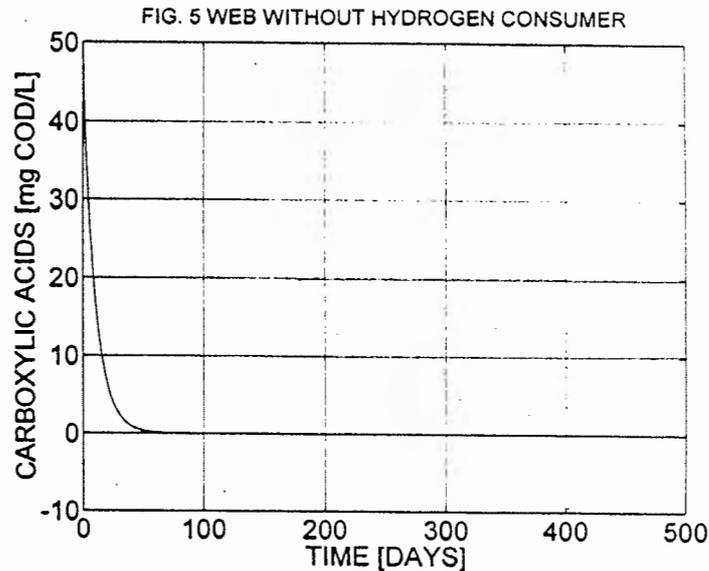
The dynamical properties of the derived system of differential equations are analyzed by simulation experiments with MATLAB and SIMULINK. Parameters are taken from the literature [EASTMAN, FERGUSON 1981; LAWRENCE, McCARTY 1969] and the results are validated with experimental data [EHRIG, SCHEELHAASE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

A typical time course of system 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.





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, Cagliari, 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

^aPhysiology of Reception Department, Pavlov Institute of Physiology, nab. Makarova 6, 199034, St. Petersburg, Russia

^bApplied Mathematics Department, Pavlov Institute of Physiology,

^cPhysiology of Reception Department, Pavlov Institute of Physiology,

^dPhysiology 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]:

$$\begin{aligned}
 \dot{V} &= F_1(V, h, n, h_a, z, Ca \mid \bar{g}_{KCa}, \bar{g}_{Ca}, I) \\
 &= I - \bar{g}_{Na} m_\infty^3(V) h (V - V_{Na}) - \bar{g}_K n^4 (V - V_K) - \bar{g}_L (V - V_L) \\
 &\quad - \bar{g}_a m_{a\infty}^3(V) h_a (V - V_K) - \bar{g}_{Ca} z \frac{V - V_{Ca}}{0.5 + Ca} - \bar{g}_{KCa} Ca \frac{V - V_K}{0.5 + Ca}, \\
 \dot{h} &= F_2(V, h) = 0.8 \frac{h_\infty(V) - h}{\tau_h(V)}, \\
 \dot{n} &= F_3(V, n) = 0.8 \frac{n_\infty(V) - n}{\tau_n(V)}, \\
 \dot{h}_a &= F_4(V, h_a) = h_{a\infty}(V) - h_a, \\
 \dot{z} &= F_5(V, z) = \frac{z_\infty(V) - z}{23.5}, \\
 \dot{Ca} &= F_6(V, z, Ca) = -\frac{1}{2} p k_{Ca} z \frac{V - V_{Ca}}{0.5 + Ca} - p Ca,
 \end{aligned} \tag{1}$$

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 $J(z, Ca | \bar{g}_{KC_a}, \bar{g}_{C_a})$ 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 $a_1 = -\text{trace } J$ 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

$$\begin{aligned} F_5(V, z) &= 0, \\ F_6(V, z, Ca) &= 0, \\ F_1(V, z, Ca | \bar{g}_{KC_a}, \bar{g}_{C_a}, I) &= 0, \\ \text{trace } J(z, Ca | \bar{g}_{KC_a}, \bar{g}_{C_a}) &= 0, \\ \det J(z, Ca | \bar{g}_{KC_a}, \bar{g}_{C_a}) &> 0. \end{aligned} \tag{4}$$

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}_{KC_a}) , (I, \bar{g}_{C_a}) can be given by the explicit parametric form

$$\begin{aligned} \bar{g}_i &= \psi(V) = -\frac{B_{11}(V)+B_{22}(V)}{A_{11}(V)+A_{22}(V)}, \\ I &= \varphi(V), \\ \psi(V)^2 C_1 + \psi(V) C_2 + C_3 &> 0, \end{aligned} \tag{5}$$

$$i = 1, 2, \bar{g}_1 = \bar{g}_{KC_a}, \bar{g}_2 = \bar{g}_{C_a}.$$

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}_{KC_a}) and (I, \bar{g}_{C_a}) 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

$$\begin{aligned}
 T(V) &= \bar{g}_{Na} m_{\infty}^3(V)(V - V_{Na})h_{\infty}(V) + \bar{g}_K n_{\infty}^4(V)(V - V_K) + g_L(V - V_L), \\
 T_1(V) &= \bar{g}_a m_{a\infty}^3(V)(V - V_K)h_{a\infty}(V), \\
 T_2(V) &= \frac{z_{\infty}(V)(V - V_{Ca})}{C_{a\infty}(V) + 0.5}, \\
 T_3(V) &= \frac{C_{a\infty}(V)(V - V_K)}{C_{a\infty}(V) + 0.5}, \varphi(V) = T(V) + \bar{g}_a T_1(V) + \bar{g}_{C_a} T_2(V) + \bar{g}_{kC_a} T_3(V), \\
 c &= 1/23.5, \quad k = -0.5pk_{C_a} \\
 C_1 &= A_{11}A_{22} - A_{12}A_{21}, \\
 C_2 &= B_{11}A_{22} + A_{11}B_{22} - A_{12}B_{21} - B_{12}A_{21}, \\
 C_3 &= B_{11}B_{22} - B_{12}B_{21}.
 \end{aligned} \tag{7}$$

Variant 1 (for constructing the Hopf bifurcation curve on the (I, \bar{g}_{KC_a}) plane).

$$\begin{aligned}
 A_{11}(V) &= -cC_{a\infty}(V), \\
 A_{12}(V) &= 0.5cz'_{\infty}(V) \frac{V_K - V}{C_{a\infty}(V) + 0.5}, \\
 A_{21}(V) &= k(V - V_{Ca}) \frac{C_{a\infty}(V)}{C_{a\infty}(V) + 0.5}, \\
 A_{22}(V) &= \frac{kz_{\infty}(V)[0.5(V_K - V) + C_{a\infty}(V)(V_{Ca} - V)]}{(C_{a\infty}(V) + 0.5)^2} - pC_{a\infty}(V), \\
 B_{11}(V) &= c[\bar{g}_{C_a}((V_{Ca} - V)z'_{\infty}(V) - z_{\infty}(V)) - (0.5 + C_{a\infty}(V))(T'(V) + \bar{g}_a T'_1(V))], \\
 B_{12}(V) &= c\bar{g}_{C_a} z'_{\infty}(V) T_2(V) \\
 B_{21}(V) &= k(V - V_{Ca})(T'(V) + \bar{g}_a T'_1(V)), \\
 B_{22}(V) &= \frac{k(V_{Ca} - V)z_{\infty}(V)}{C_{a\infty}(V) + 0.5} (T'(V) + \bar{g}_a T'_1(V)) - p[\bar{g}_{C_a} z_{\infty}(V) + (0.5 + C_{a\infty}(V))(T'(V) + \bar{g}_a T'_1(V))].
 \end{aligned}$$

Variant 2 (for constructing the Hopf bifurcation curve on the (I, \bar{g}_{Ca}) plane).

$$\begin{aligned}
 A_{11}(V) &= -c((V_{Ca} - V)z'_{\infty}(V) - z_{\infty}(V)), \\
 A_{12}(V) &= cz'_{\infty}(V)T_2(V) \\
 A_{21}(V) &= 0 \\
 A_{22}(V) &= -kp \\
 B_{11}(V) &= -c[(0.5 + Ca_{\infty}(V))(T'(V) + \bar{g}_a T'_1(V)) + \bar{g}_{kCa} Ca_{\infty}(V)], \\
 B_{12}(V) &= 0.5c\bar{g}_{kCa} z'_{\infty}(V) \frac{V_K - V}{0.5 + Ca_{\infty}(V)} \\
 B_{21}(V) &= \frac{k(V_{Ca} - V)}{c(0.5 + Ca_{\infty}(V))} B_{11}(V) \\
 B_{22}(V) &= \frac{1}{c} B_{11}(V) \left(\frac{kT_2(V)}{0.5 + Ca_{\infty}(V)} - p \right) + \frac{0.5k\bar{g}_{kCa} z_{\infty}(V)(V_K - V)}{(0.5 + Ca_{\infty}(V))^2}
 \end{aligned} \tag{9}$$

4.1. Line drawings

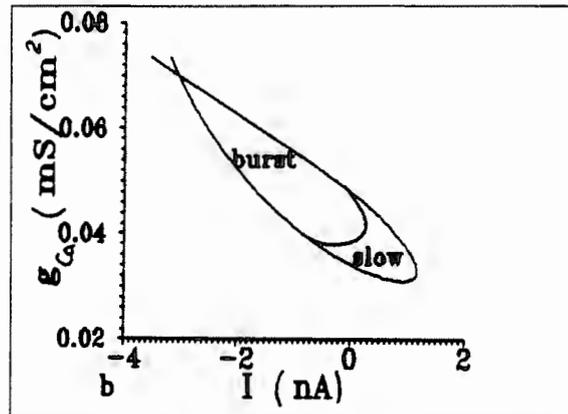
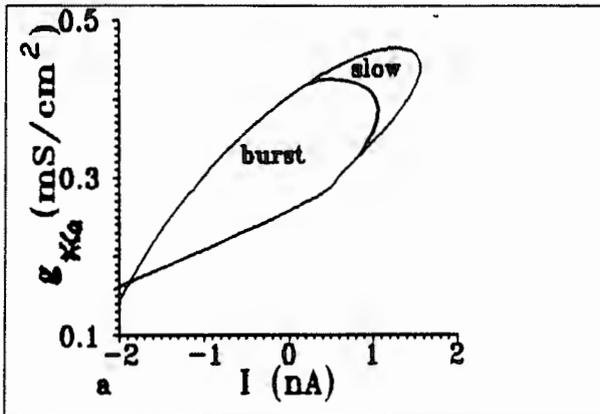
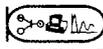


Figure 1. The Hopf bifurcation curve and bursting region on the (I, \bar{g}_{kCa}) plane.

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 electrophysiological 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 (conductive 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)).

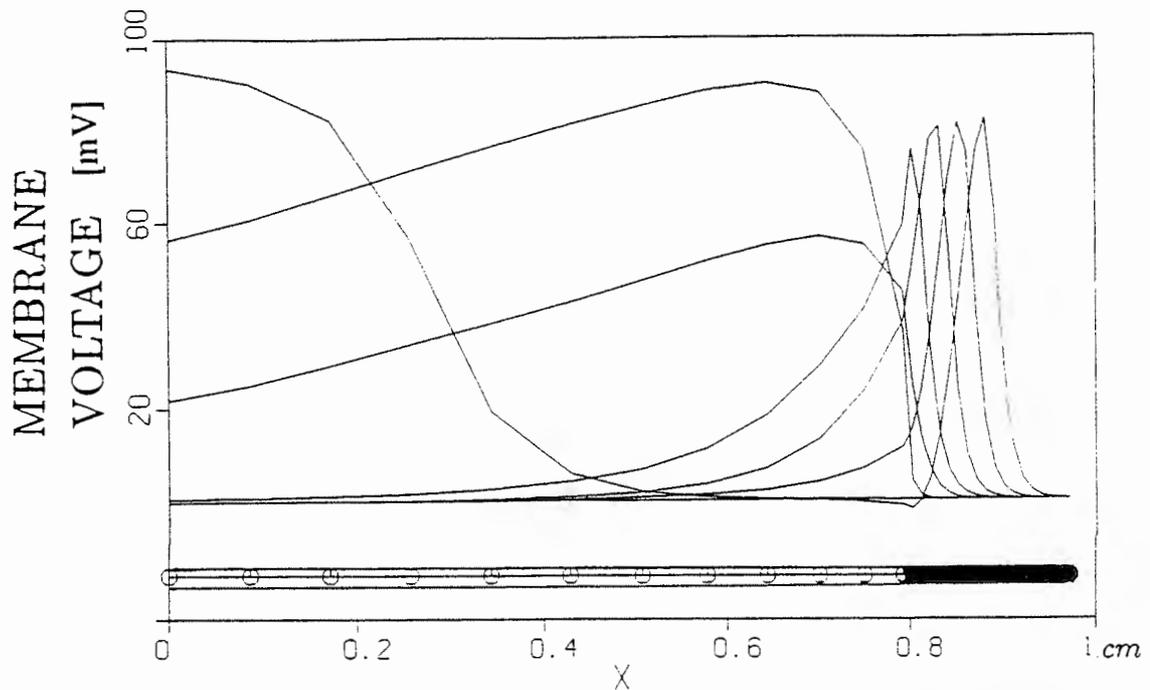


Fig. 1. Propagating action potential along a nerve fiber of 1 cm length which changes membrane composition. The curves show voltage distributions in intervals of 0.1ms. The left part of the axon is myelinated, i.e. it consists of 12 nodes of Ranvier (marked by circles in the lower part) with active membrane 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 μ s 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 characteristics 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 300 μ 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).

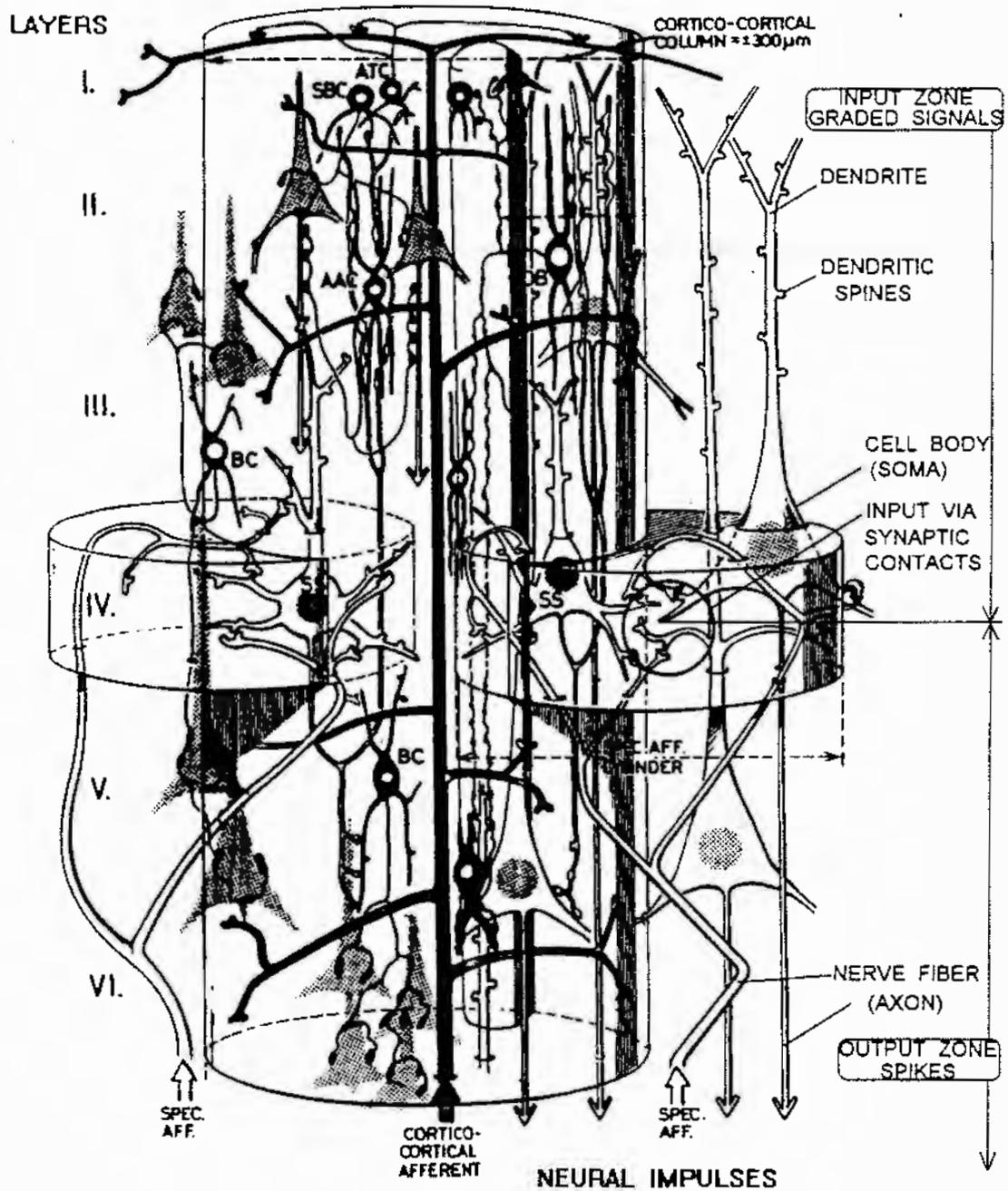


Fig. 2. Neural network of the human cortex. The terminal section of an afferent cortico-cortical axon supplies pyramidal cells and non-pyramidal cells of the six cortical layers (ATC axonal tuft cell, SBC small basket cell, AAC axoaxonic chandelier cell, BC basket cell, CDB double bouquet cell, SS spiny 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 granule 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.

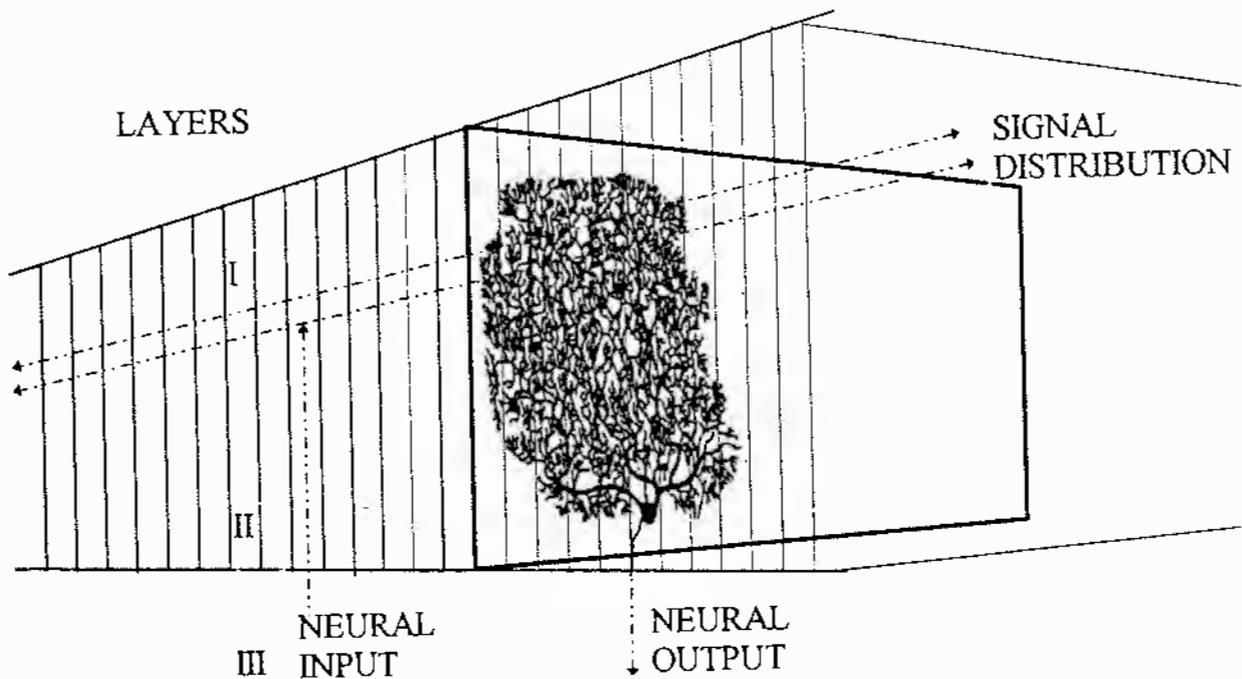


Fig. 3 Organization of the cerebellar cortex. A system of parallel axons from granule cells crosses at right angles the planes containing the dendritic branches of two-dimensional Purkinje cells. With some delay this data bus system supplies a series of Purkinje cells with the same information.

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 efficiency 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
- Llinás 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.* Sinauer Associates, Sunderland MA, 3rd. edition, p 540
- McCormick D.A. and Huguenard J.R. 1992. A model of the electrophysiological properties of Thalamocortical 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 analogs 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 medial nucleus tractus solitarius receiving 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.

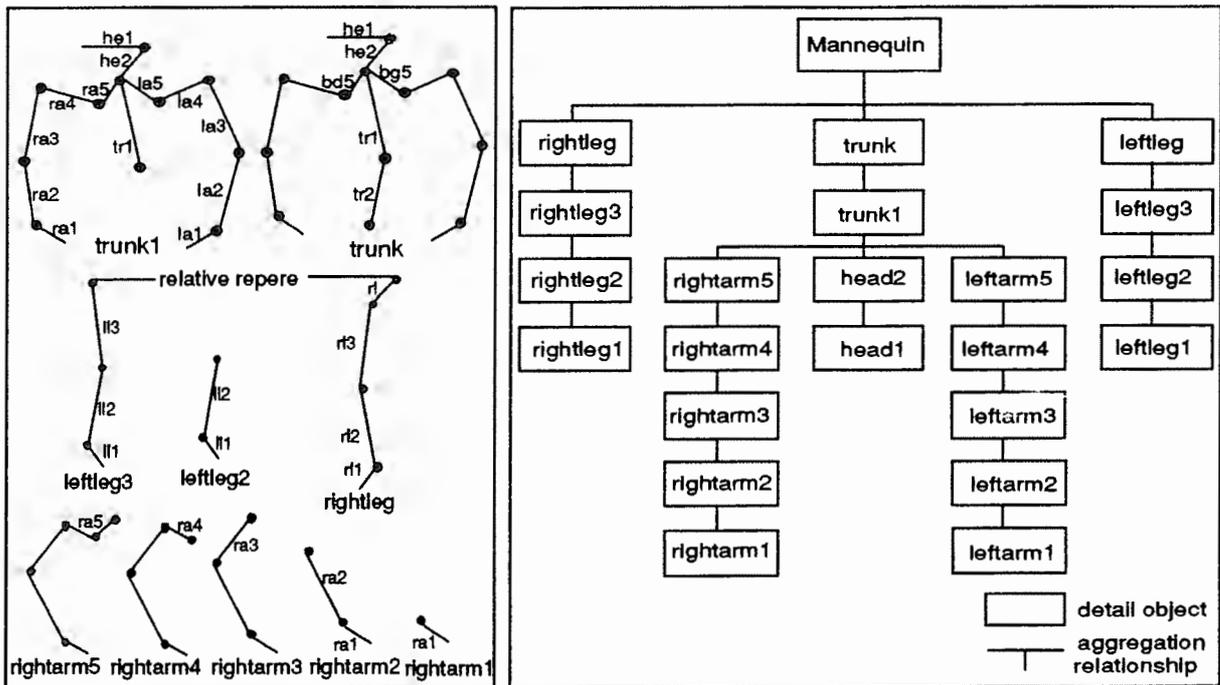


Figure 1. Detail based description of human skeleton.

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.

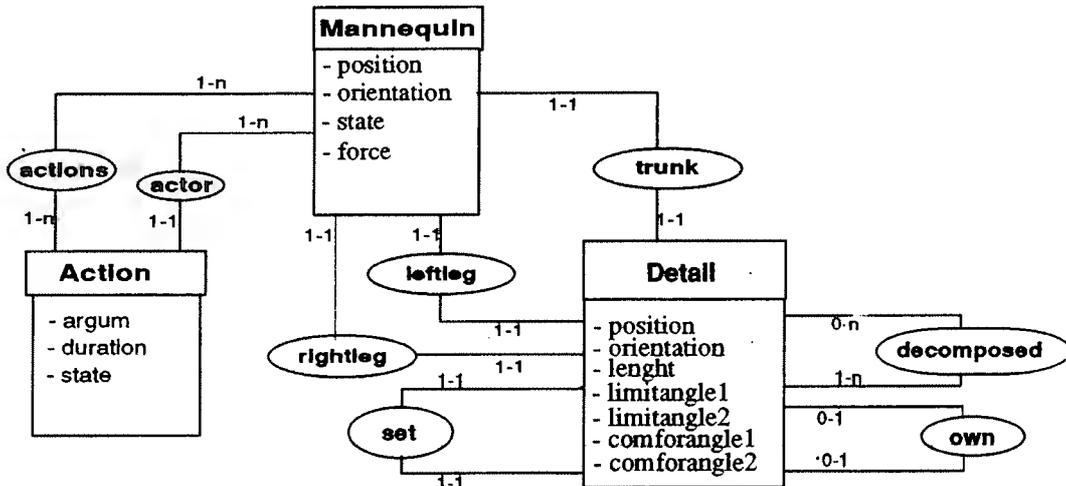


Figure 2. Human skeleton model.

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.

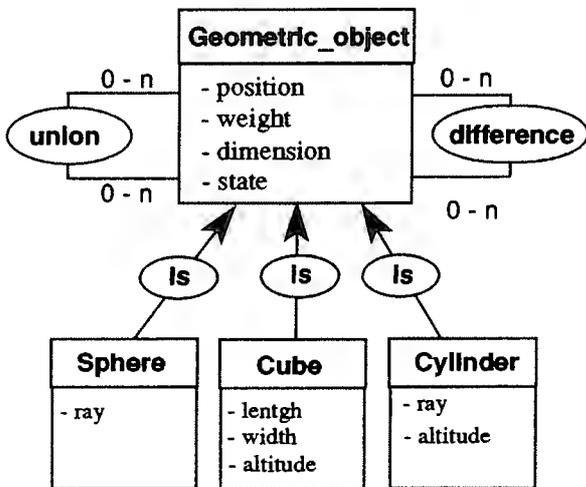


Figure 3. Environment model.

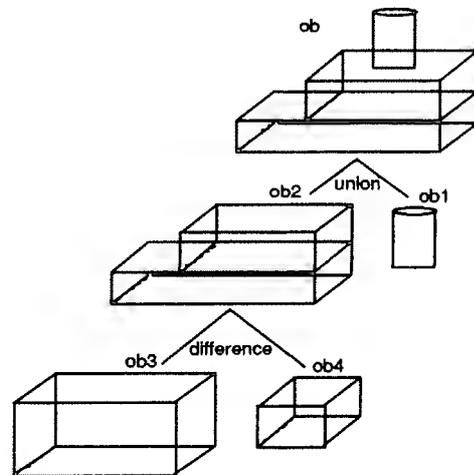


Figure 4. Example of a CSG representation.

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.

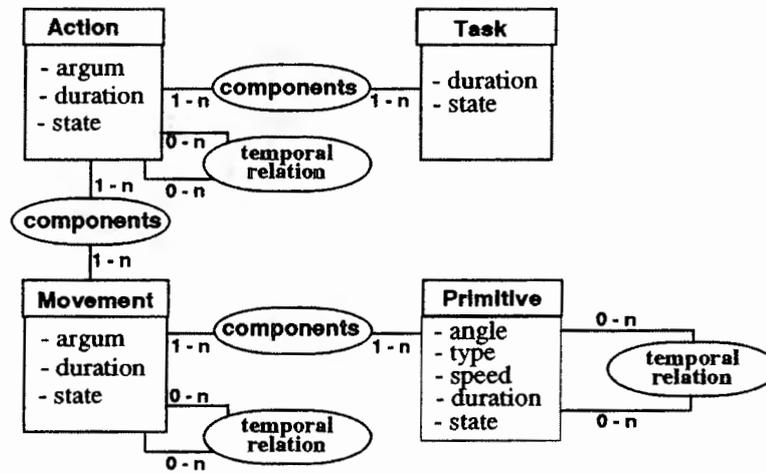


Figure 5. Operational model.

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 world 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_bust₁*, *lower_leg*, *lift_bust* and *incline_bust₂*. Arrows within levels show how the performing of certain operators is partially ordered with respect to causality relationships. For example, the movement *Incline_bust₁* must be performed before *lower_leg* and *incline_bust₂*. However these later can be carried out concurrently.

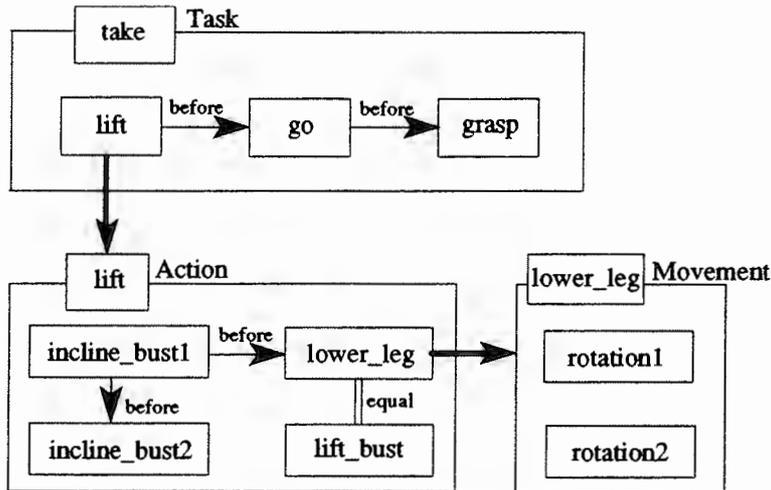


Figure 6. Hierarchical plan of take task.

To illustrate the instances generated at each level, we give below the example of the movement *lower_leg*.

```
( Movement : lower_leg
  -- argum : leftleg3
  -- components : [rotation1 , rotation2]
  -- before : [ incline_bust1]
  -- equals : [lift_bust] )
```

The plan generated to achieve this movement is constituted by the geometrical primitives *rotation₁* and *rotation₂*, 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.

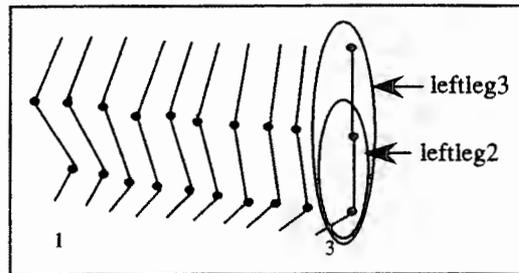
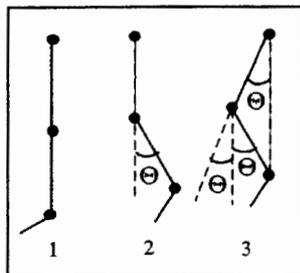


Figure 7.a. Sequential execution of plan. Figure 7.b. Parallel execution of plan.

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

- Akoev G.N. 93
 Aponte F. 81

 Bakulin A.V. 79
 Baron G.V. 73
 Bedrov Y.A. 93
 Böttner R. 87
 Breitenecker F. 9, 43
 Bruno D. 61

 Campos M.I. 81
 Cheruy A. 21

 Dawid H. 27
 Deschizeaux P. 105
 Dick O.E. 93
 Doboga F. 21
 Du T. 37

 Fourar H. 105

 Gajdár T. 49

 Huang K. 37
 Husinsky I. 9, 43

 Katkov V.L. 79
 Kerckhoffs E.J.H. 1
 Klußmann J. 15
 Kniffki K.-D. 93
 Kolumbán G. 49
 Koukam A. 105
 Krauth J. 15

 Kuznetsov G.P. 79

 Lamanna R. 81
 Lampenius B. 67
 Luciano R. 55

 Mehlmann A. 27

 Nichiporovich I.A. 79
 Niwinski J. 3

 Pfützner H. 99
 Popescu D. 21
 Porco G. 61

 Rattay F. 99

 Schleusener B. 87
 Schuster G. 43
 Splanemann R. 15

 Verelst H. 73

 Weisz W. 33

 Zha Y. 37
 Zhu L.-Q. 73

 Zinno R. 55, 61



ARGESIM Reports

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	3-901608-02-8
# 3	EUROSIM'95 - Poster Book	F. Breitenecker, I. Husinsky	3-901608-03-6
# 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 Systems and Control"	D. Murray-Smith, D.P.F. Möller, F. Breitenecker	3-901608-05-2
# 6	Seminar Modellbildung und Simulation - COMETT - Course "Object-Oriented Discrete Simulation"	N. Kraus, F. Breitenecker	3-901608-06-0
# 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	3-901608-08-7