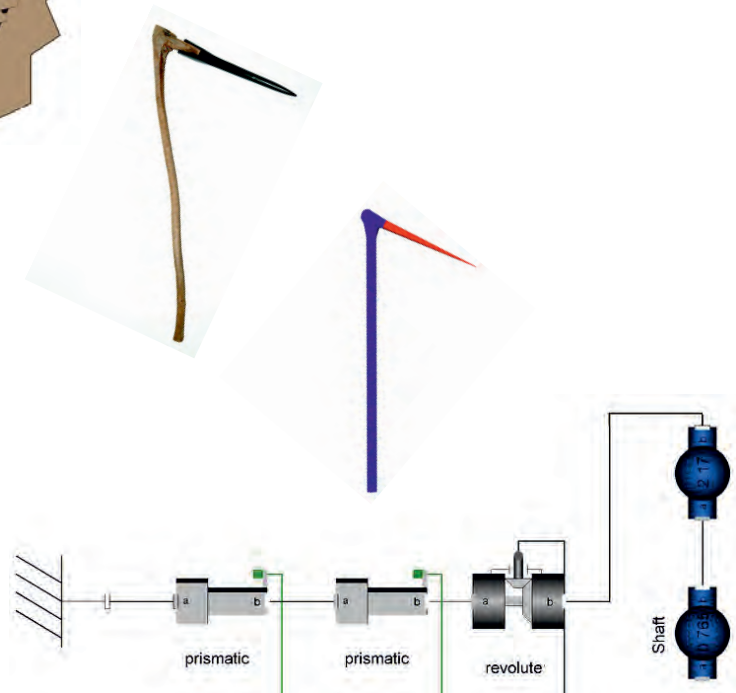
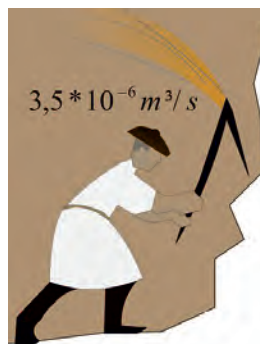
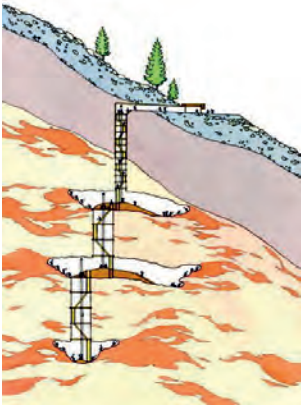


SNE SIMULATION NOTES EUROPE



Volume 22 Number 1
April 2012

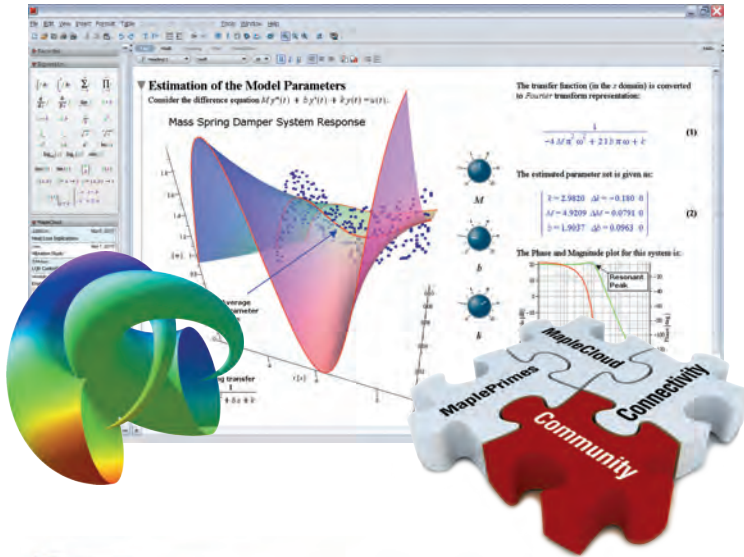
Print ISSN 2305-9974
Online ISSN 2306-0271



Journal on Developments and
Trends in Modelling and Simulation
Membership Journal for Simulation
Societies and Groups in EUROSIM

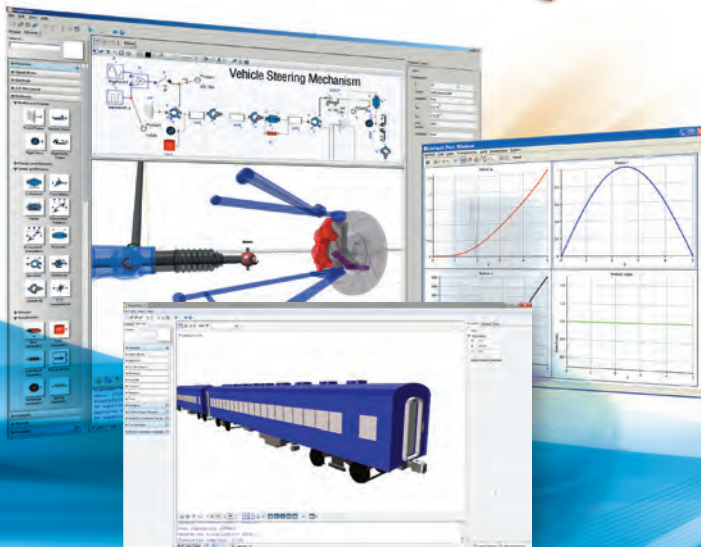


New challenges in modeling and simulation require powerful tools



Maple™ 15

The result of over 30 years of cutting-edge research and development, Maple combines the world's most powerful mathematical computation engine with an intuitive, "clickable" user interface.





MapleSim™ 5

MapleSim is a physical modeling and simulation tool built on a foundation of symbolic computation technology.

Get a Price Quote:
www.maplesoft.com/sne

Contact us:

  +49 (0)241/980919-30



Editorial

Dear Readers - *The years 2011 and 2012 are accompanied by essential changes for SNE – from Simulation News Europe to Simulation Notes Europe, from limited SNE access to an open access strategy for Online SNE, from classical citation to introduction of DOI identifiers for web availability and citation index purposes. ARGESIM, SNE's official publisher, is now member of CrossRef, the official DOI® link registration agency for scholarly and professional publications. CrossRef's citation-linking network today covers tens of millions of articles from thousands of scholarly and professional publishers.*

This first SNE issue of SNE Volume 22 publishes contributions with a broad range of modelling and simulation. As common denominator new challenges (e.g. sustainable and energy-efficient production, implementation of meta-model structures, DAE algorithms for real-time applications) can be seen. One contribution is a novelty - physical modelling for archaeology: here simulation results should not prove certain behaviour; they should allow archeologists to exclude certain behavior and unknown handling of physical tool.

We also are glad to continue with SNE 22(1) our ARGESIM Benchmark Series with a classical benchmark solution, and with the first solution to the new benchmark C20 'Complex Production System' – defined by S. Schreiber in SNE 21(3-4) as first representative of the new type 'Extended ModSim Benchmark'. As solutions to this extended benchmark type are very challenging, we allow for documentation up to six pages, so that the publication ranges as full scientific publication.

I would like to thank all authors for their contribution and all people who are helping to manage the non-trivial DOI registration process and web presentation pages for Online SNE and layout of Printed SNE.

Felix Breitenecker, Editor-in-Chief, eic@sne-journal.org

Contents

| | |
|---|--------|
| Energy Flows in Industrial Buildings and Machines using the Example of a Node Model. <i>M. Schäfer, J. Wagner, A. Schlüter, J. Hesselbach</i> | 1 |
| Meta-Model - based Estimation of Optimal System Variants with MATLAB/Simulink. <i>O. Hagedorf, T. Pawletta, R. Larek</i> | 7 |
| A Qualitative Comparison of Two Hybrid DEVS Approaches. <i>C. Deatcu, T. Pawletta</i> | 15 |
| Physical Modelling for Hallstatt Archaeology. <i>B. Heinzl, E. Auer, B. Slowacki, K. Kowarik, H. Reschreiter, N. Popper1, F. Breitenecker</i> | 25 |
| Object-Oriented Modelling for Energy-efficient Production. <i>B. Heinzl, M. Rößler, N. Popper, M. Landsiedl, A. Dimitriou, F. Breitenecker</i> | 33 |
| Automatic Detection of QRS Complex, P-Wave and T-Wave in the Electrocardiogram. <i>M. Bachler, C. Mayer, B. Hametner, S. Wassertheurer</i> | 39 |
| Evaluation and Adaptation of Techniques for Higher Index DAE with Respect to Real-Time Simulation. <i>J. Frochte</i> | 45 |
| An Actor-oriented Approach to ARGESIM Benchmark C05 'Two State Model' using Berkeley Ptolemy II. <i>P. Einzinger</i> ... | 47 |
| A Discrete-event Modeling of ARGESIM Benchmark C20 'Complex Production System' using Flexsim Simulation Software. <i>E. Schulz, J. Paul, S. Schreiber, A. Fay</i> | 47 |
| EUROSIM Societies Info & News | N1-N10 |

Reader's Info

Simulation Notes Europe publishes peer reviewed *Technical Notes*, *Short Notes* and *Overview Notes* on developments and trends in modelling and simulation in various areas and in application and theory. Furthermore SNE documents the ARGESIM Benchmarks on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions (*Benchmark Notes*). Special *Educational Notes* present the use of modelling and simulation in and for education and for e-learning.

SNE is the official membership journal of EUROSIM, the Federation of European Simulation Societies. A News Section in SNE provides information for EUROSIM Simulation Societies and Simulation Groups.

SNE is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With Online SNE the publisher ARGESIM follows the Open Access strategy, allowing download of published contributions for free. Since 2011 Online SNE contributions are identified by a DOI (Digital Object Identifier) assigned to the publisher ARGESIM (DOI prefix 10.11128). Print SNE, high-resolution Online SNE, full SNE Archive, and source codes of the *Benchmark Notes* are available for subscription via membership in a EUROSIM society.

→ www.sne-journal.org

✉ office@sne-journal.org, eic@sne-journal

✉ SNE Editorial Office, c/o ARGESIM / DWH, Neustiftgasse 57-59, 1070 Vienna, Austria

SNE Editorial Board

SNE - Simulation Notes Europe is advised and supervised by an international scientific editorial board. This board is taking care on peer reviewing and handling of *Technical Notes*, *Education Notes*, *Short Notes*, *Software Notes*, *Overview Notes*, and of *Benchmark Notes* (definitions and solutions). At present, the board is increasing:

Felix Breitenecker, Felix.Breitenecker@tuwien.ac.at

Vienna Univ. of Technology, Mathematical Modelling and Simulation; Austria, Editor-in-chief

Maja Atanasijevic-Kunc, maja.atanasijevic@fe.uni-lj.si

Univ. of Ljubljana, Lab. Modelling & Control, Slovenia

Peter Breedveld, P.C.Breedveld@el.utwente.nl

University of Twente, Div. Control Engineering, Netherlands

Agostino Bruzzone, agostino@itim.unige.it

Universita degli Studi di Genova, Italy

Francois Cellier, fcellier@inf.ethz.ch

ETH Zurich, Institute for Computational Science, Switzerland

Vlatko Čerić, vceric@efzg.hr

Univ. Zagreb, Fac. of Organization and Informatics, Croatia

Russell Cheng, rhc@maths.soton.ac.uk

University of Southampton, Fac. Mathematics/OR Group, UK

Horst Ecker, Horst.Ecker@tuwien.ac.at

Vienna Univ. of Technology, Inst. f. Mechanics, Austria

Edmond Hajrizi, ehajrizi@ubt-uni.net

University for Business and Technology, Pristina, Kosovo

András Jávör, javor@eik.bme.hu,

Budapest Univ. of Technology and Economics, Hungary

Esko Juuso, esko.juuso@oulu.fi

Univ. Oulu, Dept. Process/Environmental Eng., Finland

Kaj Juslin, kaj.juslin@vtt.fi

VTT Technical Research Centre of Finland, Finland

Francesco Longo, f.longo@unical.it

Univ. of Calabria, Mechanical Department, Italy

David Murray-Smith, d.murray-smith@elec.gla.ac.uk

University of Glasgow, Fac. Electrical Engineering, UK

Gasper Music, gasper.music@fe.uni-lj.si

Univ. of Ljubljana, Fac. Electrical Engineering, Slovenia

Thorsten Pawletta, pawel@mb.hs-wismar.de

Univ. Wismar, Dept. Computational Engineering,

Wismar, Germany

Niki Popper, niki.popper@drahtwarenhandlung.at

dwh Simulation Services, Vienna, Austria

Thomas Schriber, schriber@umich.edu

University of Michigan, Business School, USA

Yuri Senichenkov, sneyb@dcn.infos.ru

St. Petersburg Technical University, Russia

Sigrid Wenzel, S.Wenzel@uni-kassel.de

University Kassel, Inst. f. Production Technique, Germany

Author's Info

Authors are invited to submit contributions which have not been published and have not been considered for publication elsewhere to the SNE Editorial Office. SNE distinguishes different types of contributions (*Notes*):

- *Overview Note* – State-of-the-Art report in a specific area, up to 14 pages, only upon invitation
- *Technical Note* – scientific publication on specific topic in modelling and simulation, 6 – 8 (10) pages
- *Education Note* – modelling and simulation in / for education and e-learning; max. 6 pages
- *Short Note* – recent development on specific topic, max. 4 pages
- *Software Note* – specific implementation with scientific analysis, max 4 pages
- *Benchmark Note* – Solution to an ARGEIM Benchmark; basic solution 2 pages, extended and commented solution 4 pages, comparative solutions on invitation

Interested authors may find further publication and layout information at SNE's website → www.sne-journal.org (templates for *Notes*, requirements for benchmark solutions, etc.).

SNE Contact & Info

→ www.sne-journal.org

✉ office@sne-journal.org, etc@sne-journal.org

✉ SNE Editorial Office, c/o ARGESIM / DWH, Neustiftgasse 57-59, 1070 Vienna, Austria

SNE SIMULATION NOTES EUROPE

ISSN SNE Print ISSN 2305-9974, SNE Online ISSN 2306-0271

WEB: → www.sne-journal.org, DOI prefix 10.11128/sne

Scope: Technical Notes, Short Notes and Overview Notes on developments and trends in modelling and simulation in various areas and in application and theory; benchmarks and benchmark documentations of ARGESIM Benchmarks on modelling approaches and simulation implementations; modelling and simulation in and for education, simulation-based e-learning; society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).

Editor-in-Chief: Felix Breitenecker, Vienna Univ. of Technology, Inst.

f. Analysis and Scientific Computing, Div., Math. Modelling and

Simulation, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria; ✉

Felix.Breitenecker@tuwien.ac.at, ✉ etc@sne-journal.org

Layout / Administration: Anna Mathe, ARGESIM / ASIM

✉ anna.mathe@tuwien.ac.at, ✉ office@sne-journal.org

Print SNE: Grafisches Zentrum, TU Vienna,

Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria

Online SNE: DWH Simulation Services, Neustiftgasse 57-59, 1070 Vienna

Publisher: ARGESIM ARBEITSGEMEINSCHAFT SIMULATION NEWS

- WORKING COMMITTEE SIMULATION NEWS, Neustiftgasse 57-59,

1070 Vienna, Austria; → www.argesim.org, ✉ info@argesim.org

on behalf of ASIM(→ www.asim-gi.org and EUROSIM

→ www.eurosim.info

© ARGESIM / EUROSIM / ASIM 2012

Energy Flows in Industrial Buildings and Machines using the Example of a Node Model

Mirko Schäfer^{*}, Johannes Wagner, Alexander Schlüter, Jens Hesselbach

University of Kassel, Institute of Production Technology and Logistics (IPL), Sustainable Products and Processes, Kurt-Wolters-Strasse 3, 34125 Kassel, Germany; ^{*}*schaefer@upp-kassel.de*

Abstract. Within the context of the research project 'HIER! - Production in Air-Conditioned Facilities', financed by the Hessian Ministry of the Environment, Power, Agriculture, and Consumer Rights (HMUELV), the Department for Sustainable Products and Processes at the University of Kassel is developing simulation modules for the energetic evaluation of production systems.

For this purpose, the development of a node model facilitates research into the thermal properties of machines and products. With a focus on energy storage, the study demonstrates a variety of applications for this model, using the example of a cold-storage facility and a plastics processing plant for the reduction of the primary energy requirement from the process to the factory.

Introduction

There is a great energy saving potential in production plants. Savings resulting from the doubling of a company's energy-related productivity with an average energy consumption of 1.4 TWh would equal the annual energy consumption of a city the size of Stuttgart. [1] Even a medium-sized company with an electricity consumption of 10 GWh can save the energy demands of up to 1700 households by raising its efficiency levels to 50%. [10] [11]

Given the increase in environmental awareness, both within society and the political sector, an energy-efficient and environmentally friendly means of production is rapidly becoming more and more important. The energy requirements of the production are determined by so-called crossover technologies, as well as by specific main technologies. Crossover technologies may perform movements within machines and plants, transmit energy, temper and control the procedure.

In most cases, means of lowering energy consumption only come into play outside of the manufacturing machines. Significant savings may be achieved through technologies with a direct energetic link to the production process. Besides their considerable potential, such

savings may even render space heating systems obsolete. The concept of energy supply in the form of cross-over technology incorporates the provision of auxiliary energy forms, such as compressed air, heat, and cold, as well as the generation of electricity from decentralised combined heat-and-power plants in conjunction with the public power grid. [1]

The thought pattern regarding this topic follows a so-called onion-layer model - from the inside out, in other words, beginning with the process, seen in Figure 1.

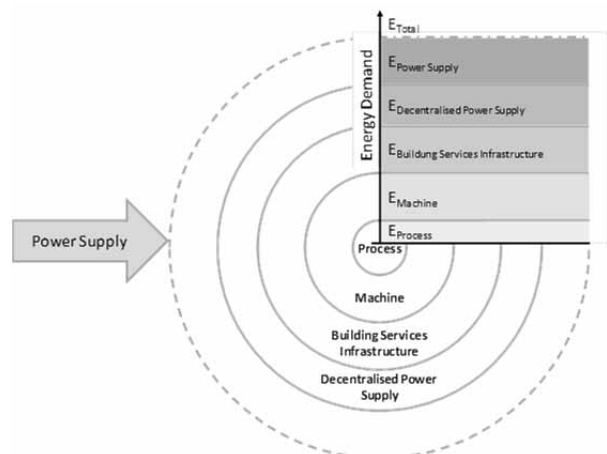


Figure 1. The onion layer model regarding the energy demand of a process [1].

In reality, the energy currents would surpass the individual layers and interact. Air-conditioned production facilities represent the extreme case of energetic linkage of production processes as well as technical building equipment as a cross-sectional task.

During winter, waste heat relieves the heating systems and during summer, every kWh of electricity registered for the proper operation of the machines leads to additional 0.33 kWh in cooling needs. [1] In some cases, such as cold-storage facilities, the energy demand for cooling purposes may even exceed 70 % of the energy demand for the entire unit. [2] In this case, the number of stored goods determines the energy consumption of

the building services equipment, which, in turn, influences the dimensioning of the entire energy supply. An approach to the energetic optimisation of a cold-storage facility would be the usage of the refrigerated products for energy storage. [1]

As manufacturing companies produce a lot of waste heat, however, the waste heat emanating from the machines influences air-conditioning technology; a heat recovery system reduces the gas requirements within the central heating plant. Awareness of the waste heat from machines, or more specifically, their heating and cooling times, allows for research into the saving of thermal energy for the lowering of energy requirements.

For the thermal analysis of processes and machines, a model, which calculates temperatures as well as currents of energy, is required. The basis for this is the node model developed by the *Department for Sustainable Products and Processes*. The contribution addresses the fundamental approach to modelling, on the basis of which the developed model is used to demonstrate its potential, identified by means of a thermal simulation, in the form of two examples.

1 Development of the Node Model: Thermal Analysis from Processes to Buildings

For the purpose of modelling and simulation of machines and facilities, energy flows for each individual energy source (electricity, heat, compressed air, etc.) will have to be observed separately.

In a practical sense, time constants may differ greatly in regards to thermal processes, due to the fact that real physical and technical processes are always subject to temporal connections. [12]

During the heating of a cold machine, electric power increases at once, whereas a change in the thermal capacity resulting from it might take hours. [5] Thus models demonstrating thermal properties differ significantly from those illustrating the necessary electrical capacity.

Within a body, thermal conduction represents an energy flow; on the surface, additional radiations and convections should be taken into consideration. [6]

A thermal analysis can be conducted mathematically with the help of location-dependant partial differential equations. However, it is possible to perform an incremental calculation with the help of the so-called node model, which links common differential equations.

The location dependency of the partial differential equations is approximated by a division into the various layers of the body. The centre of a layer is thus referred to as a node (Figure 2). [5]

As opposed to the solution of a partial differential equation, the node model reduces the required computing power and provides a substantial increase in flexibility, so that the model can easily be expanded around an internal heat source (such as the heating process of a machine) or a phase transition (such as the cooling of refrigerated goods).

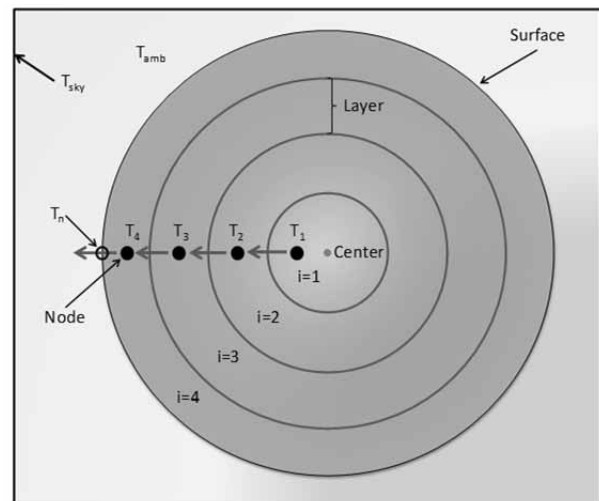


Figure 2. Sectional view of a layer model of a sphere [3].

For every layer, an assessment of mass, performance, and energy is made. Arising temperature differences cause transferred thermal energy flows through heat conduction.

The following equation shows the energy assessment of a node without an internal heat source or phase transition (Equation 1):

$$m_i c_K \frac{dT_{K,i}}{dt} = \frac{\lambda_K}{\delta} A_i (T_{K,i-1} - T_{K,i}) - \frac{\lambda_K}{\delta} A_{i+1} (T_{K,i} - T_{K,i+1}) \quad (1)$$

where T_K is the temperature of the body in K, m_i is the mass of the layer in kg, c_K is the specific heat capacity of the body in J/kgK, λ_K is the heat conductivity of the body in W/mK, δ is the distance between the nodes in m, i is the number of the layer and A is the surface measured in m².

In order to focus the calculation not on the outermost node, but on the surface layer, the latter has to be mapped through a massless node. The heat exchange between the body and the surrounding area results from

radiation and convection. In the case of many actual machines and facilities, an additional insulating layer has to be taken into account. [5] The temperature of the massless node $T_{K,n}$ is a decisive factor in the calculation of the heat exchange. The convection results from a temperature difference between $T_{K,n}$ and the factory air T_{amb} . In case of the radiation, the temperature of the surrounding wall (T_{sky}) is also relevant (Equation 2).

$$0 = \frac{2 \lambda_K}{\delta} (T_{K,n-1} - T_{K,n}) - \alpha_K A_K (T_{K,n} - T_L) - \varepsilon_K \sigma A_K (T_{K,n}^4 - T_{sky}^4) \quad (2)$$

where T_L is the room temperature in K , T_{sky} represents the radiation temperature in K , α_K is the heat transfer coefficient on the surface of the body in W/m^2K , ε_K is the emission coefficient of the body, σ is the Stefan-Boltzmann-Constant in W/m^2K^4 and n defines the outermost layer.

In reality, both machines and facilities are subject to a great number and variety of internal and external influences. [7] With the aid of the aforementioned modelling, aspects such as production cycles and alternating thermal influences within the model may be taken into consideration. This makes it possible to simulate thermal properties of the individual products, machines, and facilities.

2 Simulation of Cold Storage using the Node Model

The following example uses the node model in order to simulate frozen goods in a cold-storage facility: The frozen products include food, such as meat and fish products, but also special items like medication. This permits analysis of the thermal properties of the products, and thus their evaluation as a useful energy saving device. For this purpose research was conducted into how far the frozen goods could be further cooled down when electricity expenses are low, in order to be able to reduce the cooling during times of higher electricity rates.

A simulation using the node model is created by viewing the frozen goods geometrically as symmetrical objects. The entire heat exchange between the modelled body and its surroundings can be calculated as the sum of the convective heat flow from the surface of the body into the ambient air and the radiation exchange with the surrounding surfaces (e.g. walls and ceilings). In order to assess the thermal requirements of production facilities, the energetic correlation of weather conditions, the

building, building technology and production machines has to be realised within the model. The model's validation for use within the cooling area is assessed through a test series of the freezing- and thawing process of several different products. [3]

A simulation conducted within the context of this study compares a model with a fixed electricity rate to a model with an hourly rate (using a variable energy supply). Other options include models with a base load coverage at a fixed price and a more variable purchase for peak loads. In the case of a variable energy supply, it can be assumed that all market participants are familiar with the prices. A prognosis of the prices cannot be made at this time as they are highly dependent on the real-life availability of power plants, fuel prices, and wind forecasts, as well as the demand on behalf of potential clients. [4] An energetic linking of the products in cold-storage and the technical building services allows the identification of the necessary cooling requirements of the frozen products.

| Limiting Conditions | Description |
|---------------------|---|
| Simulation | Simulation of a summer month, accounting for different exterior air conditions. |
| Building | Length, width, height: 60 m x 80 m x 12 m Heat loss through walls: 0.5 W/m ² K |
| People | Number of employees: 10 Monday through Saturday Fresh-air content per employee: 20 m ³ /h |
| Refrigerated goods | Storage capacity: 28,800 m ³ Pallet spaces: 4,200 |
| Estimated costs | Fixed electricity rate: 100 €/MWh Variable electricity rate: 3.03 – 235.18 €/MWh (average rate: 100 €/MWh) Estimated costs in the case of both model variants include all power generation and grid usage expenses, as well as taxes. |

Table 1. Basic conditions of the simulation.

In a cold-storage facility one must constantly ensure that the temperature limits necessary to maintaining product quality are not exceeded. In the following case study the minimum temperature inside the cold-storage room is $-18\text{ }^\circ\text{C}$. Depending on their operating condition, the refrigerating machines may be turned off when electricity rates are high and may run at full capacity when rates are low in order to achieve a super-cooling of the products down to a certain temperature limit (marked at $-25\text{ }^\circ\text{C}$ in this case) and use them as an energy storage device. The estimated results of the simulation are displayed in Table 1.

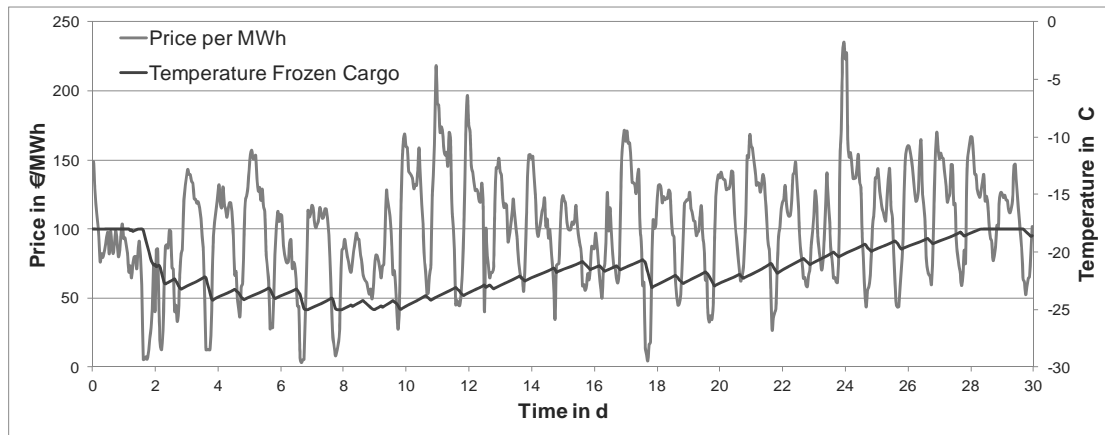


Figure 4. Temperature profile of a cold storage facility.

The result of this simulation is the temporally resolved energy necessary for the cooling of the building. This data can help in determining the cooling requirements for the entire cold-storage facility. Figure 4 shows the temperature cycles within the refrigerate goods in the case of a variable energy supply in connection with the individual electricity rates.

It becomes apparent that low electricity costs will make it possible for the refrigerated goods to be cooled down even further, so they may serve as an energy storage device during times when electricity costs are high. As opposed to the model using a fixed price for electricity (at a constant $-18\text{ }^{\circ}\text{C}$), the average room temperature of the variable model would be around $21.65\text{ }^{\circ}\text{C}$, which would invariably lead to a higher energy demand. Within the context of this observation, 37 MWh of the fixed model compares to the 41 MWh of the variable supply, whereas the expenses are lowered from $3,713\text{ }\text{€}$ to $1,854\text{ }\text{€}$ a month. This results in a savings of approximately 50% . The corresponding increase of 12% in the final energy demand may be viewed as critical.

However, under the assumption that excess electricity from renewable energy sources can be used in such a way, the energy cost-driven operation of cold-storage facilities seems practical on an energy-related level as well, since the conversion ratio of primary to final energy is decidedly better than in the burning of fossil fuels.

3 Simulation of Thermal Loads in a Plastic Factory using the Node Model

As opposed to the refrigerated goods in the cold-storage facility, injection molding machines deployed in the plastics industry have internal heat sources. These lead

to a steady heating of the machine until it reaches a static state, in which the added heating power is equal to that emitted into the room. Modelling requires the addition of internal power to the innermost junction of the machinery. Within a cylinder, a screw delivers plastic to the tool. On its way, friction within the cylinder and heat from externally applied heating collars generate the necessary energy in order to plasticise the granulate. Through a nozzle, the machine feeds the liquid mass into the tool, which then shapes the plastic during the cooling process.

Within the machine, the added electric power transforms into thermal power and splits into different parts. One part of the waste heat from the machine is discharged through water cycles, whereas the remainder is emitted into the production hall. [5] In order to increase the accuracy of a simulation in the case of very heterogeneous temperature ranges, a number of sectional modules have to be created for the various structural components and linked in order to contribute to the overall modelling. In this example, the injection moulding machine is divided into numerous spherical and cylindrical sections.

A model validation is made through the real-life operation of machines with accompanying measurements. This results in a division of the in-and outgoing energy and source streams to the individual structural elements.

The modelling was divided into the structural units of tool, extruder, control, and hydraulic system. The simulation clearly demonstrates that the dynamic properties as well as the final temperatures of the four sections differ greatly (Figure 5). In order to develop suitable solutions for an increased energy efficiency based on a simulation of an injection moulding machine, a

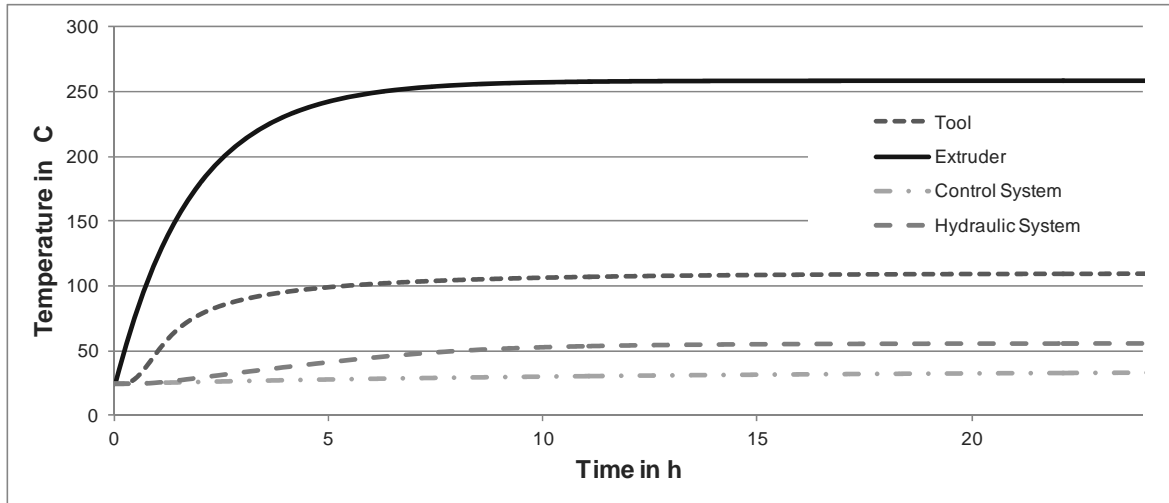


Figure 5. Simulated temperature cycles of an injection moulding machine, divided into four assembly groups.

linking of the described individual module with further components is necessary. In many production halls, for example, a surplus of heat is present due to the waste heat emitted by the machines.

Due to a temporal shift between the heat supplied within the room and the heat needed, significant thermal energy losses often occur. Furthermore, employees are often confronted with cold machines, production and offices after long switch-off periods.

A simulation conducted within the scope of this contribution compares three different heating scenarios within a plastics processing facility.

Additionally, the production plant houses injection moulding machines, which require a combined peak cooling power of about 100 kW. Due to the available thermal energy, heating the production rooms is rarely necessary. The basic approach to this is the use of waste heat for the heating of offices. An example of a building with an area of 420 m² at a room height of 2.50 m is used to demonstrate the possibilities. Production occurs in two shifts between 06:00 and 22:00, excluding Saturdays and Sundays, in order to only make the direct use of waste heat possible in this time frame. Within the simulation, an efficiency factor of 36.1 % is estimated for the transformation of primary energy into electricity [8], while an efficiency factor of 84 % is estimated for the energy chain of the natural gas necessary for the heating process. The actual state assumes a constant regulation of the room temperature at 20 °C.

In the first scenario, a night mode is implemented - in other words, the temperature outside of the operating times has to be at least 16 °C.

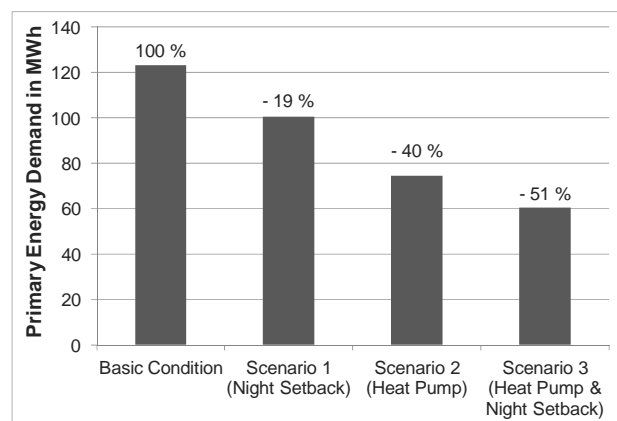


Figure 6. Primary energy demand for heating of a production hall.

In the second scenario, production management uses a heat pump, which uses energy derived from the cooling water of the injection moulding machine. During the day, this heat pump services a thermal energy storage device, while at night it releases its heat to the premises.

The third scenario combines the use of a heat pump in conjunction with the night mode. This process helps in lowering the cooling demands through the use of a heat exchange unit, which, in turn, entails limited electric energy savings. The simulation shows that using the night mode alone would help in saving around 19 % of primary energy and CO₂. Through the use of a thermal energy saving device and a heat pump, the factory can lower the primary energy demand by about 40 %. A combination of both optimisation approaches, however, will result in a saving of over 50 %, as shown in Figure 6.

4 Conclusion

The simulation of thermal processes within machines and consequently in production halls is still in its infancy. The node model shows that through the energetic interlinking of individual systems, complex thermal processes may be depicted in a sufficiently realistic manner.

The node model also represents an important basis for determining necessary cooling and heating capacities as well as room temperature peaks. Especially in the case of unsteady operating conditions (i.e. heating up a room on Monday mornings during wintertime), it provides important information for the layout of heating, ventilation, and air-conditioning technologies. Besides these layouts, both models and associated simulation runs may help assess the practicality of saving and recycling thermal energy. While conventional calculations may show that energy can be saved [9], employing the aforementioned node model may offer an additional dynamic view from the process to the production plant. The model of the cold-storage facility helps to determine the refrigerated goods' thermal properties depending on different operational stages.

This may allow, for example, the energy cost-driven operation of cold-storage facilities. The example of the synthetic plant has shown that businesses should, from a primary energy and finance-related standpoint, try to recycle their incidental waste heat for internal purposes. This may contribute to relieving the strain on existing heating systems.

Acknowledgements

The authors would like to thank the Hessian Ministry of the Environment, Power, Agriculture, and Consumer Rights (HMUELV) as well as the partners of the project 'HIER! - Production in Air-Conditioned Facilities': Limón GmbH, HSE NATURpur Energie AG and Imtech GmbH & Co. KG. The project is co-funded by the European Regional Development Fund (ERDF).

References

- [1] J. Hesselbach (ed.): *Energie- und klimaeffiziente Produktionsprozesse: Grundlagen, Leitlinien und Praxisbeispiele*. Vieweg+Teubner, Wiesbaden, 2012.
- [2] M. Weilhart: *Verbesserung der Energieeffizienz bei Kühllhäusern. Leitfaden als Hilfestellung*. KKA - Kälte Klima Aktuell, 01/2010.
- [3] L. Martin, M. Schäfer, A. Schlüter, N. Weishaar: *Energieströme simulieren: Vom Prozess bis zur Fabrik*. In: J. Hesselbach, ed.: *Energie- und klimaeffiziente Produktionsprozesse: Grundlagen, Leitlinien und Praxisbeispiel*, Wiesbaden, Vieweg+Teubner, 2012.
- [4] K. Frauendorfer, K. Kiske: *Die Spot-Volatilität der Strompreise an der EEX*. Zeitschrift für Energie, Markt, Wettbewerb, 01/2010.
- [5] A. Schlüter, M. Schäfer, J. Wagner, A. Schrod, J. Hesselbach: *Simulation von Maschinen und Anlagen als thermische Lasten in der Produktion*. In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, pp. 346-351, vol. 106, 2011.
- [6] R. Marek, K. Nitsche: *Praxis der Wärmeübertragung: Grundlagen - Anwendungen - Übungsaufgaben*. Hanser Fachbuch, München, 2007.
- [7] M. Zäh, T. Maier: *Simulation des thermischen Maschinenverhaltens*. In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, pp. 655-659, vol. 105, 2010.
- [8] Bundesministerium für Wirtschaft und Technologie: *Zahlen und Fakten - Energiedaten - nationale und internationale Entwicklung*, <http://www.bmwi.de/>.
- [9] B. Lang, A. Schlüter, M. Junge, S. Mirciov: *Simulation als Schlüssel zur Energieeffizienz*. eta energie, pp. 36-38, 01/09, 2009.
- [10] M. Becker: *Energetische Bewertung und optimierte Betriebsführung von Kälteanlagen*. KI Kälte Luft Klimatechnik, pp. 28-33, 2009.
- [11] dena: *Energieberatung in Industrie und Gewerbe. Der Schlüssel zum Kostensenken*. 2010.
- [12] H. Bode: *Matlab-Simulink*. Teubner Verlag, Wiesbaden, 2006.
- [13] M. Schäfer, A. Schlüter: *Von der Modellbildung zur Simulation: Die Vorgehensweise*. In: J. Hesselbach, ed.: *Energie- und klimaeffiziente Produktionsprozesse: Grundlagen, Leitlinien und Praxisbeispiel*, Wiesbaden, Vieweg+Teubner, 2012.

Submitted: September 2011

Accepted: February 15, 2012

Meta-Model - based Estimation of Optimal System Variants with MATLAB/Simulink

Olaf Hagendorf¹*, Thorsten Pawletta¹, Roland Larek²

¹Wismar University of Applied Sciences, Res. Group Computational Engineering and Automation (CEA), PB 1210, 23952 Wismar, Germany, *olaf.hagendorf@hs-wismar.de

²Bremen University of Applied Sciences, Foundation Institute of Materials Science, Bremen, Germany

Abstract. The idea that the efficient usage of all necessary production resources has not only an ecological importance but is also an overwhelming economical competitive factor becomes accepted since the 1980s. Modeling and simulation with integrated parameter optimization is used routinely to improve process performance. In engineering a well known environment for this task is the MATLAB/Simulink programming system. Using this or similar, established techniques only model parameter of a single model structure is optimized. Model structure is considered to be fixed as the relationships between model elements are defined during model development. Until now no tools and methods are known which can optimize product design and production processes utilizing all existing degrees of freedom. As process performance is optimized it may be necessary to redesign the model structure. The redesign is normally carried out manually by an analyst but not automatically by the optimization method. This suboptimal combination of automatic parameter optimization and manual structure changes leads to a time consuming and error-prone optimization task.

The system theoretical approach of the System Entity Structure/Model Base framework (SES/MB) is able to define alternative model structures and parameter sets in a single meta-model, called System Entity Structure (SES). Moreover, atomic models are stored in a model base (MB). Using both, SES and MB, it is possible to generate modular, hierarchical models with different structures and parameters.

Evolutionary Algorithms are a subtopic of Artificial Intelligence that are involved in combinatorial optimization problems. These algorithms are based on ideas inspired by biological evolution. They often perform well for many problem types because they do not make assumptions about the problem specific search space.

The research reported in this paper details an approach providing optimization through automatic reconfiguration of both: model structure and model parameters.

An evolutionary algorithm based optimization method is assisted by an SES/MB based model management. It searches for an optimal solution with repeated, combined model parameter and model structure changes resulting in a combined parameter and structure optimized model

Introduction

Modeling and simulation with integrated optimization is a well-established technique in engineering applications. Such techniques are used for system design, real time planning and to control production systems. The MATLAB / Simulink environment is one of the well known and widely used modeling and simulation tools in this application field [3 7 8].

With increasing complexity and flexibility of systems the requirements for modeling and simulation tools are growing. Existing applications using simulation based optimization are restricted to parameter optimizations. The user has to change model structure manually and repeat optimizations until a suitable solution is found. With increasing production system flexibility the number of possible structure variants increases and the potential benefit of automatic model structure optimization becomes significant.

The focus of this paper is the description of a methodology for a combined parameter and structure optimization for modular, hierarchical systems, which is implemented and tested in the MATLAB/Simulink environment but not restricted to it. In contrast to the common usage the model structure and model parameter values are controlled together by a super ordinate optimization module. To support the optimization method appropriate model management and generation methods are necessary.

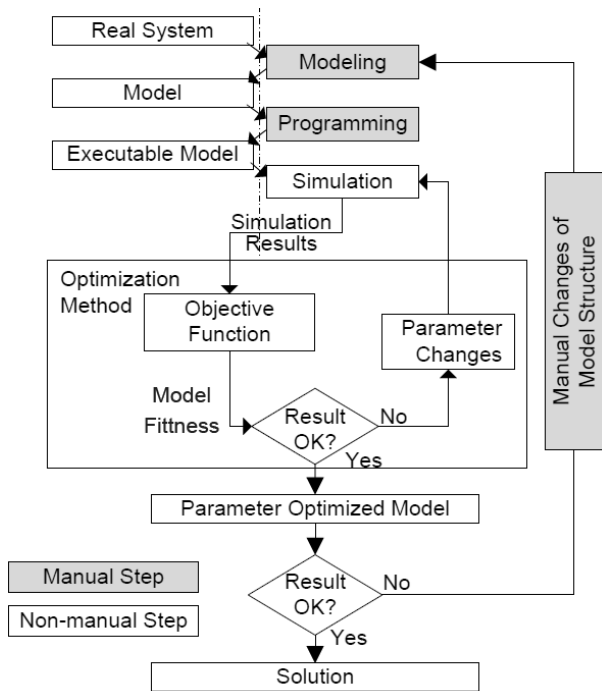


Figure 1. Simulation based parameter optimization.

As a basis for model management and generation the System Entity Structure/Model Base (SES/MB) formalism, introduced by Rozenblit, Zeigler et al [4 5 6], is employed. The SES/MB formalism is a generic, knowledge based framework consisting of a tree like entity structure and a model base. With its features the framework is able to define a set of modular, hierarchical models and to generate specific parameterized model structures.

Section 1 provides a short summary to the well-known simulation based parameter optimization approach. In the following the basic ideas of the advanced simulation based optimization approach are introduced. The synthesis of three fundamental elements: (i) optimization algorithm, (ii) model management and model generation and (iii) simulation execution to a framework for a combined structure and parameter optimization of complex modular, hierarchical systems is presented.

Finally, the new ideas are illustrated by a small logistical problem in Section 2. The implementation in MATLAB / Simulink is sketched. The conclusion refers to real world problems solved by the introduced approach.

1 A Framework for Parameter and Structure Optimization

1.1 Simulation Based Optimization

Simulation experiments can be of different complexity. The least complex ones are ordinary simulation runs. The modeling and simulation method is highly dependent on the type of the real system, the analysis objectives and the used software environment. In MATLAB / Simulink the programming step from the user's point of view consists of a graphical based model specification and often it is considered as part of the modeling. However, after examining simulation results the analyst manually changes model parameter values and/or model structure and starts the simulation again. These steps are repeated until a suitable solution is found. A more complex approach is simulation based parameter optimization, depicted in Figure 1.

The optimization method alters model parameter values to improve the result of an objective function until a stop criterion is fulfilled. The result is a parameter optimized model. Structure changes are carried out manually by a user followed by a repetition of the automated parameter optimization.

It is important to note that automatic structure changes induced by an optimization method are not possible with this approach. Instead, structure changes are carried out manually by a user. Each structure change requires a repetition of the parameter optimization. This principle is typical when using MATLAB / Simulink in a simulation based optimization. Therefore a Simulink model with a fixed structure is embedded into a super-ordinate optimization method which can alter specific Simulink block parameters to find the optimal values. Structure changes are done manually.

1.2 Principle of the Framework with combined Parameter and Structure Optimization

The idea behind this research is the extension of the simulation based parameter optimization method with the ability to additionally change the model structure thus improving the objective function result. The effect of this extension is a simulation based structure and parameter optimization.

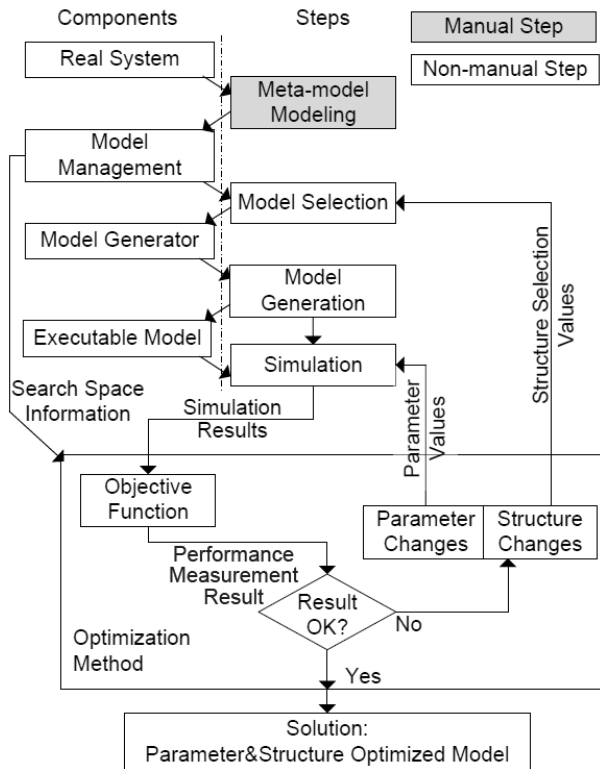


Figure 2. Principle of a simulation based parameter structure optimization.

Figure 2 presents the approach in principle with its combination of three methods: (i) a meta-model framework for model management, (ii) a modeling and simulation environment and (iii) an optimization method. In contrast to the established approaches:

1. The optimization method controls both: the model parameter values and the model structure, changing both until a stop criterion is fulfilled. The result of this process is a combined parameter and structure optimized model.
2. The user has to organize a set of models. One possibility is to define a model which describes a set of model variants instead of one single model of the real system. Such models that define the creation and interpretation of a set of models are named meta-models. Through this inclusion of the meta-model based automatic model generating element the optimizer can additionally control model structure changes to find an optimized solution.

Crucial parts and algorithms of this approach are described in the next sections.

1.3 Specification of Model Sets with SES/MB

As an appropriate meta-modeling framework the System Entity Structure/Model Base (SES/MB) formalism was determined. This formalism is a general, knowledge based framework. With its key feature to depict the three relationships (i) decomposition, (ii) taxonomy and (iii) coupling it is capable of defining a set of modular, hierarchical models [4 5 6].

Decomposition means that the formalism is able to decompose an object into sub-objects. Taxonomy means the ability to represent several, possible variants of an entity. Composition of an entity from sub-entities is done by coupling.

A SES/MB meta-model consists of two major parts: (i) a system entity structure (SES) and (ii) a model base (MB):

- The SES is a tree like structure which contains invariable and variable branches. The variable branches start at decision nodes. To create one structure variant the entity structure is pruned. At each decision node a specific branch is chosen. The result is a pruned entity structure (PES) which is the basis to create a hierarchical model that can be simulated.
- A SES can contain different node types and additionally, nodes can have optional and/or obligatory properties. One node type, the atomic entities have their counterparts in the MB.
- The MB contains a corresponding basic model for each atomic entity. The types of the basic models can vary. It depends only on the underlying simulation method e.g. in this research the model base matches the Simulink library.

A detailed description of all elements can be found in reference [1].

1.4 Optimization Process

Before an optimization can be carried out, information about the search space, in particular its dimensions and domains, is necessary. In this approach the search space is defined by the set of model structure variants established by analyzing the SES and the set of model parameters, defined by each model structure.

During the optimization process several points in the search space are examined. Each point defines one single model structure to be generated through the model generator with one set of parameter values.

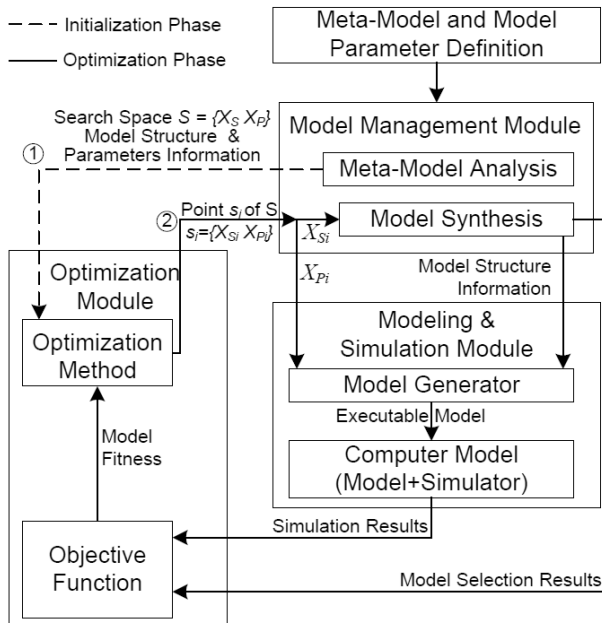


Figure 3. Structure and operation of the framework.

On closer examination of the framework, as depicted in figure 3, it is crucial to divide an optimization experiment into two phases:

1. *Initialization phase*: The model management reads and analyzes a meta-model. Results of the analysis are information about the multidimensional search space $S = (X_S, X_P)$. The optimization module is initialized with this information.
2. *Optimization phase*: During the optimization phase the search space is explored within a loop. Each examined search space point, i.e. an ordered set of values (X_{S_i}, X_{P_i}) is delivered to the model management module. This module starts up the processes: structure synthesis, model generation, simulation and performance estimation. The optimization loop ends when a predefined stop criterion is fulfilled.

1.5 Interfaces: Optimization Module – Model Management Module

Crucial parts of this framework are the interfaces (1) SES tree analysis during initialization phase and (2) transformation (search point + SES) → PES during optimization phase both depicted in Figure 3.

Within interface (1) the Model Management Module has to analyze the SES tree to transform formal meta-model structure information into numerical data useable by the Optimization Module.

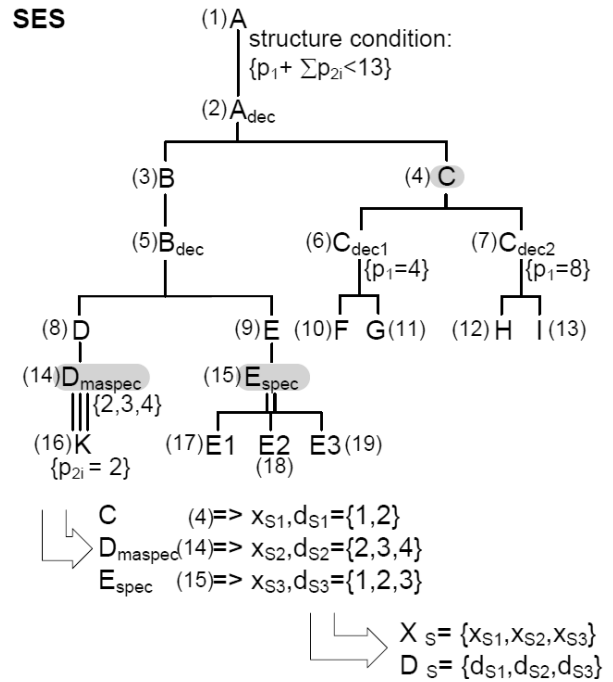


Figure 4. Transformation SES → sets $X_S + D_S$

This is done by a tree analysis starting at the root node, traversing the tree in a defined direction, namely depth-first and breadth-first analysis, and considering every node. If a node contains variable branches a structure parameter x_{S_i} is added to the structure parameter set X_S with a corresponding domain d_{S_i} added to the domain set D_S .

Figure 4 illustrates the algorithm for creating these sets using a breadth-first analysis. The steps of build-up order are marked with small sequence numbers. Starting at node A, the composite entity node C is the first decision node. A first parameter x_{S_1} is added to the search space X_S with the domain $d_{S_1} = \{1,2\}$ because it has two alternative successors.

The next decision node is D_{maspec} with a number range property $\{2,3,4\}$. A second parameter x_{S_2} is added to X_S with the domain $d_{S_2} = \{2,3,4\}$. The next node, the specialization node E_{spec} , is again a decision node. It has three alternative successors. A third parameter x_{S_3} is added to X_S with domain $d_{S_3} = \{1,2,3\}$.

The remaining nodes are non-decision nodes. Thus, the resulting structure parameter set is $X_S = \{x_{S_1}, x_{S_2}, x_{S_3}\}$ with the corresponding domain set $D_S = \{d_{S_1}, d_{S_2}, d_{S_3}\}$. These sets together with the model parameter and domain sets X_P, D_P define the search space S.

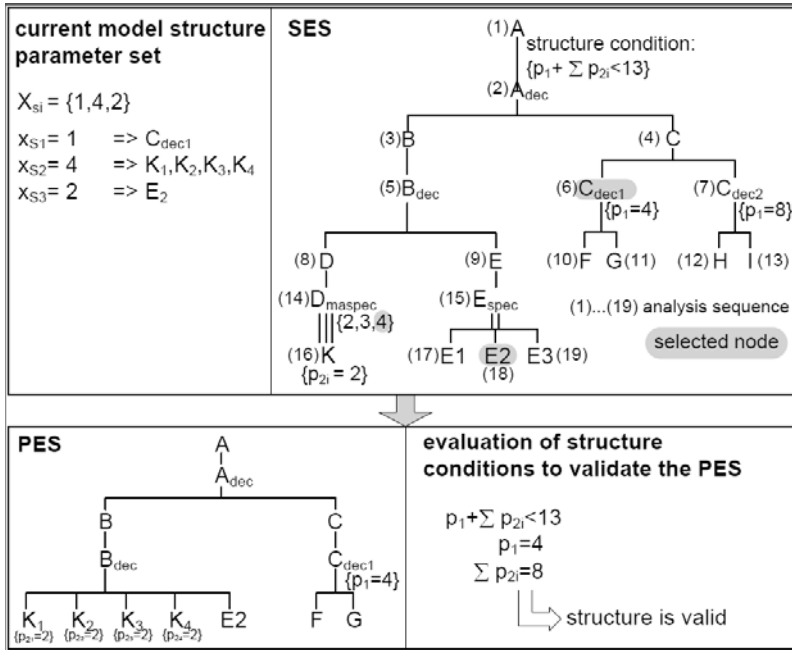


Figure 5. Transformation $X_{Si} + SES \rightarrow PES$.

The second transformation, the line (2) in Figure 3, is the reverse of the first. The Model Management Module gets a point s_i in the search space from the Optimization Module i.e. the numerical data set $X_i^* = X_{Pi} \cup X_{Si}$, where set X_{Si} codes a specific model structure and set X_{Pi} codes its model parameters. It has to synthesize the corresponding model structure and to infer the model parameters. The transformation traverses the tree using the same direction as during the first transformation.

Figure 5 illustrates the principle of the second transformation. The analysis and pruning order is marked with sequence numbers. The breadth-first analysis starts at the root node A. The first decision node is node C. The first element of X_{Si} is $x_{S1} = 1$, i.e. the first aspect node C_{dec1} is chosen for the PES. The next decision node is node D_{maspec} and the corresponding set element is $x_{S2} = 4$, i.e. the PES contains four nodes K.

The last decision node is node E_{spec} and the corresponding set element is $x_{S3} = 2$, i.e. the PES contains the second specialization E2. After pruning, the PES is verified by evaluating structure conditions, indicated at node A in Figure 4 and Figure 5. Afterwards the model generator uses the PES to generate an executable simulation model utilizing basic models stored in the MB.

As mentioned, the result of the model synthesis and model generation is an executable simulation model with a specific model structure and specific model parameter values. Simulating this model delivers the simulation results of a single simulation run. In case of stochastic parameters simulation runs have to be repeated. The simulation results or the mean values are the input for the objective function.

2 Application: Process Planning

The chosen example is a typical optimization problem of a job scheduling system with two variable properties: the job arrival time and the number of facilities. The complexity of the system is defined consciously simple to demonstrate the introduced approach.

The introduced concept of a simulation based parameter and structure optimization was implemented in the MATLAB/Simulink environment. The necessary modeling and simulation method is provided by the MATLAB/Simulink system and the additional MATLAB toolbox SimEvents itself.

In [1] a MatlabSES toolbox was described. This toolbox handles SES meta-models described by XML files and implements the algorithms explained above. During this research a new model generator has been implemented. It uses Simulink blocks i.e. original blocks from the Simulink libraries or self-implemented Simulink blocks as basic models which are referenced by the SES meta-model.

MATLAB incorporates an Optimization toolbox [9] containing an EA implementation. The MatlabSES model synthesis and the Simulink model simulation are called through function pointer by the EA toolbox.

Figure 6 depicts one variant of the process implemented as a Simulink model. The variable model parts are marked with boxes.

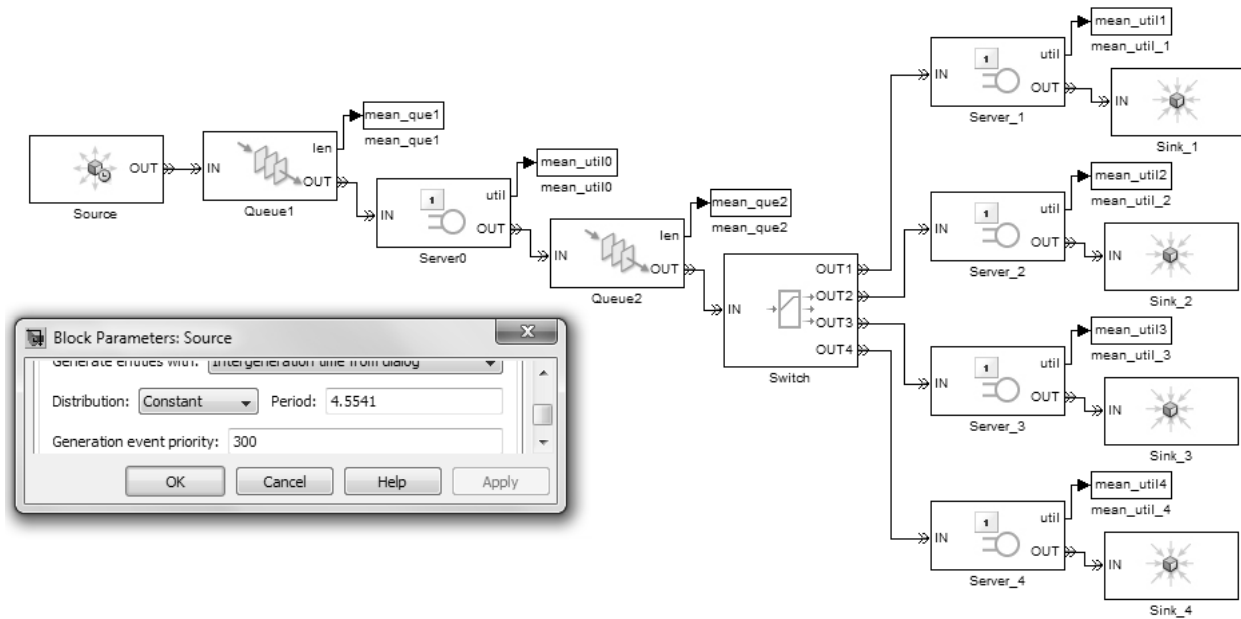


Figure 6. One variant of the example application.

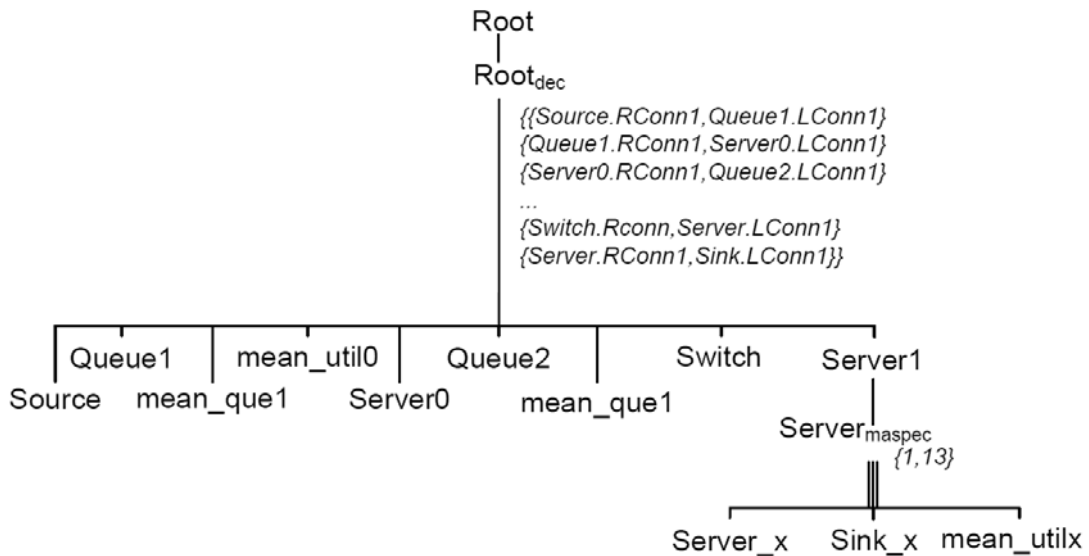


Figure 7. SES of the example application.

1. The Simulink block *Source* generates jobs with a variable arrival time within the uniform range 4..20s. The value is defined by the Simulink block parameter *Period*.

2. The *Switch* block distributes the jobs to a variable number of servers. The example in figure 6 shows 4 servers. The number is defined within the range 1..13 according with the SES in Figure 7.

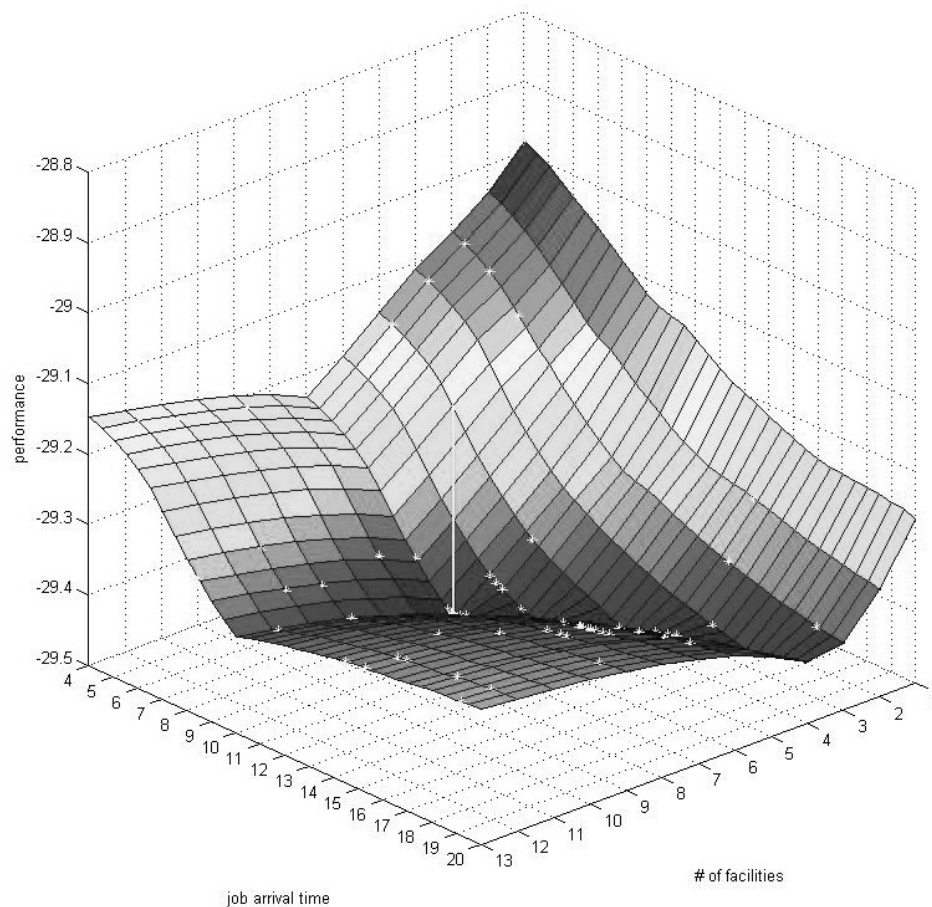


Figure 8. SES of the example application.

The *Server0* has a job handling time of 10s, all other servers have a handling time of 70s. The optimum of the system is defined as the maximal number of handled jobs in a given time.

Through the straightforwardness and the deterministic parameters of the example the optimal parameter and model structure can be taken out directly:

- job arrival time = 10s
faster arrivals fill only *Queue1*
slower arrivals cause *Server0* wait states
- number of server = 7
fewer servers fill only *Queue2*
more servers cause wait states of one or more servers

With the common procedure (see Figure 1) an analyst would repeat the cycle:

- automatic searching of the optimal model parameter for the job arrival time for a single model structure
- evaluating the model performance
- when the performance is not satisfying: changing the structure of the Simulink model and starting again
- ... until finding a satisfying solution.

This method is time-intensive and don't guaranty the finding of an optimal solution. A further disadvantage is necessary profound knowledge of the analyst about the system to guess promising structure changes for the next step.

In contrast to this common principle, using the introduced approach an analyst doesn't implement the Simulink model but determines/creates basic blocks from the Simulink Library and specifies the SES which defines all model variants within a single meta-model. Figure 7 depicts the SES of the example.

As section 1.4 describes, the next step of the algorithm is analyzing the SES tree and generating information about the search space. The analysis finds one structure parameter at the decision node *Server_{maspec}*. Together with the continuous model parameter *job arrival time* the optimization method has to search in a two-dimensional search space. Figure 8 depicts the fitness surface within the search space. Each point on this surface corresponds to one specific Simulink model structure and Simulink Source block *Period* parameter value combination.

During the optimization phase the EA estimates the performance of randomly selected points on the surface to find the optimal solution. In Figure 8 dots on the surface illustrate analyzed points in one experiment. The figure also shows the found optimal model parameter and model structure combination with a line at the expected location: job arrival time = 10 and number of servers = 7. The experiment was repeated and the average number of simulated search space points is 240.

3 Conclusions and Further Work

This paper has introduced a simulation based combined structure and parameter optimization method. To achieve this aim the approach combines three established methods and extends established optimization techniques to the fundamental model structure to enable combined structure and parameter optimization.

Three main elements have been determined: (i) a model generating meta modeling technique based on SES/MB formalism, (ii) a MATLAB/Simulink based modeling and simulation environment, (iii) an EA optimization method. The introduced approach was implemented and tested in the Scientific and technical Computing Environment MATLAB/Simulink using the additional MATLAB toolboxes SimEvents and Global Optimization.

The results of the presented application show that the approach is able to find an optimal model structure with appropriate model parameters.

The complexity of the shown example is defined consciously small to demonstrate the introduced approach. In [1 2] examples with a higher complexity but another modeling and simulation method are described. The presented framework and its application are an intermediate result in the scope of the research project supported by the German Research Foundation.

References

- [1] O.Hagendorf. *Simulation Based Parameter and Structure Optimisation of Discrete Event Systems*. PhD thesis, Liverpool John Moores University, 2009.
- [2] O. Hagendorf, T. Pawletta, C. Deatcu, R. Larek: *An approach for combined simulation based parameter and structure optimization using evolutionary algorithms*. In: Proc. of The 7th EUROSIM 2010 Congress, Vol.2, Prag, Czech Republic, 2010
- [3] T. Pawletta, C. Deatcu, O. Hagendorf, S. Pawletta, G. Colquhoun. *DEVS-Based Modeling and Simulation in Scientific and Technical Computing Environments*. In: Proc. of DEVS Integrative M&S Symposium (DEVS'06) - Part of the 2006 Spring Simulation Multiconference, Huntsville/AL, USA, 2006.
- [4] J.W. Rozenblit, B.P. Zeigler. *Concepts for Knowledge-Based System Design Environments*. In: Proc. of the 1985 Winter Simulation Conference, San Francisco/CA, USA, 1985.
- [5] B.P. Zeigler, H. Praehofer, T. G. Kim. *Theory of Modeling and Simulation*. Academic Press, 2000.
- [6] G. Zhang, B.P. Zeigler. *The system Entity Structure: Knowledge Representation for Simulation Modeling and Design*. Artificial Intelligence, Simulation, and Modeling, Widman L.E., Loparo K.A., Nielsen N.R. (Ed.), John Wiley & Sons Inc, 1989.
- [7] The Mathworks® (2011) *MATLAB*®. <http://www.mathworks.com/products/matlab/>.
- [8] The Mathworks® (2011) *SIMULINK*®. <http://www.mathworks.com/products/simulink/>.
- [9] The Mathworks® (2011) Global Optimization Toolbox. <http://www.mathworks.com/products/global-optimization/>.

Submitted: September 2011

Revised: February 16, 2012

Accepted: March 1, 2012

A Qualitative Comparison of Two Hybrid DEVS Approaches

Christina Deatcu^{*}, Thorsten Pawletta

Wismar University of Applied Sciences, Res. Group Computational Engineering and Automation (CEA),
PB 1210, 23952 Wismar, Germany; ^{*}*christina.deatcu@hs-wismar.de*

Abstract. A hybrid system is defined as a system with mixed discrete event and continuous parts. Two approaches for modeling and simulation of hybrid systems in the context of the Discrete Event System Specification (DEVS) formalism are compared. Since the system theoretic DEVS approach has been introduced during the 1970ies, it has been completed by different extensions. The hybrid extensions checked against each other here, are the Quantized State Systems (QSS) method and a wrapper concept based on traditional ODE solvers. For a comparison from engineering practice point of view, these two hybrid DEVS approaches are analyzed within the scientific and technical computing environment MATLAB.

Introduction

There are several formal descriptions of system structures and system behaviors on hand to support the precise and complete mapping of real systems. This work is based on the Discrete Event System Specification (DEVS) and its associated abstract simulator algorithms. We focus on applying DEVS-based approaches in the engineering field and in doing so try to find reasons why DEVS is relatively unknown and rarely used among engineers.

Initial research work on DEVS formalism was done in 1976 by Zeigler [1]. The DEVS formalism offers comprehensive options for describing systems that change their state according to events that appear at discrete instances of time. A main characteristic of simulation models defined according to the DEVS formalism is their modular hierarchical layout. In the decades that followed, the classic DEVS formalism was enhanced with extensions for hybrid, structure variable, parallel, real time and other system types to provide options for building models of a large range of diversity [2].

Two possible approaches for hybrid DEVS are discussed here. Hybrid systems in this context are systems with mixed discrete event and continuous system parts. It is generally accepted that modeling hybrid systems can be approached according to different worldviews.

The first, the worldview of continuous systems, maps discrete events to zero-crossing problems. The entire model is computed by a continuous simulator, i.e. by an ODE solver with event detection and event localization. Approaching the issue of hybrid systems from the second, the discrete event worldview, requires the extension of discrete event formalism in a way that allows continuous model parts to be computed. This method is convenient in case the systems that need to be modeled are of a discrete event nature and include only minor continuous parts. The simulation can be controlled by a discrete event-based simulator that calls an ODE solver between event times. The Prerequisite is the availability of a closed representation of all continuous model parts as a differential equation system. However, the usual flattening of a modular hierarchical model to achieve a closed equation-oriented representation causes the loss of important model structure information during simulation runtime.

This contribution focuses on analyzing methods for hybrid system simulation within the DEVS formalism where model structure information is kept during simulation runtime. Keeping structure information is necessary to allow modular hierarchical systems with structure variability following [3, 4] to be investigated. One way to integrate continuous model parts in the DEVS formalism without model flattening is the *wrapper concept* proposed here. Thereby, an extended DEVS simulator controls the overall simulation and cyclically calls an ODE solver between discrete event times. The so-called wrapper generates the required closed representation of the differential equations and in doing so does not affect structure information of the modular hierarchical DEVS model.

In contrast to the wrapper concept, the second approach discussed here, the Quantized State Systems (QSS) method solves the problem on the model level instead of on the simulator level. With the QSS method Kofman [5, 6, and 7] proposes a quantization of the state variables instead of the traditionally accepted discretization of time to solve a system of differential

equations by numerical approximation. This approach allows asymptotic solutions by special DEVS model components. Adjusting the DEVS formalism itself is not necessary. Therefore, a hybrid model based on the QSS approach can be computed by any standard DEVS simulator.

In the following, after a short introduction to DEVS formalism, we discuss the two hybrid approaches in more depth. Based on the detailed examination, the suitability of the approaches for engineering problems is evaluated from the practitioner's point of view. Both approaches are studied by means of prototypical implementations within the scientific and technical computing environment (SCE) MATLAB. Numerical aspects like accuracy or stability problems of the examined methods are explicitly not subject to discussion.

1 Overview of DEVS

Models designed as formal DEVS models are composed in a modular hierarchical manner. They consist of two basic model types. The first type, in which dynamic behavior is defined, is the *atomic* DEVS model. The second type, the *coupled* DEVS model, describes a set of interacting components. These components are other coupled DEVS models or atomic DEVS. Beside model definitions, the DEVS formalism comprises data structures and algorithms that allow the hierarchical DEVS model to be simulated. For simulation purposes, a *simulator* is assigned to each atomic model and a *coordinator* is assigned to each coupled model as depicted in Figure 1. Simulators and coordinators are responsible for executing the model.

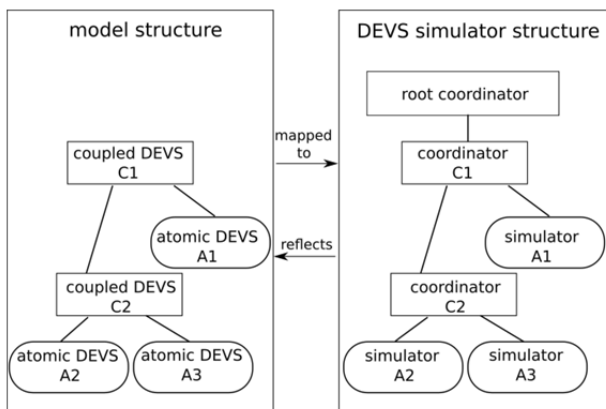


Figure 1. Model components and associated simulators / coordinators according to Zeigler.

The conjunction of a model component and a simulator or a coordinator is depicted as a *simulation object* below. The superordinate *root coordinator* initializes all simulation objects and starts and controls the simulation run. The root coordinator and the associated simulators and coordinators communicate with one another through messages.

1.1 Classic DEVS

A classic atomic DEVS A and a classic coupled DEVS N are defined as follows [2]:

$$A = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, ta) \quad (1)$$

$$N = (X_N, Y_N, D, \{M_d \mid d \in D\}, EIC, EOC, IC, Select) \quad (2)$$

An atomic DEVS A is described by the set of input events X , the set of output events Y , the set of states S , and by the four characteristic functions: internal transition function δ_{int} , external transition function δ_{ext} , output function λ and time advance function ta . The characteristic functions specify the dynamics of the atomic DEVS. Accordingly, sets of input and output events X_N and Y_N are defined for the coupled DEVS N . The set M_d specifies the subcomponents, which may be atomic or coupled DEVS. The set D is the corresponding index set.

The way in which subcomponents are connected is described by the coupling relations *EIC* (External Input Coupling), *EOC* (External Output Coupling) and *IC* (Internal Coupling). The *Select* function controls model execution if simultaneous events appear. Hence, an atomic DEVS is responsible for describing of the dynamics of each component, whereas a coupled DEVS represents the composition of components.

1.2 PDEVS

To break up conflicts caused by simultaneous events solely at the level of atomic models, the classic DEVS formalism has been enhanced mainly by Chow and Zeigler to parallel DEVS (PDEVS) formalism since 1994. Atomic and coupled PDEVS are defined as follows:

$$A_P = (X, Y, S, \delta_{\text{int}}, \delta_{\text{ext}}, \delta_{\text{conf}}, \lambda, ta) \quad (3)$$

$$N_P = (X_N, Y_N, D, \{M_d \mid d \in D\}, EIC, EOC, IC) \quad (4)$$

In classic DEVS simultaneous events in subcomponents of a coupled system need a definition of priorities within the *Select* function of the coupled model at superordinate level. The definition of priorities and therefore the *Select* function can be omitted in PDEVS. Instead, the dynamic description of the atomic PDEVS is complemented by the confluent function δ_{conf} . This function is called, if for an atomic PDEVS A_p one or more external events and an internal event occur at the same time. Additionally, simulator and coordinator algorithms were comprehensively modified for executing PDEVS models.

1.3 DEVS with structure variability

Structure variability in the context of this work is defined as the possibility of modifying the hierarchical model structure during simulation runtime. It includes the creation, deletion and exchange of model components or the modification of coupling relations.

Barros [3] proposes the addition of a special atomic model called network executive to each coupled model which holds the structure information and describes the structure dynamics. In contrast to this approach, we favor extending the PDEVS coupled model definition in a way that allows holding structure dynamics specification directly in the coupled model according to [4].

This preference of keep structure information at the coupled model level is based on the principles of the primal DEVS theory where the atomic level specifies dynamics and the coupled level specifies structure issues. In both approaches the atomic model definitions remain unchanged.

We call the current composition of a coupled system, which means its set of subsystems and its coupling relations, a structure state $s_i \in \mathcal{S}_N$. A structure variable PDEVS coupled model can have different structure states $s_0, s_1, \dots, s_n \in \mathcal{S}_N$. Furthermore, structure dynamics information, e.g. the number and kind of structure changes already achieved, need to be stored. The set of structural variables H_N holds this information. A structure state s_i , which is an element of the set of sequential structure states \mathcal{S}_N is defined as follows:

$$s_i = (H_N, D, \{M_d \mid d \in D\}, EIC, EOC, IC) \quad (5)$$

The set of sequential structure states extends the formal definition of coupled PDEVS without structure variability so that structure variable PDEVS can be defined as follows:

$$N_{pdy} = (X_N, Y_N, \{d_N\}, S_N, \delta_{Nint}, \delta_{Nnext}, \delta_{Nconf}, \lambda_N, ta_N) \quad (6)$$

It should be noted that coupling information is now encapsulated in the set of structure states \mathcal{S}_N . The name of the coupled system is stored in d_N . Furthermore, the functions δ_{Nint} , δ_{Nnext} , δ_{Nconf} , λ_N , and ta_N for coupled PDEVS systems are introduced to provide operations similar but not identical to atomic systems.

In analogy to event-oriented dynamic behavior of atomic PDEVS systems, dynamic structure changes in coupled DEVS are induced by events. The characteristic functions describe reactions on structure events. An application example for dynamic structure DEVS in engineering area that deals with structure and parameter optimization is given in [8].

2 Wrapper Concept

Scientific and technical computing environments (SCEs) such as the widely-used SCE Matlab offer a large number of established numerical methods for solving differential equations. The wrapper concept aims to make these methods applicable to hybrid, modular hierarchical DEVS models. A wrapper concept which is adapted to PDEVS is presented here. PDEVS is chosen because it offers a way of achieving an easy and realistic mapping of simultaneous events.

To keep structure information available and modifiable, the *flattening* which is typically done before computation of hybrid models must not be realized. This is a strong motivation for the wrapper concept. The issue of the structure variable PDEVS extension is sketched out here but not discussed in detail for reasons of clarity. However, it should be kept in mind that the wrapper approach is applicable to static structure as well as to structure variable models.

2.1 Formal Model Definitions

A hybrid classic atomic DEVS was defined for the first time in the 1990ies by Prähofer [9]. To allow computation of hybrid atomic DEVS Prähofer added the explicit

Euler method to the DEVS simulator.

Advanced numerical methods as they are common in engineering applications like implicit integration algorithms, predictor corrector methods, or ODE solvers with adaptive step size cannot be used. The wrapper concept starts from Prähofer’s formal hybrid atomic DEVS definition and adapts it to PDEVS formalism.

For the wrapper concept, just the formal model specification resembles Prähofer’s definition, whereas the simulation approach and algorithms are completely different. We define a hybrid atomic PDEVS as follows:

$$A_{hybridp} = (X, Y, S, f, c_{se}, \lambda_c, \delta_{state}, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda_d, ta)(7)$$

The sets of inputs X , outputs Y and states S of hybrid DEVS may contain discrete as well as continuous values. The continuous dynamics of a component is mapped by the rate of change function f and the continuous output function λ_c . Discrete events can be external or internal events, or can be state events caused by discontinuities in the continuous parts. The state event condition function c_{se} defines the conditions under which a state event is generated.

External events, internal events, and state events alter the system’s state stored in S . External events achieve this via the external transition function δ_{ext} while state events are computed within the state event transition function δ_{state} . Internal events activate the discrete output function λ_d first and subsequently the internal transition function δ_{int} , which then alters the system’s state by manipulating S .

For hybrid PDEVS not just external and internal events may occur simultaneously. So can state events in continuous variables with discontinuities detected by the ODE solver. Therefore, the confluent function δ_{conf} needs to be adapted to handle three different types of potentially simultaneous events. After any change of state the time advance function ta calculates the time interval until the next internal event.

The specification for coupled models needs to be modified marginally. A coupled structure variable PDEVS N_{pdyn} following the specification in equation (6) is suitable for hybrid models after extending its

δ_{Next} function in a way that it can react on state events occurring in continuous parts of submodels, too. Furthermore, coupling relations encapsulated in the set of structure states S_N need to be differentiated between discrete and continuous relations. In addition, the sets of inputs X_N and outputs Y_N may contain discrete as well as continuous values.

2.2 Extended Simulator Algorithms and Data Structures

In a hybrid, modular hierarchical model, continuous model parts are typically distributed over several model components. To allow the deployment of advanced numerical integration methods for computation without a previous model flattening, these distributed continuous model parts need to be collected, united and provided in a way suitable for the chosen ODE solver’s interface. The *ODE wrapper* function supplies the closed representation of continuous model parts required by the ODE solver. Coevally, structure information remains available. The modular hierarchical model itself is not modified. Rather, a closed representation of continuous model parts is generated additionally during simulation runtime.

Figure 2 and Listing 1 show how the DEVS simulation environment is extended by adequate data structures, an adapted *root coordinator*, and the *ODE wrapper* function.

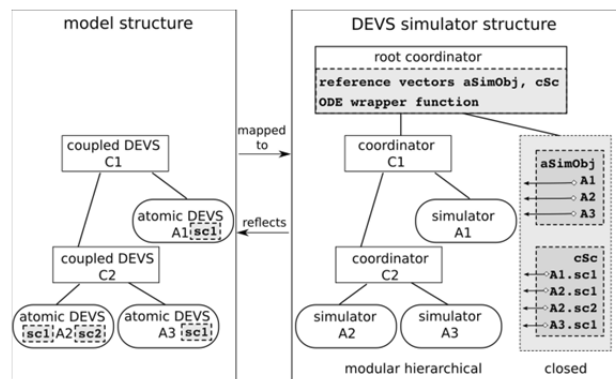


Figure 2. Modular hierarchical model structure and simulator structure with wrapper components.

The *root coordinator* needs to be extended to operate in three different phases: i) initialization phase, ii) discrete phase, and iii) continuous phase.

```

begin root coordinator

set current simulation time to start time
set phase to 0 # start the initialization phase

while simulation time not over

  switch phase

  -----

  case 0 # initialization phase
  # determine time of first internal event
  tnext <- send i-message to outermost coupled

  if tnext == current simulation time
    set phase to 1 # start a discrete phase
  elseif tnext > current sim. time
    set phase to 2 # start a continuous phase
  end

  -----

  case 1 # discrete phase
  # carry out internal event and
  # determine time of next internal event
  tnext <- send *-message to outermost coupled

  if tnext == current sim. time
    set phase to 1 # remain in discrete phase
  elseif tnext > current sim. time
    set phase to 2 # start a continuous phase
  end

  -----

  case 2 # continuous phase
  # establish references to atomic sim. objects and
  # cont. state variables in vectors aSimObj and cSc
  [aSimObj, cSc] <- send z-message to outermost coupled

  # establish direct links between outputs and inputs
  send z2-message to outermost coupled

  sort vector aSimObj depending on moore/mealy behavior

  # specify ODE-wrapper with flag=0 as event function
  set options for ODE-solver

  # call ODE-solver with ODE-wrapper, flag=1,
  # as model function
  # integration interval:
  # [current sim. time, tnext or tfinal]
  [times, values, events] <-
    ODE-solver(ODE-wrapper(1, aSimObj, cSc), interval)

  # set current sim. time to biggest value in times
  current sim. time = max(times)

  if ODE-solver has returned state events
    # send state-event-message to
    # localize atomic obj. where event has occured.
    # simulator of this atomic calls its deltastate func.
    # determine time of next internal event
    tnext <- send se-message to outermost coupled
  end

  if tnext == current simulation time
    set phase to 1 # start a discrete phase
  else
    set phase to 2 # remain in continuous phase
  end

  -----

end switch

end while

end root coordinator

```

Listing 1. Root coordinator algorithm.

In addition to the well-known messages from traditional DEVS simulators, i-message, *-message, x-message, and y-message, we introduce a z- and a z2-message for vector configuration purposes and an se-message for handling state events of the continuous model parts. For exact mapping of simultaneous events we use an interpellation y-message as proposed by [10].

At startup of a simulation run, the initialization phase is entered and the model is initialized by an initialization message (i-message) sent by the *root coordinator* to the outermost coupled model. The initialization message is passed down the hierarchy among the simulation objects until it reaches the leaves of the object tree.

Based on the minimum of time stamps for the next internal events (as they are returned by the subcomponents), the *root coordinator* determines whether a discrete or a continuous simulation phase has to be started next. If there is no imminent internal event at current simulation time, a continuous cycle is initiated and starts with an update of the wrapper data structures. This step is necessary for modular hierarchical systems with structure variability at the beginning of each continuous phase because the model structure may have changed due to a previous event.

For this purpose the *root coordinator* sends recursive configuration messages (z- and z2-message) to the associated coordinator of the outermost coupled model. Those messages are passed down the hierarchy tree depicted in Figure 2. References to all hybrid atomic simulation objects (aSimObj) as well as direct references to all continuous state variables contained (cSc) are returned. Furthermore, direct links between continuous output and input variables are established by interpreting the coupling relations. This linking information is written back to atomic simulation objects with the z2-message.

The behavior of atomic models containing continuous variables resembles either Moore automata $\lambda_c : S_c \rightarrow Y_c$ or Mealy automata $\lambda_c : S_c \times X_c \rightarrow Y_c$. Depending on the type of atomic models to which the continuous variables belong, the references to the atomic simulation objects in aSimObj are divided into two groups in a first step. As the next task, inside of the mealy group, references need to be sorted based on interdependencies. The sorted reference vectors with objects of mealy and moore type are passed to the *ODE wrapper* function.

The *ODE wrapper* function is handed over as the closed representation of all continuous model parts to the ODE solver. Via the references to the atomic simulation objects the hybrid atomic model functions λ_c , f and c_{se} can be called by the *ODE wrapper*. For each integration step the *ODE wrapper* calls the continuous output functions λ_c and the rate of change functions f of all hybrid atomic subcomponents referenced in vector `aSimObj` to calculate continuous outputs and derivatives. Additionally, the state event condition function c_{se} of each subcomponent referenced in `aSimObj` is evaluated to check for state events of continuous values that may have occurred. The *ODE wrapper* distinguishes between the calls of λ_c and f or the call of c_{se} by evaluating the flag parameter.

The continuous cycle is iterated as long as either the next discrete internal event becomes imminent or the cycle is interrupted by a state event. If the ODE solver returns a state event, an *se*-message is passed down the tree to find out which atomic subcomponent is affected. The simulator of this atomic then calls the associated state event transition function δ_{state} . Based on the time for next internal discrete event returned by the *se*-message, the *root coordinator* decides whether to start a discrete or a continuous phase next.

A prototype of an advanced PDEVS simulation environment including the wrapper algorithms sketched out has been implemented in the SCE Matlab. It could be proven that, except the `ode15i` for fully implicit differential equations, the entire set of ODE solvers provided by the SCE Matlab, including variable-step solvers as e.g. `ode45` and implicit solvers as e.g. `ode15s`, can be used without modifications for simulating hybrid PDEVS systems with structure variability. A consistent separation of model definitions and numerical solver algorithms is also ensured.

3 Quantized State Systems

Zeigler and Lee defined Quantized Systems (QSS) and their representation in conjunction with DEVS models in the 90ies [11]. The QSS method was improved by Kofman and Junco [5] and offers a completely new innovative approach to numerically solve differential equations.

This approach does not need to be implemented within DEVS necessarily, but it is closely connected to this formalism from the beginning. In contrast to conventional methods for numerically calculating differential equations, time is not discretized within the QSS method, but the states. That point in time is calculated where the continuous state variable has changed more than a certain quantum D .

According to [12] the QSS method is applicable to any category of differential equation systems. Based on selected examples, in [7] Kofman compares solutions obtained by the QSS methods QSS1 and QSS2 with solutions calculated by conventional ODE solvers of the commercial SCE Matlab/Simulink. For the examples studied, he states a considerably better performance of the QSS solutions. The basic approach of the QSS method is introduced below with references to the advanced QSS methods.

3.1 QSS1

For one ordinary differential equation given as $\dot{x}(t) = f(x(t))$ Nutaro [13] specifies the following simulation algorithm, which is independent of the DEVS formalism:

```

t ← 0
x ← x0

while terminating condition not fulfilled
  print t, x
  if f(x) = 0 then
    h ← ∞
  else
    h ←  $\frac{D}{|f(x)|}$ 
  end if
  if h = ∞ then
    stop simulation
  else
    t ← t + h
    x ← x + D * sgn(f(x))
  end if
end while

```


D is the value of the quantum. The step size h is variable and depends on the rate of change $f(x(t))$. Thus, h defines the time interval after which the system's state variable x has changed by D . The state values x result as the sum of the start value x_0 and a multiple k of the quantum D . Thereby, the discrete character of the approximatively calculated state variable becomes apparent.

If not just one differential equation $\dot{x}(t) = f(x(t))$ but a system of differential equations $\dot{\bar{x}}(t) = \bar{f}(\bar{x}(t))$ is computed, the derivatives of all state variables dependent on x_i need to be recalculated at time $t_{x_i} + h_{x_i}$. This modus operandi basically remains the same – also for the advanced QSS algorithms.

3.2 Advanced QSS Algorithms

[6, 7] introduce a wide variety of advanced QSS methods. Second order QSS (QSS2) applies a different quantization (*First Order Quantizer*) to the continuous state variables. The quantized state variable is not constant between two integration steps as it is for QSS1 but follows a linear function. Thereby, the number of event times required for computing a differential equation system is reduced.

Furthermore, there are several implicit QSS methods. The Backward QSS (BQSS) method is designed especially for stiff systems. A combination of BQSS and QSS the Centered QSS (CQSS) is suitable for computing permanently oscillating systems. The Linearly Implicit QSS (LIQSS) are an enhancement of BQSS.

3.3 QSS in DEVS

To permit computation of differential equation systems using QSS in a DEVS environment, the QSS algorithms (QSS integrators) need to be mapped to atomic DEVS models. As far as model design is concerned, this means that one integrator and thus one atomic DEVS are required for each continuous state variable. Detailed specifications for DEVS-based QSS integrators are published in [5, 12, and 13]. Furthermore, each differential equation needs to be mapped to a separate atomic DEVS model.

The consequence of this approach is a rather complex model structure. Figure 3 illustrates the basic model structure of a DEVS model for simulating a differential equation system with two interconnected state vari-

ables using the QSS1 method. The atomic DEVS of a differential equation in the case of QSS1 employment receives the quantized variables q_i as external events from the associated QSS1 integrators. The δ_{ext} and ta functions of the atomic DEVS of the ODE then generate an instantaneous internal event. Thus, its output function λ is immediately executed and computes the actual rate of change value \dot{x}_i and sends it as an output event y_i to the connected atomic DEVS coding the QSS integrator.

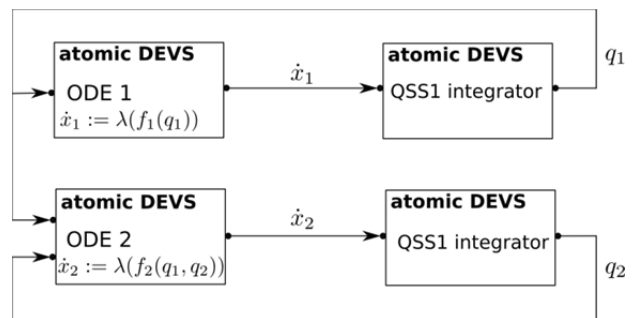


Figure 3. DEVS model structure of a differential equation system with two state variables and associated QSS1 integrators.

Figure 4 shows the basic DEVS model structure of the same ODE system as it needs to be designed, if you aim to use the QSS2 method for computation. According to that not only the atomic DEVS models of the QSS integrators have to be exchanged. The atomic DEVS specifying the differential equations need to be adapted, too. It is essential that the atomic DEVS of the differential equations and the atomic DEVS of the QSS integrators are attuned to one another.

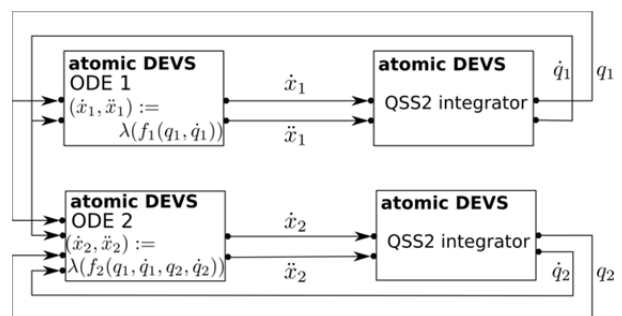


Figure 4. DEVS model structure of a differential equation system with two state variables and associated QSS2 integrators.

Hence, before specifying the atomic DEVS models for the differential equations it needs to be known which QSS method will be used for computation [12]. At this point it becomes clear that there is no strict separation between model description and numerical processing for the QSS method.

The discussed examples are pure continuous systems. But of course, the quantization of the continuous variables and the discrete event-oriented computation allows a seamless specification of hybrid systems too. Event detection and localization functionality has to be implemented in atomic DEVS of the QSS integrators or added as separate atomic DEVS models to the system. The model structure information is kept during simulation runtime.

For software engineering practice, purpose-made templates of e.g. atomic DEVS models for specification of differential equations can be stored in a model library. The modeler simply takes a template file and adapts the concrete algorithm for the derivative in the output function. QSS method specific integrators can be stored in the model library as predefined atomic DEVS models as well. Here, the user can apply those methods without detailed knowledge of the QSS algorithms.

An example of a DEVS-based model library and simulation environment that offers support for QSS based hybrid models is the *PowerDEVS* software suite by E. Kofman [14], which is integrated into the freely available SCE *Scilab* [15].

4 Qualitative Comparison

A key aspect of the comparison subsumed in Table 1 is the quality of integration of the DEVS formalism for hybrid systems into a SCE like Matlab and the resulting potential for engineering application area from user's point of view. Both approaches ensure structure information of the modular hierarchical model during runtime is completely conserved and are therefore both suited for computing structure variable problems in engineering applications.

The QSS approach incorporates the numerical integration algorithms directly into the model description. Model components specifying the differential equations cannot be implemented independently from the integration algorithms the analyst has chosen. In contrast, the wrapper concept is based on a strict separation of model and solution algorithm. Furthermore, the wrapper concept allows the straight, unadapted use of the established ODE solvers provided by the selected SCE.

| | Wrapper Concept | Quantized State Systems |
|---|-------------------------------|--------------------------------|
| Preservation of structure information | yes | yes |
| Computation of continuous solution | at simulator level | at model level |
| Adapted DEVS simulator algorithms | yes | no |
| Usage of established numerical integration algorithms | yes | no |
| Event detection and localization | yes (performed by ODE solver) | yes (bounded to quant. level) |
| Model complexity | lower | higher |
| Reusability of models | yes | yes |
| Reality equivalence of models | very high | high |
| Numerical characteristics | not studied | not studied |
| Runtime / performance | not studied | not studied |

Table 1. Application oriented comparison of two hybrid DEVS approaches.

With the wrapper concept and its prototypical implementation for MATLAB, using varying solvers to execute the same hybrid model is effortless. As all Matlab ODE solvers, except the `ode15i` for fully implicit differential equations, function with the same interface - just one line of code within the *root coordinator* has to be modified.

Due to the fact that the wrapper concept requires separate data structures, additional messages and an interface to the ODE solver, extended DEVS simulator algorithms need to be implemented. In contrast, QSS models can be computed by the standard DEVS simulator algorithms.

Event detection and localization is an essential task in hybrid system simulation. The wrapper concept delegates this task to the ODE solvers of the SCE, while, if following the QSS approach, the atomic models of QSS integrators either need to be extended or additional atomic DEVS models for event detection and localization have to be added to the model. Event detection and localization are not integral capabilities of QSS integration algorithms but can be implemented as extensions to handle events within the bounds of quantization levels.

Continuous model parts designed for computation following the QSS approach consist of at least two atomic models per state variable: One atomic DEVS to specify the differential equation and another atomic DEVS implementing the QSS integration algorithm.

The wrapper concept allows more than one continuous state variable per atomic DEVS model to be defined. Moreover, additional discrete functionality can be added to the same atomic DEVS model. Hence, model structure of models following the wrapper concept appears to be more compact. Model complexity of QSS models compared to models following the wrapper concept increases unavoidably since integration algorithms and event detection and localization methods become part of the model.

However, both methods provide an excellent setting for the modeling and simulating hybrid systems whose modular hierarchical structure may change during simulation runtime. Modeling problems of this type is elaborate and difficult to impossible using standard simulation tools in SCEs such as the toolboxes Simulink and SimEvents for the SCE Matlab.

Reusability of models defined once is given for both approaches. The modular hierarchical characteristic of the DEVS formalism supports this feature *per se*.

In discrete event modeling and simulation for engineering applications as e.g. modeling of manufacturing systems, it is common practice to represent objects of the real world, e.g. work pieces, as entities [16]. One single atomic hybrid DEVS according to the wrapper concept can represent such an entity, including its entire discrete event behavior, and its continuous dynamic behavior. Modeling the same characteristics using the QSS approach requires more than one atomic DEVS.

The wrapper concept has benefits with respect to the degree of reality equivalence of the models over the QSS approach. For the wrapper concept one work piece can be represented by one atomic DEVS model. Thus, one element in the model equates one element in reality. To improve reality equivalence for QSS models, the multiple atomic DEVS components needed to model one work piece could be combined to become a coupled DEVS model. In this case, an 1-to-1-mapping is achieved from the user perspective.

Detailed studies regarding numerical aspects, performance and runtime of both approaches within a SCE have not yet been carried out.

5 Summary

Both approaches offer the potential to make the DEVS formalism for modeling and simulation of hybrid modular hierarchical systems accessible to engineers and support them in modeling and simulating their hybrid engineering applications more effectively. From an academic point of view, the wrapper concept and QSS both lead to comparable, correct results. But today, the DEVS formalism and the QSS method are fairly unknown in the engineering community and are rarely applied to real engineering problems.

Especially for the QSS approach this may be caused by the fact that people like to continue using tools and algorithms with which they have been familiar for years and intuitively refuse to use unknown concepts. An engineer who is familiar with traditional integration algorithms could maybe be convinced more easily to try out the wrapper approach in which he only has to study one new concept, namely the DEVS theory.

The main benefits of the wrapper concept are the integration of established numerical integration methods as they are familiar to engineers, and the consequent separation of the model from the simulation or rather integration algorithms.

The QSS approach is a methodically new approach. It allows the further use of the existing DEVS simulation algorithms for computation of models of hybrid systems. The lack of acceptance for the QSS approach among engineers is probably also related to the lack of available implementations of the integration algorithms for commercial SCEs like Matlab. Presently, QSS implementations such as the PowerDEVS suite [14] are only available for the non-commercial SCE Scilab or implemented as libraries for other simulation environments such as *ModelicaDEVS* [17]. Acceptance of the DEVS formalism itself as well as the acceptance of the QSS method or the wrapper concept is obviously linked to the availability of adequate tools.

References

- [1] B. P. Zeigler. *Theory of Modeling and Simulation (first ed.)*. Wiley Interscience, New York, 1976.
- [2] B. Zeigler, H. Praehofer and T. G. Kim. *Theory of Modeling and Simulation (second ed.)*. Academic Press, New York, 2000.
- [3] F. Barros. *The Dynamic Structure Discrete Event System Specification Formalism*. *Trans. Soc. Comput. Simul. Int.* 13:35–46, 1996
- [4] T. Pawletta, B. Lampe, S. Pawletta and W. Drewelow. *A DEVS-Based Approach for Modeling and Simulation of Hybrid Variable Structure Systems*. In *Modelling, Analysis, and Design of Hybrid Systems*, edited by G. F. S. Engell and E. Schnieder, 107–129. Springer Verlag, 2002.
- [5] E. Kofman, S. Junco. *Quantized-State Systems: a DEVS Approach for Continuous System Simulation*. *Trans. Soc. Comput. Simul. Int.* 18, 123–132, 2001.
- [6] E. Kofman. *Discrete Event Systems and Control of Continuous Systems*. Phd thesis, Universidad Nacional de Rosario, Argentinien, 2003.
- [7] E. Kofman. *Discrete Event Simulation of Hybrid Systems*. *SIAM Journal on Scientific Computing*, 25(5), 1771–1797, 2004. doi:10.1137/S1064827502418379
- [8] O. Hagendorf and T. Pawletta. *A Framework for Simulation Based Structure and Parameter Optimization of Discrete Event Systems*. In: *Discrete-Event Modeling and Simulation: Theory and Applications*, Editors: G.A. Wainer and P.J. Mosterman, CRC Press Inc. of Taylor & Francis Group, USA, 199-222, 2011.
- [9] H. Prähofer. *System Theoretic Foundations for Combined Discrete-Continuous System Simulation*. Phd thesis, VWG, Vienna, Austria, 1992.
- [10] T. Schwatinski and T. Pawletta. *An Advanced Simulation Approach for Parallel DEVS with Ports*. In *Proceedings of Spring Simulation Multiconference 2010, Book 4 - Symposium on Theory of Modeling & Simulation*, 132-139, 2010.
- [11] B. P. Zeigler and J. Lee. *Theory of Quantized Systems: Formal Basis for DEVS/HLA Distributed Simulation Environment*. In *SPIE proceedings*, 49–58, 1998.
- [12] F. E. Cellier and E. Kofman. *Continuous System Simulation*. Springer Verlag, 2006.
- [13] J. Nutaro, J. *Discrete Event Simulation of Continuous Systems*. In *Handbook of Dynamic System Modeling*, edited by P. A. Fishwick, 11–1–11–23. Chapman & Hall/CRC, 2007.
- [14] PowerDEVS by E. Kofman. *PowerDEVS Website*. Accessed Apr. 2, 2012. <http://www.fceia.unr.edu.ar/lspd/powerdevs/>. 2003.
- [15] Scilab by INRIA. *Scilab Website*. Accessed Apr. 2, 2012. <http://www.scilab.org> 2012
- [16] C. Deatcu, T. Pawletta, O. Hagendorf, and B. Lampe. *Considering Workpieces as Integral Parts of a DEVS Model*. In *Proceedings of 21st European Modeling & Simulation Symposium, Volume 1*, 27–35. 2009.
- [17] ModelicaDEVS by T. Beltrame. *ModelicaDEVS at Modelica Association Website*. Accessed Apr.2, 2012. <https://www.modelica.org/libraries/ModelicaDEVS>

Submitted: October 2011

Revised: January 2012

Accepted: March 17, 2012

Physical Modelling for Hallstatt Archaeology

Bernhard Heinzl^{1*}, Erik Auer², Bernhard Slowacki², Kerstin Kowarik³, Hans Reschreiter³,
Nikolas Popper¹, Felix Breitenecker²

¹ dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria; * bernhard.heinzl@dwh.at

² Institute for Analysis and Scientific Computing, Vienna Univ. Technology, Vienna, Austria

³ Natural History Museum Vienna, Burgring 7, 1010 Vienna, Austria

Abstract. Computer simulation can contribute important insights into different areas of archaeology as an addition to traditional experimental methods used to find answers for certain questions arising from archaeological findings, e.g. how tools were used or simply why things were the way they were. In a cooperative project between the Natural History Museum Vienna and the Vienna University of Technology, questions regarding prehistoric physical systems are analyzed. These systems originate from archaeological investigations on the prehistoric salt mines in Hallstatt (Austria) in the Bronze Age.

Three case studies are presented in this article. The first example studies different designs of rope pull systems used to hoist the broken salt from the mining halls through shafts to the surface. In the second task, bronze picks for breaking the salt are investigated using dynamic simulation. The third and last task presents some calculations regarding illumination and air consumption in the mining halls. The simulations allow comparison of different scenarios, help falsifying propositions and gain knowledge in certain aspects regarding technical equipment and working conditions in the prehistoric salt mines in Hallstatt.

Introduction

The prehistoric salt mines of Hallstatt in Austria are subject of great interest for archaeologists, not only for their technological aspects, but also for their complex economic structure. Salt mining activities are dated to 1458-1245 B.C. in the Bronze Age [1].

A large amount of archaeological findings of technical equipment and organic materials (timber, wooden tools, strings of bast, fur etc.) and the perfect conditions of preservation in the mines due to the conserving properties of salt allow for a reconstruction of the working process in the mines [2].

Archaeologists develop a technological reconstruction in form of a mental model (see Figure 1). This mental model suggests that mining was organized in an efficient, nearly industrial manner with highly specialized tools. Salt was mined in underground mining chambers using special bronze picks. The broken salt was collected in buckets and carried to the vertical shaft where it was hoisted to the surface using a wool sack or cloth attached to a linden bast rope.

The high degree of specialization and functionality observable on certain tools and the design for high efficiency suggest that the workforce had highly specialized knowledge in mining technology and infrastructural processes [2].

This mental model raises further questions, which need to be formalized and are typically investigated in experiments using reconstructions of prehistoric tools and utilities. As a new approach, computer simulation can also help to gain knowledge about technological issues and quantitatively compare different possibilities. Results of this work (experiments and simulation) and their interpretation then allow archaeologists to make conclusions about the technological concept and together with new excavation findings update their mental model leading to an iterative process. This process is visualized in Figure 2.

As part of a collaboration between the Natural History Museum Vienna and the Vienna University of Technology, several questions from this mental model regarding technical equipment and utilities in the Hallstatt salt mines are formalized and investigated using software tools for modelling and simulation of physical systems. Different scenarios are considered and comparisons are quantified in order to be able to evaluate the different possibilities regarding their plausibility and eligibility.



Figure 1. Impression of a Bronze Age mining hall as reconstructed for the Christian v. Tusch-Werk in Hallstatt (© D. Gröbner, H. Reschreiter, NHM Vienna). The workers used special bronze picks to break the salt. The salt pieces were then collected in buckets and carried to a vertical shaft where it was hoisted to the surface using a rope pull system. Burning woodchips served as illumination in the underground mining halls.

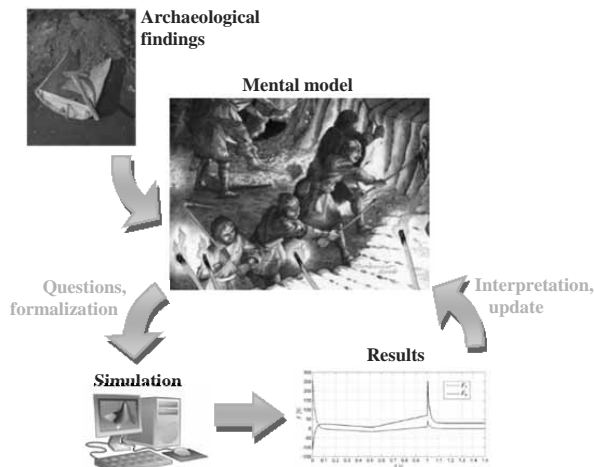


Figure 2. Iterative process for adapting the mental model. New archaeological findings provide information. Questions arising from the mental model are formalized and mapped into a computer model. Simulation then generates results for different scenarios, which are interpreted and allow conclusions for the mental model (© A. Rausch, D. Gröbner, H. Reschreiter, NHM Vienna).

1 Rope Pull Systems

Rope pull systems were used to hoist the broken salt from the mining halls through shafts to the surface [2] (see Figure 3). While there are archaeological findings of bast ropes and other appliances, there are still some matters regarding the construction, length and arrangement of the rope pull systems at issue.

To estimate and compare the time and strength requirements for transporting the salt, various options are analyzed using simulation models. For example, we compare two variants of the rope design, an open and a closed version, depicted in Figure 4.

An important issue also concerns modelling of the rope guide, for which we consider two possibilities, one with sliding friction on a log and one with return pulley.

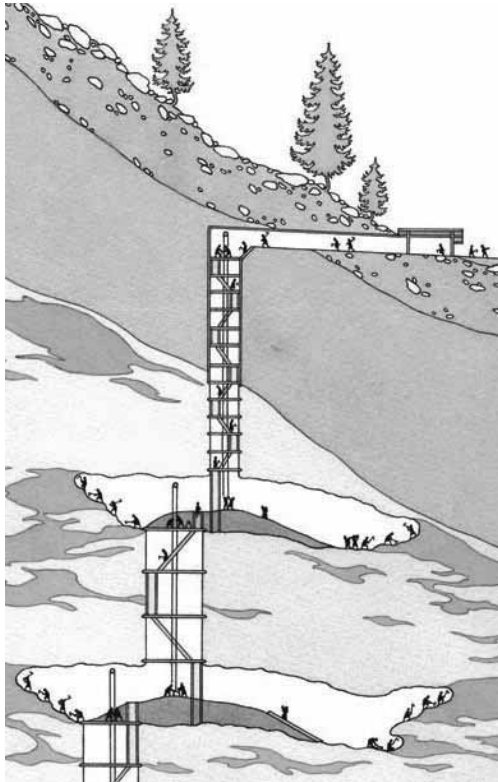


Figure 3. Schematic reconstruction of the mining halls and shaft structure with rope pull systems (© D. Gröbner, H. Reschreiter, NHM Vienna).

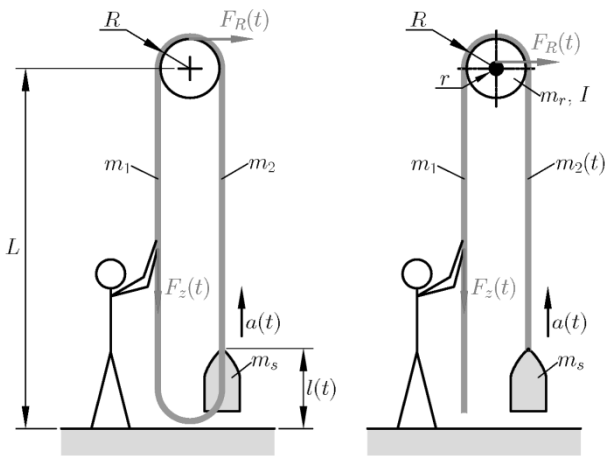


Figure 4. Different design options for the rope pull system with closed rope (left) or open rope (right). The schematic also shows two possibilities for the rope guide: sliding friction on a log (left) and rotating return pulley (right).

1.1 Rope Guide on a Log

Considering simple balance of forces according to Figure 4 taking into account mass of the rope on both sides (m_1 and m_2 , resp.) and forces of inertia leads to

$$F_z(t) + m_1(g - a(t)) - F_R(t) = m_2(t)(g + a(t)) + m_s(g + a(t)) \quad (1)$$

with the external force $F_z(t)$, predefined acceleration $a(t)$ (see equation (4)), acceleration of gravity g and mass of the salt bags m_s . The coulomb friction force $F_R(t)$ is given as

$$F_R(t) = \mu(v(t))[F_z(t) + m_1(g - a(t)) + m_2(t)(g + a(t)) + m_s(g + a(t))] \quad (2)$$

The expression in square brackets denotes the total normal force on the log. For the friction coefficient μ , we define the following expression depending on the rope velocity v , also visualized in Figure 5.

$$\mu(v) = (\mu_c + (\mu_{brk} - \mu_c) \exp(-c_v|v|) \operatorname{sgn}(v) + fv) \quad (3)$$

with a coulomb friction coefficient μ_c , breakaway friction coefficient μ_{brk} , viscous friction coefficient f and coefficient c_v . This friction model takes into account sticking friction at velocities near zero (μ_{brk}) as well as viscous friction at high velocities (fv).

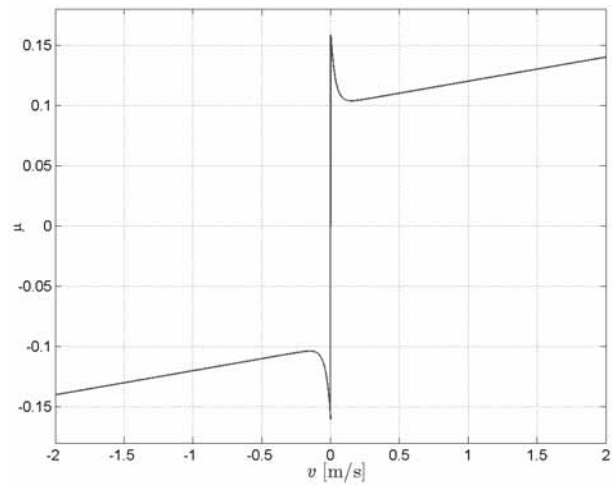


Figure 5. Model of the friction coefficient μ with sticking and viscous components for parameters $\mu_c = 0.1$, $\mu_{brk} = 0.16$, $c_v = 30$ and $f = 1/50$.

In order to determine the force necessary for hoisting the salt to the surface, the proposed model receives the desired velocity over time as input signal and calculates the necessary external force $F_z(t)$. One possible input signal is depicted in Figure 6. It shows a smooth nearly periodic signal (due to alternating pull of a single person) with global upward trend.

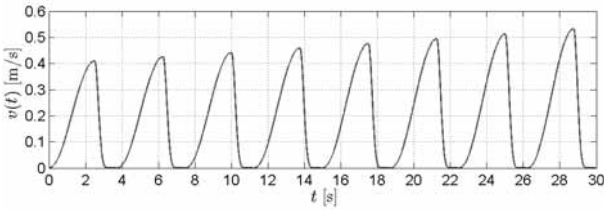


Figure 6. Possible input signal $v(t)$ with smooth progression and global upward trend.

The external force can then be calculated using equations (1), (2) and (3) and the time derivative

$$a(t) = \frac{d}{dt} v(t) \quad (4)$$

of the input velocity. For an open rope, the mass $m_2(t)$ of the rope on the right side of the log depends on the current position $l(t)$,

$$m_2(t) = m_{2,0} - \rho l(t), \quad (5)$$

with the assumed density $\rho = 1.5 \text{ kg/m}$ and starting mass $m_{2,0} = \rho L$, and therefore decreases over time according to

$$\frac{d}{dt} m_2(t) = -\rho \frac{d}{dt} l(t) = -\rho v(t) \quad (6)$$

with given input velocity. If the rope is closed (see Figure 4 left), then the mass m_2 is constant

$$m_2 = \rho L. \quad (7)$$

In both cases (open and closed rope), the mass m_1 on the left side is assumed constant, also with

$$m_1 = \rho L. \quad (8)$$

The mass of one bag of salt is assumed to be about 28 kg which corresponds to a volume of 20 l. For the value of m_s , one or more bags of salt are considered,

$$m_s = k \cdot 28 \text{ kg}, \quad k = 1, 2, 3, \dots \quad (9)$$

with the number of bags k .

Evaluation of a simulation using the presented equations implemented in MATLAB and parameters $L = 20 \text{ m}$, $k = 1$ for example for a greased log ($\mu_c = 0.1$, $\mu_{brk} = 0.16$) provide the results given in Figure 7. The results show a total duration of about 81 s to hoist one bag of salt over a height of 20 m, maximum external force is about 500 N. The height $l(t)$ pulsates due to the periodic velocity (see Figure 6).

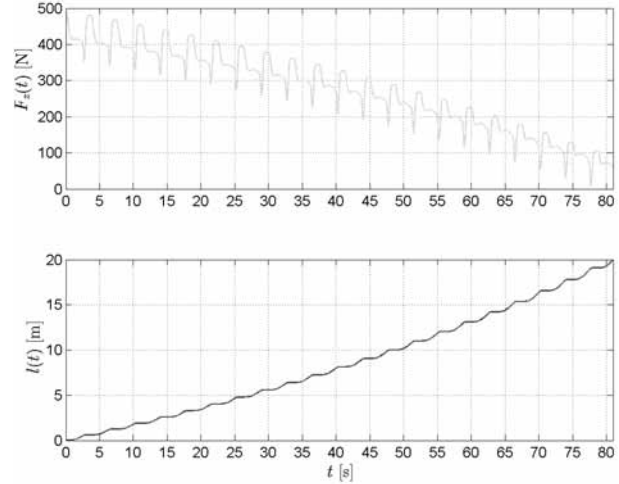


Figure 7. Simulation results for a greased log with open rope and $L = 20 \text{ m}$, $k = 1$.

Similar investigations for different shaft heights L as well as different number of bags lead to the results presented in Figure 8. It shows that each additional bag of salt adds about 400 N to the external force, which is more than the own weight because of additional friction.

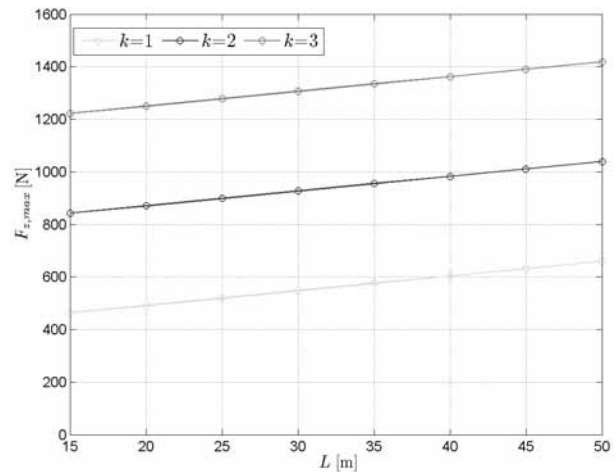


Figure 8. Maximum external force F_z for the scenario with log for different shaft heights L and different number of bags k .

1.2 Rope Guide with Rotating Return Pulley

Using a return pulley with outer Radius R and axle radius r instead of a static log adds terms for pulley mass m_r and inertia I to equations (1) and (2), thus, with regard to Figure 4, resulting in the equations

$$F_z(t) + m_1(g - a(t)) - F_R(t) \frac{r}{R} - \frac{Ia(t)}{R^2} = m_2(t)(g + a(t)) + m_s(g + a(t)), \quad (10)$$

$$F_R(t) = \mu(v(t))[F_z(t) + m_1(g - a(t)) + m_2(t)(g + a(t)) + m_s(g + a(t)) + m_r]. \quad (11)$$

Different values can be considered for the parameters of the pulley depending on its material (wood, bronze, etc.). For a pulley made of wood we get for example the following parameters:

$$\begin{aligned} R &= 0.6 \text{ m}, \quad r = 0.2 \text{ m}, \quad m_r = 16.3 \text{ kg}, \\ I &= 3.3 \text{ kgm}^2, \quad \mu_c = 0.06, \quad \mu_{brk} = 0.11. \end{aligned} \quad (12)$$

As a simulation result, maximum required external force F_z for the scenario with return pulley and different values for shaft height L and number of bags k is presented in Figure 9. Comparison with Figure 8 shows significant reduced force requirement due to lower friction.

Figure 10 compares force input between the version with open rope and with closed rope (i.e. where both ends are connected). Since the moving mass is constant for the variant with closed rope, the required force is also constant in average, whereas in the other case the average decreases steadily over time. A closed rope enables more uniform distribution of the rope mass and therefore requires less external force.

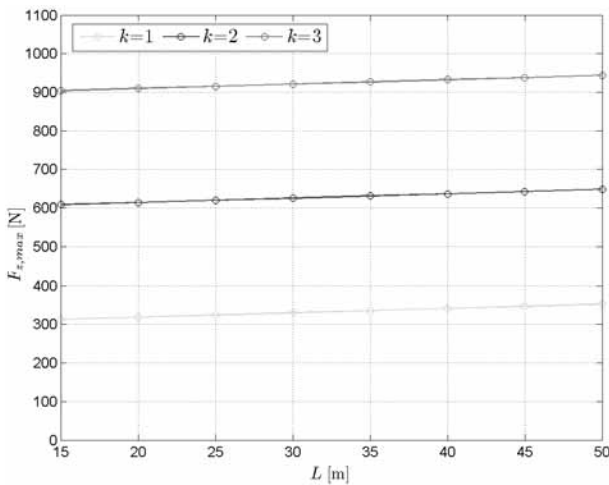


Figure 9. Maximum external force F_z for the scenario with return pulley for different shaft height L and different number of bags k .

In summary the simulation results show significant force requirement for the model with sliding friction, especially because of the high mass of the rope. This is also the reason for limitations regarding the maximum allowed shaft height.

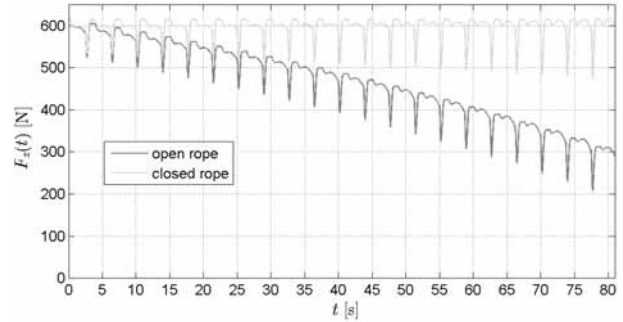


Figure 10. External force $F_z(t)$ over time for the scenario with return pulley and both variants of rope design.

With the presented equations it is also possible to provide the external force $F_z(t)$ as an input signal and calculate resulting velocity $v(t)$ and position $l(t)$ over time. This is still ongoing work and will be investigated in further studies.

2 Bronze Picks

Archaeologists know from excavation findings that the salt was mined using bronze picks with wooden handle [2] (see Figure 11 left). Highly interesting is the unusual shape of these picks with a typical angle between the shaft and tip of about 55 to 75 degrees. It is believed that this particular shape was adapted to the specific working conditions in the Hallstatt mines, especially since no similar devices have been found at other archaeological sites. The small angle does not allow typical circular hacking motion, which is why it is not yet completely clear exactly how such a pick was used.

Modelling the pick as a rigid body system (Figure 11 right) allows dynamic simulation and evaluating possible movement scenarios and comparisons regarding acceleration and reaction forces.

Different usage of the pick can be considered, e.g. breaking the salt on the wall or on the floor, different grip points, using one or two hands, etc. Figure 12 shows a visualization of a wall impact as an example scenario presented in this article.

An object-oriented modelling approach for mechanical rigid multibody systems (in our case using MATLAB/SimMechanics [3]) allows quick and intuitive structured model development using predefined components. Dynamic and constraint equations are generated, sorted and processed for numeric simulation automatically by the software, allowing easy adaption to different scenarios, e.g. forward or inverse dynamics.



Figure 11. Left: Reconstructed Bronze Age pick (© A. Rausch, NHM Vienna). Right: Rigid body model of the pick in MATLAB/SimMechanics.

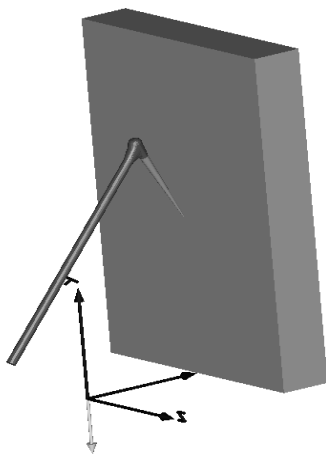


Figure 12. Visualization of an example scenario, where the pick breaks salt off a wall located at $x = 0.5$ m.

Figure 13 depicts the rigid body model for the pick developed in MATLAB/SimMechanics. Two phases are considered:

- During free motion (block 'Acceleration'), the pick accelerates along a fixed trajectory.
- When collision with the wall is detected, the actuation switches to the block 'Collision', and the body decelerates until it comes to a stop.

The pick has three degrees of freedom - two translational (x and y) and one rotational (phi , along the z -axis). Sensor elements allow measuring acceleration as well as reaction forces.

Movement trajectories are obtained by geometrical considerations like shown in Figure 14. Several points are defined along a trajectory of the tool tip depending on the range of motion for a human. Taking into account the time relation, this trajectory also determines velocity and acceleration behaviour. Coordinates for one example trajectory are presented in Figure 15.

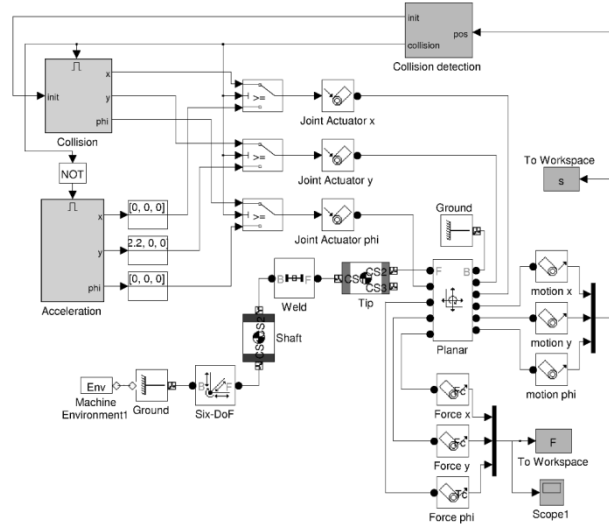


Figure 13. Rigid body model for the bronze pick in MATLAB/SimMechanics. Motion actuation is switched between two states: During free motion (block 'Acceleration'), movement is restricted along a predefined trajectory. After the impact, the tip decelerates until standstill (block 'Collision').

Special focus has to be put on the angle in which the tool tip hits the wall. In order to use the tool effectively (this results from the shape of the tip), the impact angle is required to be

$$\alpha = 20^\circ \dots 30^\circ. \tag{13}$$

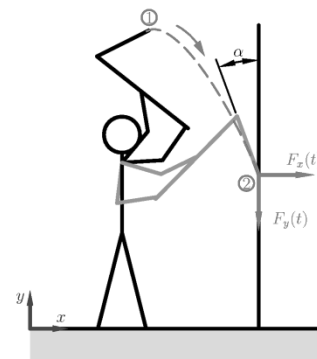


Figure 14. Trajectory for movement of the tool tip and reaction forces F_x and F_y .

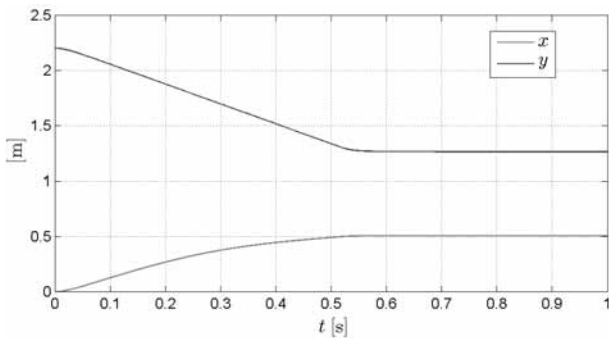


Figure 15. x - and y -coordinates over time for an example trajectory.

Using this geometry data as well as mass and inertia calculations for shaft and tip, the rigid body model can be set up. Simulation provided by inverse dynamics features of the simulator then calculates for example reaction forces (F_x, F_y) on the tool tip, see Figure 16.

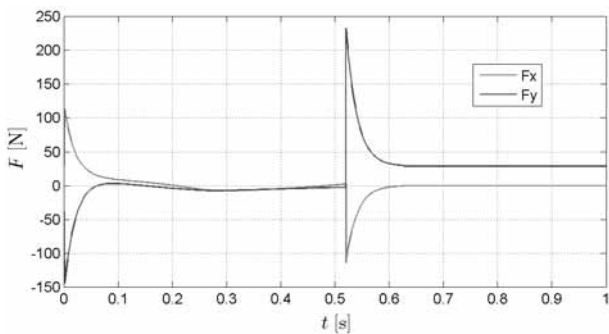


Figure 16. Reaction forces in x - and y -direction on the tool tip as a simulation result for the presented scenario. At $t = 0.52$ s, the impact on the wall occurs leading to a peak in the vertical force.

Other scenarios like mining the salt on horizontal or inclined surfaces are also currently investigated. However, it already shows that mining on the floor seems more exhausting, requires more energy and is therefore less efficient than mining on the wall.

3 Woodchip Flame, Lighting and Air Consumption

During mining, burning sticks of wood served as the only illumination in the mining halls (see also Figure 1). Burnt down woodchips were found during excavation in large quantities [2].

The resulting light intensity depending on the number of burning woodchips can be estimated using a uniform arrangement shown in Figure 17. The flame of a burning woodchip is comparable to a flame of a burning candle, therefore we assume a light intensity for each flame of

$$I_v = 1 \text{ cd.} \quad (14)$$

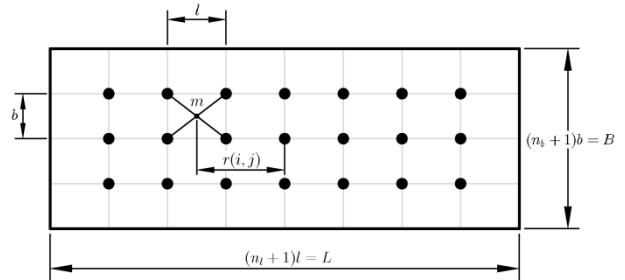


Figure 17. Uniform arrangement of burning woodchips (n_l in length, n_b in width) in a rectangular mining hall.

Local minimum illumination occurs at points of maximum distance to the light sources (for example point m in Figure 17). For such a point, the distance $r(i, j)$ to each woodchip can be calculated:

$$r(i, j) = \sqrt{\left(\left(i - 1\right)l + \frac{l}{2}\right)^2 + \left(\left(j - 1\right)b + \frac{b}{2}\right)^2} \quad (15)$$

and the sum over all positions leads to the total illumination

$$E_{v,m} = \sum_{i=1}^{n_l} \sum_{j=1}^{n_b} \frac{I_v}{r(i, j)^2}. \quad (16)$$

For different densities of woodchips in length and width,

$$\rho_l = \frac{n_l}{L}, \quad \rho_b = \frac{n_b}{B}, \quad (17)$$

calculation results are shown in Figure 18.

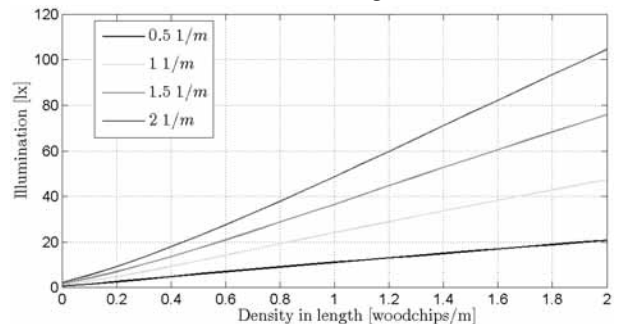


Figure 18. Calculation results for illumination for different densities of woodchips in length and width.

Furthermore, some static calculations help estimating the oxygen consumption of the flames, which, in addition to the oxygen demand of the workers, give information about the necessary air ventilation. For the air consumption of a person doing heavy labour, we assume a typical value of

$$A_p = 0.81 \frac{\text{m}^3}{\text{h}} \quad (18)$$

at 21% percentage of oxygen in the air. A typical woodchip made of fir wood consumes about

$$A_w = 0.014 \text{ m}^3 \quad (19)$$

of oxygen during burning. Altogether, this data leads to the following formula for the total air consumption:

$$A_{\text{total}} = kA_p + n \frac{A_w}{21\% \cdot T} \quad (20)$$

with the number of workers k , total number of woodchips n ($n = n_l \cdot n_b$) and burning time T of a woodchip. Figure 19 depicts calculation results for different parameter values.

This air consumption represents the minimum necessary air exchange in the mining halls for a certain illumination in order to keep the oxygen level constant.

Future work will focus on fluid simulations for the air distribution in the underground mining halls and shaft system in order to determine possible air change rates and - combined with the results above - quantify limitations regarding maximum number of workers in the mining halls.

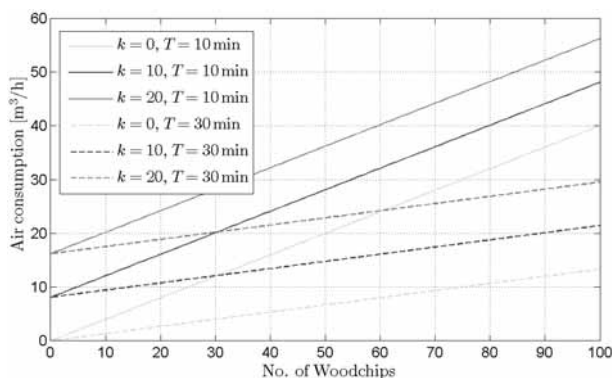


Figure 19. Calculation results for air consumption for different number of woodchips, burning time T and workers k .

4 Conclusion

The presented simulations provide quantitative results for different scenarios and experiments in order to assess their plausibility. Even though the results may not produce many definitive answers for archaeologists, it still allows falsifying certain propositions and thus restricts the total set of possibilities.

Therefore, modelling and simulation concludes to be a useful tool in experimental archaeology which can be nicely combined with physical experiments (a method which is quite common in archaeology) in order to obtain data input and parameter values for the simulation.

The calculations and simulation results are visualized graphically. Their evaluation and interpretation helps archaeologists gain knowledge about transport mechanisms and working conditions in the prehistoric salt mines in Hallstatt and in conclusion adapt and refine the mental models in an iterative process.

New object-oriented modelling approaches for physical systems enable the user to develop modular models in a structural manner, which can easily be adapted to different scenarios.

References

- [1] M. Grabner, H. Reschreiter, F.E. Barth, A. Klein, D. Geihofer, R. Wimmer. *Die Dendrochronologie in Hallstatt*. Archäologie Österreichs, 17(1): 49-58, 2006.
- [2] H. Reschreiter, K. Kowarik. The Bronze Age. In: Kern et al. *Kingdom of Salt: 7000 Years of Hallstatt*. Vienna: VPA3, Natural History Museum Vienna, 48-64, 2009.
- [3] MathWorks, Inc. *SimMechanics User's Guide*. 2011. Available online: http://www.mathworks.de/help/pdf_doc/physmod/mech/mech Ug.pdf

Submitted: January 2012

Revised: February 10, 2012

Accepted: March 20, 2012

Object-Oriented Modelling for Energy-efficient Production

Bernhard Heinzl^{1*}, Matthias Rößler^{1**}, Nikolas Popper², Michael Landsiedl²,
Alexandros Athanassios Dimitriou³, Felix Breitenecker¹

¹Institute for Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria, *bernhard.heinzl@tuwien.ac.at, **matthias.roessler@tuwien.ac.at

²dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria

³Institute for Production Engineering and Laser Technology, Vienna University of Technology, Vienna, Austria

Abstract. Energy optimization in production plants is at present a very current topic. In this context, the research project INFO deals with a comprehensive simulation of production halls with all micro- and macrostructures, in order to be able to make qualified predictions about the efficiency of different energy saving measures. An important sub-range of this research concerns modelling and simulation of machine tools, two parts of which will be presented in this article.

The first part focuses on general multi-domain modelling of an actual machine tool. An object-oriented modelling approach allows combining electrical, mechanical as well as thermal aspects in a structural manner. Combined top-down and bottom-up modelling techniques with increasing level of detail identify the degree of modelling effort necessary or sufficient for certain applications and simulating certain scenarios. Future work plans to validate the simulation results against measurement data obtained from machine tests with the lathe.

The second part concentrates on the thermal coupling of the machine tool components with the environment. For that we selected a specific area of a machine tool, i.e. a linear guiding device with drive motor. In a first step the thermal behavior is studied and in a further step we consider the heat transfer and other effects on the environment. For this, the room around these tools is discretised in thermal compartments, which can also affect each other. The object-oriented modelling approach allows refining the discretisation easily, so that effects of a higher resolution can be studied too.

Introduction

The research project INFO (supported by the Austrian Research Promotion Agency (FFG)) [1] pursues the primary goal to increase energy efficiency in production

plants by considering various disciplines of energy technology, production technology and building design in a comprehensive approach [2]. Qualified and customized predictions and recommendations about the efficiency of different energy saving measures can be made using comprehensive simulation models of the production plant including all relevant micro- and macro-structures and therefore identify potential savings in manufacturing plants.

Due to the size of the project and the various aspects that are considered, the project is highly interdisciplinary. This makes it necessary to gather experts from many different fields, e.g. engineering, thermodynamics or building simulation, who build the models specific for their field of research using specific approaches and specialized simulation tools. The different partial models are then combined to an overall simulation.

One interesting topic in the context of the project is to look at machine tools from an energetic point of view including power consumption and heat emission [3]. There are two main reasons this topic is interesting with regard to a comprehensive simulation:

First, the size of the resulting model makes it necessary to study the level of detail in which the individual parts can be modelled and if it makes sense to have partial models in a very high resolution. Therefore the first part of this study deals with multi-domain simulation model of a machine tool that is modelled in three levels of detail and analysed regarding simulation time and numerical stability.

Second, the behaviour and the time scales of the models are very different, so it is essential to study the possibilities on how to couple the different model parts. To analyse the problems that occur because of these aspects a part of a machine tool, a linear guiding device, and a room model are considered for the second part of

this study. In order to consider only effects that occur due to the coupling of the two model parts, the partial models themselves are kept on a rather low level of detail.

The new object-oriented modelling approach applied in both parts of this study (called Physical Modelling) is designed for modelling and simulation of physical systems in various domains and has proven its worth over the last years. Not only does this approach make the modelling process simpler, faster and more intuitive, the models are also more modular allowing the user to easily refine the structure by adding new components in a structural manner.

Since there are a number of software tools available for Physical Modelling in multi-domain applications, choosing two different tools for our implementations (i.e. MATLAB / SimscapeTM [4] and MapleSim [7] in our case) allow comparisons regarding numerical performance, suitability and usage.

1 Multi-Domain Modelling of a Turning Lathe

In our study of modeling and simulation of machine tools, we consider not only electric drive components and mechanical elements, but we are also interested in the waste heat developing in different places. In order to be able to validate the model against actual measurement data, we attempt to recreate an actual turning lathe, which is provided by the Institute for Production Engineering and Laser Technology from the Vienna University of Technology. Although the considered machine tool is rather simple compared to others, it provides sufficient possibilities for our investigations. There are three main drivelines:

- Main drive: Main motor, gear belt drive, spindle with chuck and workpiece
- Longitudinal feed (z-axis): Servomotor, leadscrew drive and slide holding the cross feed
- Cross feed (x-axis): Servomotor, leadscrew drive, cross-slide with tool holder and cutting tool

The main drive sets the workpiece into rotation, longitudinal and cross feed drives allow positioning the tool in z- and x-direction (axial and radial to the workpiece), see Figure 1 During machining, the cutting tool penetrates the workpiece and removes material in form of a chip during relative motion.

The cutting energy is mostly converted into thermal energy. All three drivelines receive their electric power from an inverter, which is simplified as ideal voltage source.

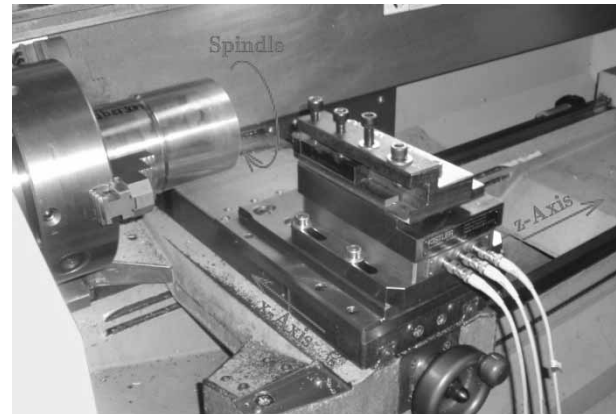


Figure 1. The three axes of the lathe: Spindle for driving the workpiece, longitudinal and cross feed for positioning the tool in the z- and x-direction.

The model is implemented step by step with gradually increasing level of detail allowing to identify numerical boundaries of our simulation. On the other hand we can also show what degree of modelling effort is necessary or sufficient for certain applications, where significant changes in the simulation results are to be seen and which simulation scenarios are possible and reasonable. The next section presents the first of the developed models and afterwards some simulation results are shown.

1.1 Modelling

The first overall model is relatively simple, but already contains the main mechanical and electrical components of the main drive as well as the slides for automatic feed and infeed. The system is implemented in MATLAB / Simscape. Figure 2 shows the model part for the main drive with asynchronous motor, electrical control, voltage supply, gear belt drive, friction components and mechanical loads. The basic structure of the drive is easy to see which is helpful for further model adjustments, therefore pointing out one of the advantages of this object-oriented modelling approach.

The asynchronous motor as well as the servo motors for the remaining drives of the lathe and some of the basic mechanical components like gear belt drive, lead screw and linear bearings are modelled as Simscape components using the Simscape Language [5] with parameters extracted from available data sheets.

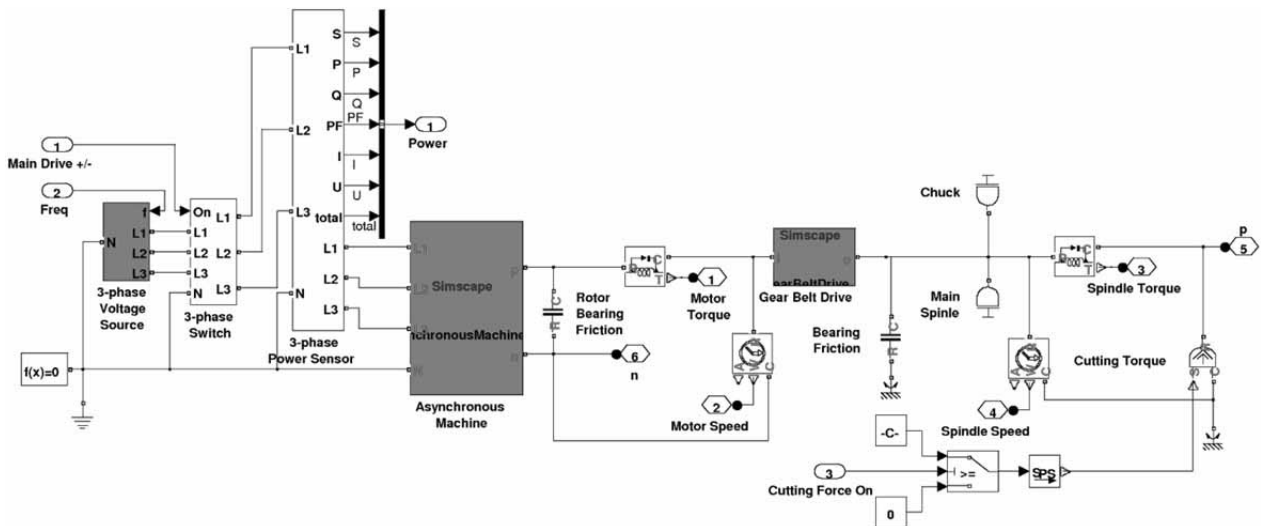


Figure 2. Main drive of the turning lathe model with power supply, electric control, asynchronous motor, gear belt drive, load and cutting force (over both columns).

A code fragment of this implementation for the asynchronous machine is shown in Listing 1. It implements common formulas for this electric drive in normalized space vector description.

```

component AsynchronousMachine
(...)
parameters (Access = public, Hidden = true)
    M=2/3*[1, -1/2, -1/2; 0, sqrt(3)/2, sqrt(3)/2];
end
equations
(...)
    us' == M*[u1; u2; u3]; is'==M*[i1; i2; i3];
    i1+i2+i3==0;
    %Standardized equations for squirrel cage ASM
    Us == is*rs + psis.der/0megaref el;
    ur == ir*rr + psir.der/0megaref el...
        -[-psir(2), psir(1)]*omegam;
    psis == ls*is + ls*(1-sigma)*ir;
    psir == ls*(1-sigma)*(is+ir);
    ur == [0, 0];
    %Torque equation
    mr == is(2)*psir(1)-is(1)*psir(2);
end
    
```

Listing 1. Code fragment of asynchronous machine model in Simscape Language.

Existing Simscape blocks from the Simscape foundation library [6] complete the model with components for inertia, friction and sensor blocks for measuring state variables.

During the machining process, the cutting force generates an additional torque on the motor, which is calculated externally using common formulas and parameters [8].

Further parts of the overall model include similar drivetrains for both slides for automatic feed and infeed. In order to keep the first model simple, motor control, power electronics and thermal aspects are not yet taken into account. Instead, an ideal 3-phase voltage source directly provides appropriate voltage signals with variable frequency and amplitude controlled by external signals. This however limits possible simulation scenarios, for example only cases with constant motor speed can be considered.

1.2 Simulation Results

As output of this model, the results of a basic simulation run with the lathe are presented in Figure 3. It shows spindle speed, feed speed and cross-slide position during a single longitudinal machining process, also sketched in Figure 4. The process starts with the run-up of the main spindle, slide and cross-slide are initialized with appropriate speed. When the cutting tool hits the workpiece (at about 3.8 s), cutting an feed forces set in causing disturbances in speed and displacement. Since there is no feedback motor control, the remaining deviation has to be accepted for this simple model. After the cutting tool has left the workpiece at time 13 s and a run-out length of 20 mm, main spindle and slide decelerate between 16.5 s and 17 s until they come to a stop.

For energy investigations, Figure 5 depicts the total power consumption for the described machining process. It stands out that the apparent power is exceptionally high during the acceleration phase of the main drive resulting from the simplified motor model and control.

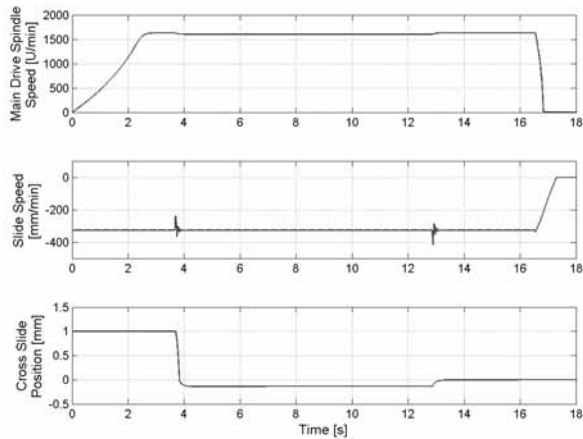


Figure 3. Simulation results of a basic simulation run: Spindle speed (top), slide speed (middle) and cross-slide position (bottom).

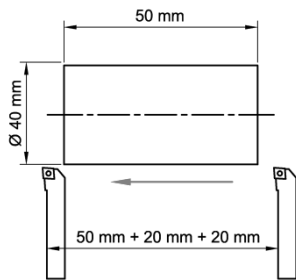


Figure 4. Schematic of the simulation run shown in Figure 3.

2 Thermal Coupling with the Environment

For the second part the thermal coupling of a machine tool with its environment shall be analyzed. In a first step two partial models are built. The first one is a model of a simple part of a machine tool, namely a linear guiding device, depicted in Figure 6, which is also provided by the Institute for Production Engineering and Laser Technology like the turning lathe from the first part of this study.

The second one is a model of the environment surrounding the machine, which is realized as a compartment model, a sketch of the discretization of the room can be seen in Figure 7.

It also depicts the machine that serves as a heat source for the room and is located in two different compartments, allowing it to distinguish between different heat sources within the machine, e.g. the electrical and the mechanical parts.

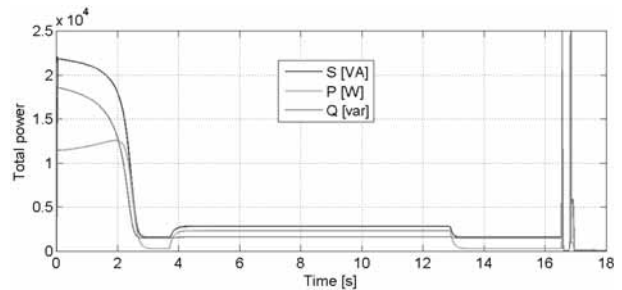


Figure 5. Total power consumption for all drives during the presented simulation run.

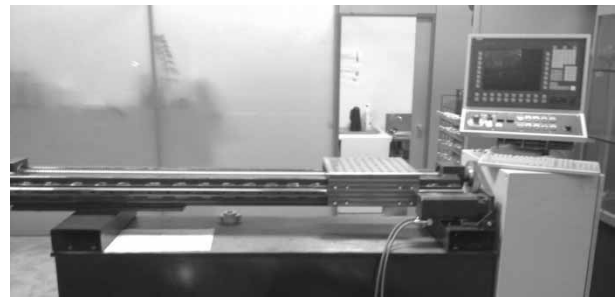


Figure 6. Test setup for the linear guiding device which is to be modelled.

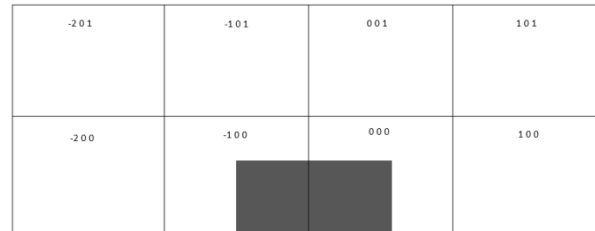


Figure 7. Schematic view of the discretisation and the two heat sources representing different parts of the machine.

2.1 Modelling

The linear guiding device as depicted in Figure 6 consists of a permanent magnet DC motor, a gear belt, a thread bar and the cart, where the mass is attached. In the model The motor is represented by a DC Permanent Magnet Motor block. For the gear belt, the gear ratio is modelled by a gear, the elasticity by a linear spring and the friction by a translational friction component.

The thread bar is modelled through its inertia and lead. Furthermore, the friction between the cart, the thread bar and the sliding mass are being taken into account. All components are from the Modelica Standard Library [9] and the model is implemented using MapleSim [7]. Figure 8 shows the model of the machine tool.

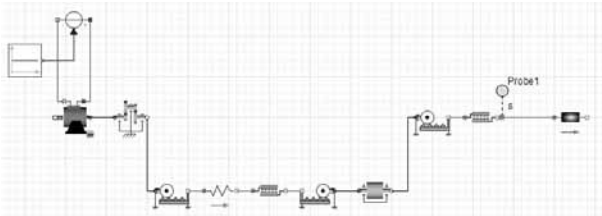


Figure 8. MapleSim model of the linear guiding device with blocks from the Modelica Standard Library.

For this first model, the source for the motor has a constant voltage, but it is planned that in further implementations measurement data will be incorporated. The parameters for the parts of the linear guiding device are taken from data sheets. For the model of the environment a room with the proportions 20m x 10m x 6m is assumed. In a first step, this room is discretised with eight compartments of equal volume, at first only two-dimensional for reasons of simplicity as can be seen in Figure 7. The modularity of the model structure suggests the usage of a single component that describes the whole compartment, which is depicted in Figure 9.

For the model several assumptions are made to keep it simple. First of all, the walls of the room are perfectly isolated, that means there is no heat exchange between the wall and the air. The only mechanism of heat exchange between the compartments is conductance, so there is no convection and the transfer is only permitted to adjacent compartments. Furthermore, the machine is the only source of heat in the room. For the parameters of the air, namely the density, specific heat capacity and thermal conductivity, we used typical values [10]. All parameters are assumed to be constant with a reference temperature of 20 °C.

The overall room model is depicted in Figure 10. It shows the modular structure of the model and the similarities to the schematics in Figure 7. The compartments are all equal in size, which is 5m x 5m x 6m and they are only horizontally and vertically connected, which respects the model assumption of the direction of the heat flow. On the bottom of the model there are two heat sources representing two different discretised part of the machine, which emit heat (motor, friction).

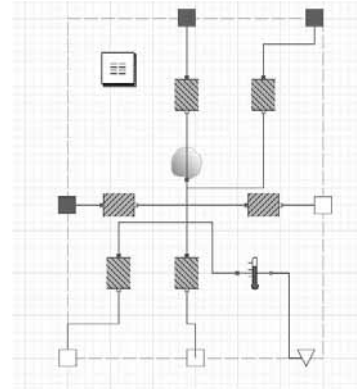


Figure 9. Single Compartment of room model in MapleSim.

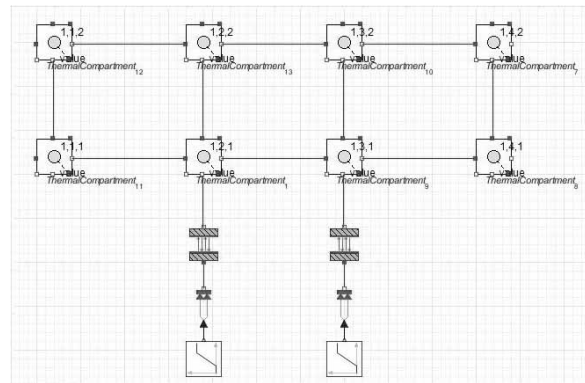


Figure 10. Thermal model of the discretised environment with eight compartments, two heat sources and conductive heat transfer.

2.2 Simulation Results

As mentioned before a constant voltage source was used for the first simulation runs of the linear guiding device. As a result the linear translation of the sliding mass is shown in Figure 11.

The results are somewhat realistic in the way that you can see the run-up of the motor and the linear characteristics of the model in steady state.

The simulation of the room model starts with a running machine (i.e. it emits heat) and an initial temperature of 20 °C in all compartments. The machine runs for an hour, where the two different sources constantly emit a certain amount of heat. After that, the machine is turned off and is cooling down during the next four hours, cf. Figure 12. Figure 13 shows the development of the temperature in each compartment.

One can observe that the biggest increase of temperature is in the compartment with the stronger heat source. The compartments adjacent to it have an observable increase of temperature after a short time and the compartments further away are heating up much slower.

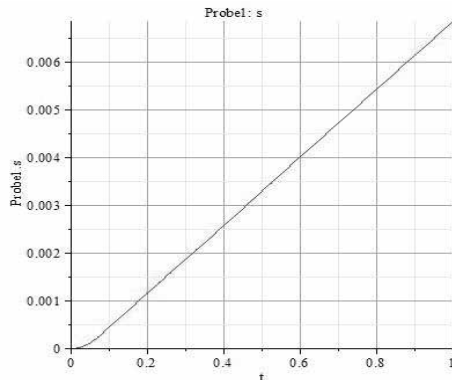


Figure 11. Acceleration and linear translation of the sliding mass over time.

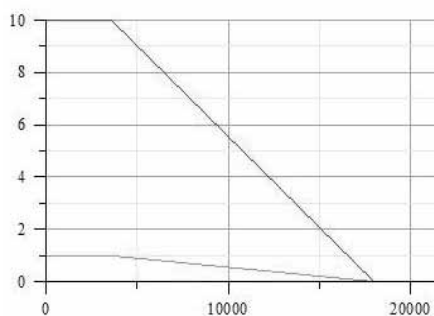


Figure 12. Heat emission of the two different sources over time.

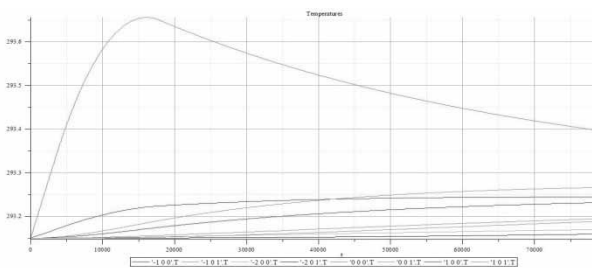


Figure 13. Temperatures in the different compartments over time.

There is also a little difference between the trajectory of the compartment with the lesser heat source in it and the other compartments without a heat source, but the amount of emitted heat is too low to make a significant impact. Furthermore it can be seen, that the temperatures in the compartments are approaching a steady state, where the temperature is equal for all compartments.

3 Summary and Conclusion

The object-oriented modelling approach is confirmed to be a suitable tool for multi-domain modelling and thermal considerations of machine tools resulting in modular models, which can easily be modified and refined.

However, simulations show that the models are comparatively complex with a larger amount of equations compared to classical equation-oriented modelling which leads to less efficient numerical simulations resulting in higher computation times. Especially multi-domain applications usually lead to stiff equation systems that require additional effort during simulation.

4 Outlook

Future work regarding the studies shown in this paper will focus on refinement of the models, e.g. for detailed investigation of thermal aspects of the machine tool, and on the coupling of the models that have been presented. For model validation it is planned to obtain measurement data from the turning lathe as well as the linear guiding device and compare these measurements with corresponding results from the simulation.

References

- [1] <http://www.projekt-info.org/>
- [2] Dorn, Ch., Bleicher, F.: *Interdisziplinäre Forschung zur Energieoptimierung in Fertigungsbetrieben*. Junior Science Conference 2010
- [3] Stark, M., Dorn, Ch., Bleicher, F.: *Energetic Comparison of Machine Tools*. DAS 2010, 27th Symposium on Advances in experimental Mechanics, Breslau, Poland
- [4] <http://www.mathworks.com/products/simscape>
- [5] MathWorks, Inc.: *Simscape Language Guide*. 2011. Available online: http://www.mathworks.cn/help/pdf_doc/physmod/simscape/simscape_lang.pdf
- [6] MathWorks, Inc.: *Simscape User's Guide*. 2011. Available online: http://www.mathworks.cn/help/pdf_doc/physmod/simscape/simscape_ug.pdf
- [7] <http://www.maplesoft.com/products/maplesim>
- [8] Degner, W., Lutze, H., Smejkal, E.: *Spanende Formung, Theorie – Berechnung – Richtwerte*. Hanser Verlag, 2009
- [9] <https://www.modelica.org/>
- [10] http://www.engineeringtoolbox.com/air-properties-d_156.html

Submitted: November, 2011

Revised: January 20, 2012

Accepted: March 30, 2012

Automatic Detection of QRS Complex, P-Wave and T-Wave in the Electrocardiogram

Martin Bachler^{1,*}, Christopher Mayer², Bernhard Hametner¹, Siegfried Wassertheurer²

¹ Department of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *martin.bachler@student.tuwien.ac.at

² AIT Austrian Institute of Technology, Health & Environment Department, Business Unit Biomedical Systems, Muthgasse 11, 1190 Vienna, Austria

Abstract. Every third death in developed countries is caused by cardiac diseases, which are the number one cause of death. Duration and dynamic changes of certain intervals of the ECG are well established indicators in the diagnosis of cardiac diseases. Furthermore, several agencies require the assessment of the effect of newly developed drugs on the QT interval.

Automated measurement and annotation of the ECG shows numerous advantages over manual methods, therefore the long term aim is to develop an all-in-one device for data acquisition and ECG analysis. The development process is conducted in different stages, whereas the first step and short term aim described in this paper consists of creating algorithms in MATLAB® and validating them against ECG signals manually annotated by medical experts. This early stage is followed by porting all algorithms to the aimed platform and finally by hardware-in-the-loop simulations coupling the measurement hardware with the MATLAB® model.

The presented algorithm detects R peaks based on the signals amplitude and first derivative as well as RR intervals. False positive detections due to artifacts are prevented by analyzing the signal's local statistic characteristics. These intermediate results are automatically classified to distinguish normal heartbeats from potential premature ventricular contractions. QRS complexes, P and T waves are detected by their first derivative for each class of heartbeats and are separately refined for each detected heartbeat.

The algorithm has been verified against four PhysioNet databases and achieved a sensitivity of 98.5% and a positive predictive value of 98.3%, respectively.

These results are promising, but further work is still required to implement the algorithm on an embedded system to build an easy to use all-in-one device.

Introduction

Every third death in the United States and actually nearly every second one in Europe is caused by cardiovascular diseases [1, 2]. Their main forms are coronary heart diseases, causing nearly half of all deaths caused by cardiovascular diseases. Coronary heart diseases are the most and second most common cause of death in Europe and the United States, respectively [1, 2].

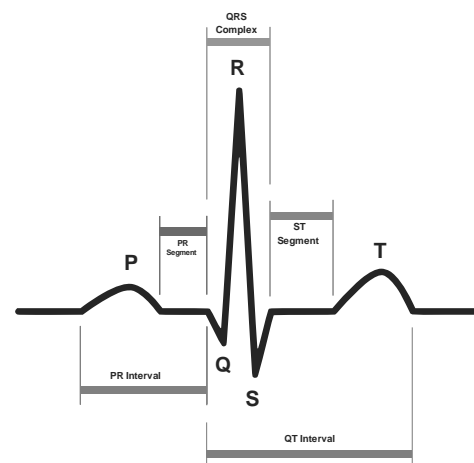


Figure 1. Schematic representation of a normal heartbeat and its features seen on ECG (modified from [3]).

Electrocardiography (ECG) is a widespread, non-invasive and painless technique to measure physiologic activity and pathologic changes of the myocardium. As shown in Figure 1, the tracing of one heartbeat consists of a P wave (atrial depolarization), a QRS complex (ventricular depolarization) and a T wave (ventricular repolarization). Several well-defined segments and intervals between these features are well established indicators in the diagnosis of cardiac diseases, most notably the PR interval (from the onset of the P wave to the onset of the QRS complex) and the QT interval (from the onset of the QRS complex to the offset of the

T wave). In addition, some non-antiarrhythmic drugs may have the undesired property of prolonging the QT interval, therefore several agencies and national regulators require the assessment of this effect in newly developed drugs [4]. Automated methods for measurement and annotation of the ECG offer several advantages over manual ones, such as immunity to observer related errors and operator fatigue, higher accuracy in repeated measurements and faster or more extensive testing at lower cost.

In the last decades, a lot of ECG analysis methods have been presented. Especially the rapid development of powerful computing hardware led to a widespread application of software ECG annotation algorithms in the last 30 years. Despite the usage of many different approaches such as signal derivatives [5], digital filters, wavelets [6] and neural networks, most methods focus only on the detection of the QRS complex [7]. Other software algorithms extend existing QRS detectors with the evaluation of QT intervals [8, 9] or P waves [6], but these methods are only suitable for offline ECG analysis. This paper presents an algorithm combining some of these methods and adopting them for online (real time) measurements.

1 Development Process

To facilitate the use of full automatic ECG annotation, the aim is to develop an all-in-one-device for ECG acquisition and analysis. As Figure 2 shows, the process of development is conducted in several stages, starting with offline prototyping and verification using MATLAB®, followed by porting the algorithm to an embedded system and finally performing hardware-in-the-loop simulations to validate the functionality.

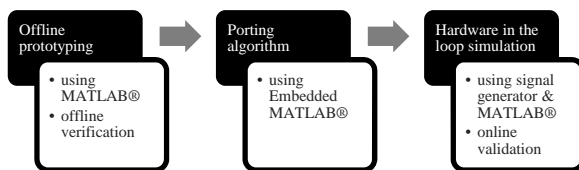


Figure 2. Overview of the development process.

MATLAB®, a numerical computing software developed by MathWorks™, allows easy matrix and vector manipulations and can interface with other programming languages including C. These properties make MATLAB® the ideal choice for rapid development of signal processing algorithms intended to be used on

digital signal processors. Besides, based on its scripting features, it allows the automation of the verification process. The developed algorithm has been verified against ECG signals manually annotated by medical experts from different PhysioNet databases [10, 11].

After its successful validation, the algorithm is ported to an embedded system containing a digital signal processor. Embedded MATLAB®, a subset of the MATLAB® language, supports efficient code generation for deployment for embedded systems and therefore is the optimal choice for this task.

Finally the embedded system is validated using a hardware-in-the-loop simulation. In this step, the final system is ready to use, but instead of measuring ECGs in real subjects, they will be simulated using a signal generator controlled by MATLAB®. A hardware-in-the-loop simulation allows the reproduction of previously annotated signals and hence an efficient verification of the results as well as a validation of the final device.

2 Measurement Algorithm

R peaks are the most prominent feature in ECG tracings. Thus they can be used as reference point for further features and are a good choice to start detection with. The measurement algorithm continuously tries to detect them based on the signal amplitude and its first derivative [5]. Local statistics of the signal are evaluated to distinguish correctly detected R peaks from artifacts caused by movements of the subject. Once an R peak is found a classification is applied in real time to separate normal QRS complexes from potential premature ventricular contractions. Creating templates by averaging the signals reduces noise and allows a more accurate detection of all further features. Subsequently, QRS on- and offset are detected based on the signal's local amplitude. T and P waves as well as their on- and offsets are derived based on their first derivatives [9]. QRS on- and offset as well as all parts of T and P waves are primarily detected in the template of their respective class. The actual signal is only used for local refinement.

To detect R peaks, a feature signal is continuously calculated as follows:

- Calculate the first discrete derivative D_t of the signal S_t
- $$D_t = S_t - S_{t-1} \quad (1)$$
- Evaluate the amplitudes SA_t of S_t and DA_t of D_t within a moving window ($w = 60$ ms)

$$SA_t = \max(S_{(t-w)...t}) - \min(S_{(t-w)...t}) \quad (2)$$

$$DA_t = \max(D_{(t-w)...t}) - \min(D_{(t-w)...t}) \quad (3)$$

- Combine SA_t and DA_t

$$C_t = SA_t^2 \cdot DA_t \quad (4)$$

- Calculate feature signal FS_t within a moving window ($w = 100$ ms)

$$FS_t = \max(C_{(t-w)...t}) \quad (5)$$

- Use the mean value of the last 2 seconds of FS_t as threshold Th_t ($w = 2$ s)

$$Th_t = \frac{1}{w} \sum_{k=t-w}^t FS_k \quad (6)$$

Figure 3 shows three different ECG signals (containing noise and small artifacts; high T wave; sudden subject motion) with their corresponding feature signals. The feature signal is robust regarding noise, small artifacts and prominent T waves, but does respond to sudden motion of the subject.

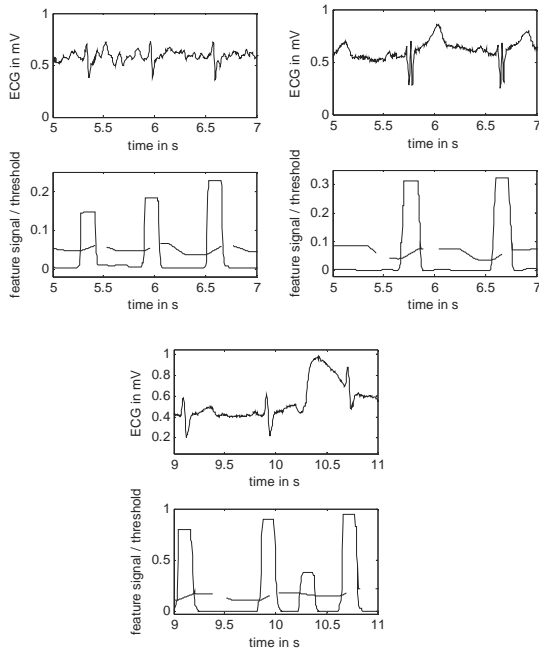


Figure 3. Differently shaped ECG signals with their corresponding response of the feature signal FS_t (equation 5) and its threshold Th_t (equation 6).

Whenever the feature signal exceeds the threshold, the following statistic criterions have to be fulfilled. Otherwise the detected part of the signal is interpreted as an artifact:

- Standard deviation σ within the last 400 ms

$$FS_t > Th_t + 6 \cdot \sigma^3 \quad (7)$$

- Kurtosis β_2 within the last 2.5 seconds

$$\beta_2 > 4 \quad (8)$$

Within each region containing an R peak, differences between local minima and maxima are calculated. The maximum with the highest difference to its surrounding minima is chosen as the exact position of the R peak. If more than two R peaks have already been detected, a template is built by averaging them. The correlation between the newly detected R peak and the template is computed as similarity measure and has to exceed a predefined threshold. Otherwise the R peak is discarded.

To avoid not detecting potentially missed or wrongly discarded R peaks, the intervals between two consecutive R peaks (RR interval) are calculated. If one RR interval exceeds 1.8 times the previous ones, the section between those R peaks is searched again with lower thresholds.

Classification of the R peaks is performed in real time. The ECG signal at each R peak ± 0.5 seconds is compared to a predefined number of classes using correlation and is assigned to its most similar class. If classes among each other correlate better than with the current R peak, these classes are merged and a new class is created from the current signal. This approach results in dynamically evolving classes, continuously enhancing with the duration of the measurement.

To reduce noise, templates are created by averaging each class and are used for the detection of all subsequent features (QRS on- and offset, P and T wave as well as their on- and offsets). Only a local refinement is performed using the original signal.

To detect the onset of the QRS complex, an interval of 150 ms straight before the R peak of the template is analyzed as follows:

- Calculate the amplitudes TAt and TDA_t of the template Tt and its first discrete derivate within a moving window ($w = 30$ ms)

$$TA_t = \max(T_{(t-w)...t}) - \min(T_{(t-w)...t}) \quad (9)$$

$$TDA_t = \max(T'_{(t-w)...t}) - \min(T'_{(t-w)...t}) \quad (10)$$

- Calculate a threshold TT and TD for the amplitudes TA_t and TDA_t

$$TT = c_1 \cdot (\max(TA_t) - \min(TA_t)) + \min(TA_t) \quad (11)$$

$$TD = c_2 \cdot (\max(TDA_t) - \min(TDA_t)) + \min(TDA_t) \quad (12)$$

with c_1 and c_2 being predefined constants.

- The point closest to the R peak, where TA_t is below TT or TDA_t is below TD is marked as reference for the QRS onset.
- The exact QRS onset is found within a 40 ms region prior to the reference point at the closest extremum to the R peak in the original signal. If no extremum is present, the point with the lowest slope is considered to be the QRS onset.

The computation of the QRS offset is very similar to the onset, with two exceptions:

- In order to annotate prolonged QRS complexes correctly, the analyzed interval is chosen larger.
- In equation (9) and (10), window w is 60 ms.

The peak of the T wave is detected by a special ‘wings’ function W as described by Christov and Simova [9] as follows:

$$W_1 = T_{t-40ms} - T_t \quad (13)$$

$$W_2 = T_t - T_{t+40ms} \quad (14)$$

$$W = W_1 \cdot W_2 \quad (15)$$

The ‘wings’ function is applied to the template T_t between the previously detected QRS offset and the end of the template (0.5 seconds after the R peak). As shown in Figure 4, the position of the minimum of the ‘wings’ function represents the peak of the T wave, regardless of the polarization of the T wave. Subsequently, the position of the peak of the T wave is refined using the original signal by finding a local minimum or maximum, depending on the wave’s polarization.

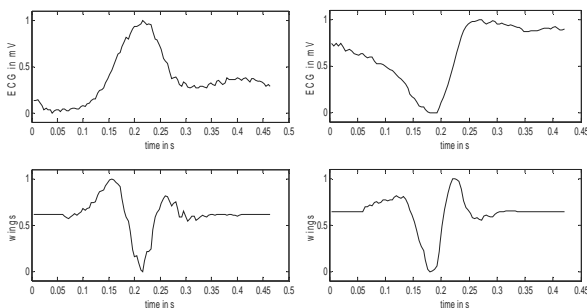


Figure 4. Top: Different T waves (left: positive T wave, right: negative T wave). Bottom: corresponding ‘wings’ function.

The part of the template between QRS offset and T peak is searched for the closest extremum to the T peak, which is used as reference point for T onset. If no extremum is present, the point with the flattest slope is used instead. Like the T peak, the T onset is refined by finding a local extremum in the original signal. In some cases, especially in ECG traces showing prolonged duration of the QRS complex, T wave and QRS complex overlap. In this case, no reasonable T onset point can be found, even not by medical experts. This condition leads to detection of the T onset straight before the T peak, allowing easy recognition. In this case, no T onset will be annotated at all.

The detection of the T offset is performed in a similar manner between the T peak and the end of the template (0.5 seconds after the R peak).

To detect the peak of the P wave, the interval between the preceding T offset and the current QRS onset is evaluated in the template by a slightly altered ‘wings’ function (equation 13, 14, 15) which responds just to positive peaks. Again, the peak is located at the minimum of the ‘wings’ function. P onset and P offset are found in the same way as T onset and T offset.

Due to very small amplitude or high noise level, the P wave in some ECG recordings is indiscernible. To prevent the algorithm from false detections, the amplitude of the P wave (derived as difference in voltage between the P peak and the mean of P onset and P offset) has to exceed a certain fraction of the amplitude of the QRS complex, otherwise no P wave (peak, onset and offset) is detected at all.

3 Results

PhysioNet databases are a collection of recordings of different physiological modalities such as electroencephalogram, electrocardiogram, blood pressure, respiration and others. Depending on their objective, several databases contain different kinds of annotations [10, 11]. Hence, annotations done by medical experts can be used to verify automated algorithms. The following databases have been chosen for the verification of the algorithm presented in this paper due to a wide range of different ECG signals as well as a reasonable amount of expert annotations:

- QT Database, created to evaluate algorithms detecting the QT interval [12].
- AF Termination Challenge Database, designed to be used in ‘Computers in Cardiology Challenge 2004’.

- MIT-BIH Arrhythmia Database, test material for evaluation of arrhythmia detectors.
- Fantasia Database, originally used for testing automated arrhythmia detection [13].

The presented algorithm was tested against all four databases with respect to the detection rate of QRS complexes. The American National Standards Institute (ANSI) essentially recommends two parameters for the evaluation of the detection rate [14]: the sensitivity Se

$$Se = \frac{TP}{TP + FN} \quad (16)$$

and the positive predictive value PPV

$$PPV = \frac{TP}{TP + FP} \quad (17)$$

where TP is the number of true positive, FN the number of false negative and FP the number of false positive detections.

A sensitivity of 98.5% and a positive predictive value of 98.3% were achieved in the verification of the detection rate. Time differences between detected and corresponding annotated points of the QT database are shown in Table 1. Durations of the PR interval, the QRS complex and the QT interval were calculated from the results of the algorithm and the expert annotations, respectively. The differences between the results of the algorithm and the annotations are shown in Figure 5 as Bland-Altman diagrams [15]. Mean and standard deviation of these differences are -1.1 ± 19.9 ms for the PR interval, 3.6 ± 16.5 ms for the QRS complex and -4.8 ± 35.3 ms for the QT interval. These results are satisfying and match existing offline algorithms [7, 8, 16].

| Feature | Algorithm-expert deviation | | Inter-expert deviation | |
|-------------------|----------------------------|---------|------------------------|---------|
| | Mean | Std | Mean | Std |
| P onset | 2.3 ms | 23.8 ms | * | * |
| P peak | 0.5 ms | 22.2 ms | * | * |
| P offset | - 0.3 ms | 27.2 ms | * | * |
| QRS onset | 0.5 ms | 10.2 ms | 3.8 ms | 14.2 ms |
| R peak | - 9.1 ms | 14.4 ms | 0.1 ms | 2.4 ms |
| QRS offset | 4.3 ms | 12.5 ms | 2.7 ms | 17.0 ms |
| T onset | 10.8 ms | 63.0 ms | 9.5 ms | 44.9 ms |
| T peak | - 3.3 ms | 33.1 ms | 3.5 ms | 30.0 ms |
| T offset | - 4.2 ms | 38.8 ms | 5.8 ms | 39.9 ms |

Table 1: Means and standard deviations of differences in time between annotated and detected points as well as between different expert annotations. (* Annotated by one expert only).

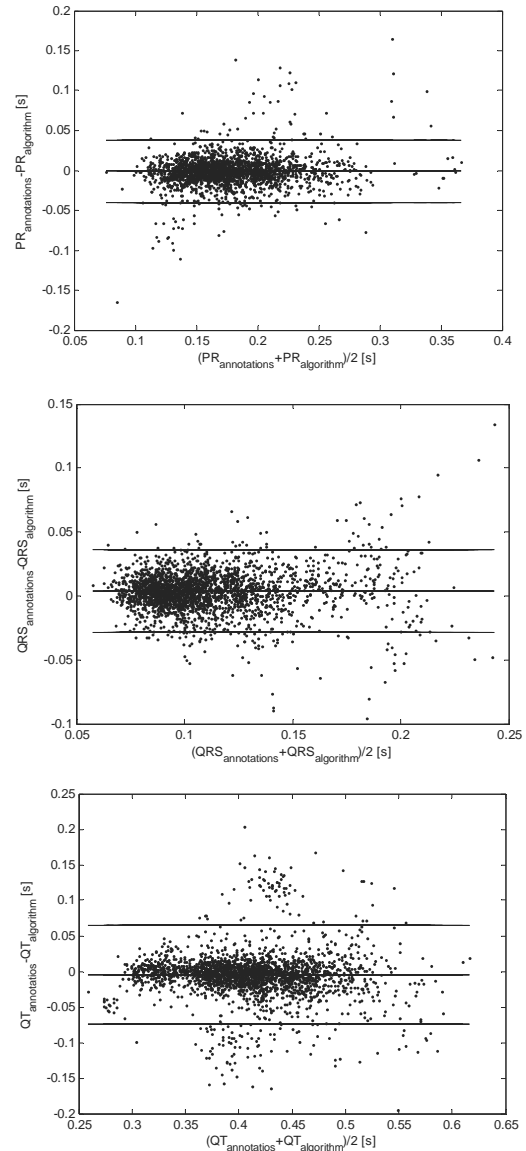


Figure 5. Bland-Altman diagrams comparing the duration of certain intervals derived from expert annotations with the results of the algorithm. Top left: PR interval. Top right: QRS complex. Bottom: QT interval.

4 Discussion and Conclusion

In most cases, the average deviations shown in Table 1 are in the range of the sampling interval of 4 ms, suggesting an insignificant error. The standard deviations of the differences between algorithm results and expert annotations also match with those of inter-expert annotations. Reflecting the uncertainty regarding the exact position of these features among experts, these results suggest that the presented algorithm performs approximately as well as humans.

The two outliers in Table 1, R peak and T onset, are partially arising from bad or unusual annotations in the QT Database. R peaks are sometimes annotated at negative peaks within the QRS complex, whereas they ought to be positive by definition. The position of the T onset point is often ambiguous due to overlapping of the T wave with the QRS complex. This fact is also reflected by the high standard deviation of the inter-expert deviations. Nevertheless, even unambiguous T onset points often have not been annotated in the QT database, making reasonable comparison with the algorithm difficult.

Bland-Altman diagrams comparing the annotated and calculated durations of the PR interval, the QRS complex and the QT interval in Figure 5 do not show any trends and thus do not suggest any methodical error. Their means and standard deviations are in the same range as those in Table 1 and therefore do not show any abnormalities.

The results of the offline verification process are promising. Further work is required to implement the algorithm on an embedded system and run hardware-in-the-loop simulations for validation in order to build an all-in-one device for ECG measurement with real time annotation.

References

- [1] V.L. Roger, A.S. Go, D.M. Lloyd-Jones, R.J. Adams, J.D. Berry, T.M. Brown, M.R. Carnethon, S. Dai, G. de Simone, E.S. Ford, and others: *Heart disease and stroke statistics-2011 update: A report from the American Heart Association*. *Circulation* 123 (2011), e18-e209.
- [2] S. Allender, P. Scarborough, V. Peto, M. Rayner, J. Leal, R. Luengo-Fernandez, and A. Gray: *European cardiovascular disease statistics*. European Heart Network (2008)
- [3] A. Atkielski: *SinusRhythmLabels*. <http://en.wikipedia.org/wiki/File:SinusRhythmLabels.svg> (23:40, 13 January 2007).
- [4] U.S. Department of Health and Human Services - Food and Drug Administration and Center for Drug Evaluation and Research (CDER) and Center for Biologics Evaluation and Research (CBER): *Guidance for Industry: E14 Clinical Evaluation of QT/QTc Interval Prolongation and Proarrhythmic Potential for Non-Antiarrhythmic Drugs*. Regulatory Information (10 2005).
- [5] V.S. Chouhan, and S.S. Mehta: *Detection of QRS complexes in 12-lead ECG using adaptive quantized threshold*. *IJCSNS* 8 (2008), 155-63.
- [6] A. Diery, D. Rowlands, T.R.H. Cutmore, and D. James: *Automated ECG diagnostic P-wave analysis using wavelets*. *Computer Methods and Programs in Biomedicine* (2010).
- [7] B.U. Köhler, C. Hennig, and R. Orglmeister: *The principles of software QRS detection*. *IEEE Engineering in Medicine and Biology Magazine* 21 (2002), 42-57.
- [8] D. Hayn, A. Kollmann, and G. Schreier: *Automated QT interval measurement from multilead ECG signals*. *Computers in Cardiology* (2008), 381-384
- [9] I. Christov, and I. Simova: *Fully automated method for QT interval measurement in ECG*. *Computers in Cardiology* (2008), 321-324
- [10] A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.Ch. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.-K. Peng, and H.E. Stanley: *PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals*. *Circulation* 101 (2000), e215-e220
- [11] PhysioNet: *PhysioBank*. <http://physionet.org/physiobank/database> (19:21, 15 April 2011).
- [12] P. Laguna, R.G. Mark, A. Goldberg, and G.B. Moody: *A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG*. *Computers in Cardiology* (1997), 673-676.
- [13] N. Iyengar, C.-K. Peng, R. Morin, A.L. Goldberger, and L.A. Lipsitz: *Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics*. *American Journal of Physiology* 271 (1996), R1078-R1084
- [14] ANSI/AAMI EC57: *Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms*. AAMI Recommended Practice/American National Standard (1998). Available: <http://www.aami.org>; Order Code: EC57-293.
- [15] J.M. Bland and D.G. Altman: *Statistical methods for assessing agreement between two methods of clinical measurement*. *The LANCET* 1 (1986), 307-310
- [16] G. Schreier, D. Hayn, and S. Lobodzinski: *Development of a new QT algorithm with heterogenous ECG databases*. *Journal of electrocardiology* 36 (2003), 145-150.

Submitted: October 2011

Revised: January 2012

Accepted: March 15, 2012

Evaluation and Adaptation of Techniques for Higher Index DAE with Respect to Real-Time Simulation

Jörg Frochte

Bochum University of Applied Sciences, Dept. Electrical Engineering and Computer Science,
Höselers Platz 2, 42579 Heiligenhaus, Germany; joerg.frochte@hs-bochum.de

Abstract. In this paper we will evaluate approaches to the simulation of DAE of higher order under real-time conditions. Some of these approaches are new variants of well-known methods. For the purpose of evaluation we used, for example, explicit multistep methods with and without subsequent inexact projection, and BDF approaches under real-time termination conditions. The Hardware in the loop (HIL) Simulation requires plant models which can be simulated under hard real-time conditions. Thus the combination of DAE and real-time simulation is very interesting for model-based generation of plant models because the so-called hybrid differential algebraic equations (HDAE), see e.g. [11] Appendix C, are the mathematical foundation of modeling languages such as Modelica and Simscape. So beyond the numerical properties of a technique we will watch out wherever they are suitable for model-based code generation.

Introduction

The modeling process takes place on several levels of abstraction: In a certain way, the top level is the least abstract because on this level one can still deal with the real object, which is set by reality. The real object serves as a model for the ‘physical model’. This process is based on physical simplification: Geometrical details and effects, e.g. temperature dependency, need to be neglected. The mathematical representation of this physical model is now the ‘mathematical model’. This mathematical model is taken to construct a numerical model. The model finally gained is then used in simulation studies.

Modelica tools or Simscape give the modeler an opportunity to design the plan model primarily on the physical level and not on the mathematical one. Latter one is carried out with the help of modeling tools such as Simulink or by directly coding the model to C/C++.

If these model based code generation tools were used in a perfect world, there would be only little need for the modeler to bother about the mathematical level.

On the numerical level there would be no bother at all. Unfortunately this is not the case up to now. Modelers still need a lot of knowledge about all levels of modeling, including the numerical level. This is particularly true for real-time simulation.

The purpose of this paper is to improve this type of simulation by evaluating and adapting established methods for real-time use case. It is well known that the numerical simulation of DAE with a high index requires a high level of effort and care. As part of an automatic code generation from physical models some approaches in the literature are hard to apply, if at all. The reason is that they require more knowledge than a DAE system without modeling-context can provide to a code generation software. Even with more data about the context the skills how to treat the system are hardly capable to be set in today's software expert systems. Beyond this in some techniques, e.g. Baumgarte's method, parameters depend on the discretization and used integration method.

This has already been pointed out by Ascher et al. [1] and seems to be still state of the art for modern simulation tools, see [14]. In consequence, we do e.g. not consider Baumgarte's method in this paper because it seems impossible up to now to choose an optimal set of parameters automatically for such a general use case. Neither do we consider approaches limited to special application areas such as mechanical models because we are looking for general approaches to multi-physical tools.

1 Real-time Requirements - Test Case

Performance must be defined with respect to whether there are real-time requirements or not. Most numerical simulations are carried-out without real-time requirements.

In such cases ‘performance’ means that the simulation is completed as quickly as possible. This goal is frequently followed by means of continually or temporarily large time steps. The latter are the outcome of adaptive techniques. On the one hand, a time step of $\Delta t = 1e-3$ seconds is in most simulations considered to be abnormally small. Hence, it would only be used in an emergency for a limited number of time steps. On the other hand, $\Delta t = 1e-3$ seconds is the typical time step size of a HIL simulation with hard real-time requirements.

Hence, it is fairly reasonable to explore whether techniques, generally considered as inefficient in the literature, may perform well in real-time simulation. In the context of many approaches this question has not been answered yet because real-time simulation gains only little attention in numerical research. Even less attention is paid to high index DAE in this context. For example, explicit Runge-Kutta approaches combined with projection methods are often regarded to be quite efficient techniques in the field of differential algebraic equations with a high index (see e.g. Hairer et. al. [7]). But the used Runge-Kutta style solver DOPRI5 is unsuitable for real-time simulation, even if it is the basis of the MATLAB and Simulink solver ode45 and so nearly an industrial standard because as an adaptive solver it is designed to achieve accuracy and stability by chosen variable time steps sizes.

We can therefore summarize that we are looking for approaches with fixed time step. The CPU costs per time step must be predictable, and the technique must meet high requirements with respect to stability and accuracy.

As test case we are going to use the Lagrangian formulation of a mathematical pendulum with a mass and a line length of one.

$$\frac{dy}{dt} = u; \frac{du}{dt} = \lambda x; \frac{dy}{dt} = v; \frac{dv}{dt} = \lambda y - g$$

$$0 = x^2 + y^2 - 1$$

We take this set of equations because it is a straightforward standard example of an index 3 problem, which is, for instance, discussed in [2], [7], [5]. So there are plenty of data available for. In order to compare the results with [5] and to obtain an exact period of 2, we choose $g=13.7503716373294544$.

The initial conditions are $x = 1, y = 0, u = 0$ and $v = 0$. It is well known (e.g. [7] p. 454ff) that this index-3 formulation can be transferred to index-1 or index-0 formulation by repeatedly differentiating the algebraic constraint and replace it by one of its derivatives.

$$\begin{aligned} \text{index} = 3 & \quad 0 = x^2 + y^2 - 1 \\ \text{index} = 2 & \quad 0 = x \frac{dx}{dt} + y \frac{dy}{dt} \\ \text{index} = 1 & \quad 0 = \lambda x^2 + \lambda y^2 - g y + u^2 + v^2 \\ \text{index} = 0 & \quad 0 = x^2 \frac{d\lambda}{dt} + 2\lambda x u + y^2 \frac{d\lambda}{dt} + 2\lambda x v - \\ & \quad g v + 2u \frac{du}{dt} + 2v \frac{dv}{dt} \end{aligned}$$

As we can see, it is in the index-1 formulation easy to express lambda out of the other values:

$$\lambda = \frac{-g y + u^2 + v^2}{x^2 + y^2}$$

In other words, no matter if we choose to use an index-1 integrator such as IDA or a standard ODE integrator like CVODE - both are part of the Sundial software collection [8].

Generally it makes sense to stop at an index-1 formulation, as long as the formulation can be transformed into a form that is compatible to an ODE. But first we should explain why the way most Modelica tools deal with these kinds of equations seems to be unsuitable for real-time purposes..

2 Considered and Adapted Techniques

2.1 Simulation using the Pantelides Algorithm and Dummy Derivatives

Most Modelica tools use a mixture of the dummy derivatives approach [10] with the Pantelides Algorithm [12], which was designed to initialize DAE. The resulting algorithm is described by Cellier and Kofman (see [2]). One reason for using it might be that it leads to a straightforward code generation that can be used with standard ODE integrators.

The discussion by Cellier and Kofman ([2], chapter 7.8) e.g. shows some problems that occur when this algorithm is applied to the index-3 formulation of the pendulum.

To cut a long story short: it fails completely, no matter if it is combined with techniques like inline- or mixed-mode integration.

One may argue, as Cellier and Kofman do, that a smart modeler should use the following index-1 formulation to describe the pendulum.

$$\frac{d\varphi}{dt} = \theta; \frac{d\theta}{dt} = -g \cdot \sin(\varphi)$$

$$x = \cos\left(\varphi + \frac{3\pi}{2}\right); y = \sin\left(\varphi + \frac{3\pi}{2}\right)$$

On the one hand, this represents a fairly sensible strategy and on the other hand, this means that we limit our options in the fields of library and model construction. Furthermore, we postulate that users are able to identify the point at which the differential index of a formulation rises especially in a more complex and connected model. Relying on the skilled modeler with deep knowledge in all modeling levels seems to be inappropriate. This is particularly true when the objective lies in the model-based generation of a software from a mathematical or physical modelling level.

However, not all hope is lost because the Pantelides variant from [2] can handle this, at least for considerable number of problems, using a technique known from Dymola as ‘dynamic state selection’. In the context of real-time simulation this workaround is associated with a number of drawbacks. In the very instance in which a state-switch becomes necessary, a multiple of the original CPU time is needed per time step. In general, this causes an overrun in the real-time simulation. Let us recapitulate that predictability of CPU-time is critical for real-time simulation. Otherwise we would always have to calculate our resources for the biggest possible CPU resource need during the simulation. Another point to be made is that this Pantelides variant is not fully theoretically understood.

For example: For a while a lot of people thought that the structural index used by the Pantelides algorithm is +/-1 the same as the differential index of the numerical theory, but Reissig, Martinson and Barton manage to demonstrate (see [13]) that in some cases an equation of a differential index 1 may have an arbitrarily high structural index.

Now it is proved that the Pantelides algorithm might transform a totally harmless set of equations into an unsuitable appearance.

So in fact, the Pantelides algorithm and its variants - with or without dummy derivatives - are no silver bullets for higher index problems. Thus, it seems that the real-time simulation of DAE with higher index calls in for alternatives. We assume that they are already on the market but have not yet been modified for the use under real-time conditions. For this reason we will examine approaches and present slight modifications for real-time simulation in the following section.

2.2 Integration of the Index-1 Formulation with Multistep Method

So now we use the Index-1/Index-0 formulation with standard ODE-Solvers.

$$\lambda := \frac{-gy + u^2 + v^2}{x^2 + y^2}$$

$$\frac{dy}{dt} = u; \frac{du}{dt} = \lambda x; \frac{dy}{dt} = v; \frac{dv}{dt} = \lambda y - g$$

A lot of people associate real-time and HIL with the explicit Euler algorithm. One reason for this might be that it is easy to generate code with an embedded explicit Euler from Simulink using the RTW. Unfortunately, the results for stiff systems are often devastating, as Figure 1 shows.

As a consequence, ‘.. for physical models, MathWorks recommends implicit solvers, such as ode14x, ...’, [9], or to switch for a better performance to local approaches. In this approach the local solver is by default the implicit Euler, in other words BDF(1). So maybe the implicit Euler is an approach for our ODE above? In the numerical community it is quite well known that it is not.

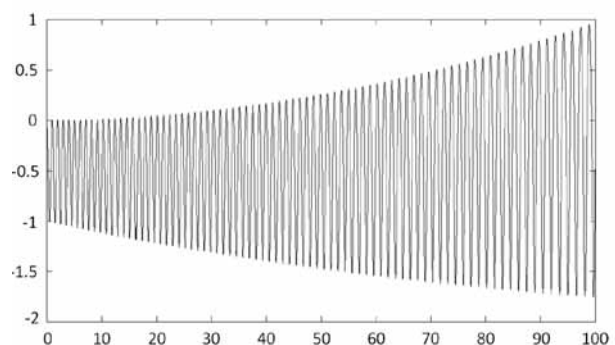


Figure 1. y-coordinate of the pendulum computed with the explicit Euler algorithm $\Delta t = 0.001$.

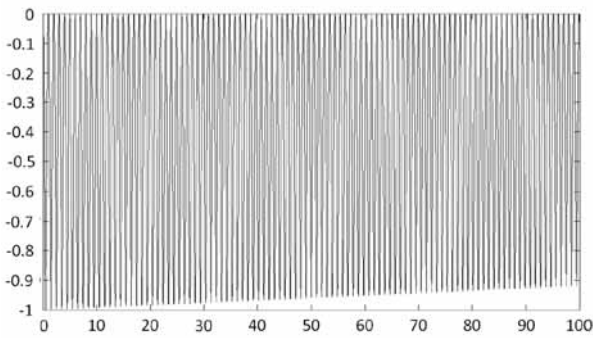


Figure 2. BDF(1), drift effect in y -coordinate with $\Delta t = 6:25e-5$.

The implicit Euler is afflicted with the drift effect (see Figure 2), and unstable for this problem. But there are numerous alternatives with fixed timestep size.

Adams-Bashforth Methods. The failure experienced when using the Euler does not mean that all explicit methods fail. Multi-step methods like the Adams-Bashforth Methods, see e.g. [6] p. 357f. or [2] p. 122f. for details, share a number of features that we require: Firstly CPU costs are low, just one evaluation of the given right side f , and secondly one can achieve high accuracy if the solution is smooth enough. In general the third order is a good compromise between stability and accuracy. So we try

$$y_{n+1} = y_n + h \cdot \left(\frac{23}{12} f(t_n, y_n) - \frac{26}{12} f(t_{n-1}, y_{n-1}) + \frac{26}{12} f(t_{n-2}, y_{n-2}) \right)$$

The advantage of multi-step methods lies in the use of the history of the simulation. A minor drawback might be the start phase. At the first time step there obviously is no simulation history to rely on. For this reason we have two major strategies: We start with an order 1 approach for the first time step, next perform an order 2 approach and then continue the simulation with the order 3.

Unfortunately, one loses some accuracy in this starting phase because of the two steps with lower order. To minimize this effect one should split the first step into three smaller ones. For a real-time simulation this probably results in an overrun in the first time step, but on most HIL-Systems one overrun in the first time step is no problem (see e.g [4]).

The other strategy is to assume that the initial situation has endured for a longer period of time, hence creating an artificial history. For our example this means $x(-2\Delta t) = x(-\Delta t) = x(0) = 1$; $y(-2\Delta t) = y(-\Delta t) = y(0) = 0$ and $u(-2\Delta t) = u(-\Delta t) = u(0) = 0$; $v(-2\Delta t) = v(-\Delta t) = v(0) = 0$. In other situations, however, its not physical. In other words, it cannot be generalized. The reason why we use it is to avoid initialization effects for the numerical test.

Backward Differentiation Formulas. Backward Differentiation Formulas, for a detailed description see [7] p. 246ff. or [2] p. 128ff., are in contrast to Adams-Bashforth Methods implicit linear multistep methods. These methods are particularly suitable for solving stiff differential equations. Our ODEs generated out of DAE are very often fairly stiff, so they seem to be a natural approach.

However, the use of implicit methods is objectionable because they seem to undermine the predictability of CPU resources. They would iterate until a given error boundary is reached. Knowing this, we use a fairly widespread approach to make the algorithm predictable. We limit the number of Newton-iterations. The Number of Newton iterations needed to fulfill a given limit of error depends, of course, both on the initial value and on the problem as a given, unchangeable factor. If we take the value from the last time step as the initial value for the next time step, it becomes clear that a smaller time step size leads to better initial values.

In our test case with time step size $\Delta t = 1e-3$ the average number of Newton iterations per time step to achieve a residual smaller than $1e-10$ is 2. Because of its good mixture of stability and accuracy we choose the BDF(3) scheme.

$$\frac{11}{6} y_{n+1} - 3y_n + \frac{3}{2} y_{n-1} - \frac{1}{3} y_{n-2} - h \cdot f(t_{n+1}, y_{n+1}) = 0$$

For the initial values back in time we use the same approach as for the Adams-Bashforth Methods. Higher order BDF methods have good properties concerning DAE of low order index anyway. That is one reason why they are the basis of the DAE-Index-1-solver DASSL. Therefore it makes sense to test them for our transformed index-1 problem.

The first challenge is: Can we avoid a drift effect using the typical HIL time step size? As you can see from the table in section 4 the approach looks very stable.

3 An Inexact Projection Approach

Concerning stability, numerical literature identifies a residual risk in the restrictive application of high order BDF or Adams-Bashforth methods to high order DAE.

Maybe the typical small time step sizes in real-time simulation save our necks but we would like to explore whether stability can be improved by projection techniques (see e. g. [7], p. 470ff.). Unfortunately, the methods applied to this problem generally make use of information that an automatic code generator often does not have.

The process of embedding these data is quite difficult. The successful application of these approaches, e.g. discussed in the paper [5], to a huge set of equations, which are not sorted by application domain, derived from a multi-physical tool such as Modelica tools, seems very unlikely at this moment. It seems easier to use a technique based on the concept of dealing with high order DAE by using overdetermined approaches (see e.g. [7] p. 477ff.). In this case one just assembles all the equations that arise during the generation of the index-1- formulation. This leads to a set of equations, containing more equations than degrees of freedom. In our test case this will look like this:

$$\begin{aligned} \frac{dy}{dt} &= u; \quad \frac{du}{dt} = \lambda x; \quad \frac{dy}{dt} = v; \quad \frac{dv}{dt} = \lambda y - g \\ 0 &= x^2 + y^2 - 1; \quad 0 = xu + yv \end{aligned}$$

This analytical problem is in actual fact not overdetermined because we can find a solution, so that the residual is zero. The set of equations can be solved exactly. But we need to discretize these equations to simulate them. We do that using the BDF approach which leads to

$$\frac{dy}{dt} \approx \frac{11}{6} y_{n+1} - 3y_n + \frac{3}{2} y_{n-1} - \frac{1}{3} y_{n-2}$$

and similar expressions for the other derivatives. The result is a discrete system which cannot be solved exactly.

However, this is not our goal anyway. We will perform only a very limited number of iterations of a kind of Gauss - Newton algorithm. The result is an inexact projection approach.

4 Results

We can easily compare the results at $t = 100$ because we know that the period of the solution is 2. There we define

$$RES := |x^2 + y^2 - 1|2$$

and

$$ERR := \Delta x + \Delta y + RES$$

In Table 1, in the column *Newton - Steps* the first summand is the number of steps per time step for the BDF part and the second for the inexact projection. Beyond this, we compute a reference solution based on the formulation presented in section 2.1.

The reference solution is computed by ode15s. The parameter 'RelTol' is set to 1e-12 and 'AbsTol' to 1e-14. The goal of this reference solution is to give us a feeling for the global error and its development. In Figure 3 we can see the difference between the Adams-Bashforth(3) followed by a single inexact projection step.

In Figure 4 the same is displayed for the BDF(3), with only one single newton step followed by a single inexact projection step. Both seems to be very stable. While the implicit approach performs better for $t = 100$, as one can see from the table above, the Adams-Bashforth(3) variant seems altogether preferable.

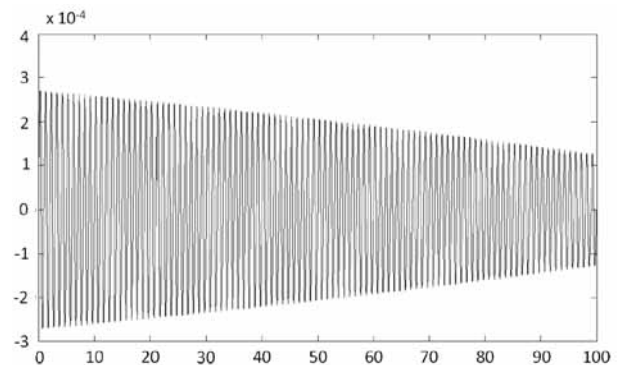


Figure 3. y-coordinate: Comparison Adams-Bash.(3) + projection approach, $\Delta t = 0.00025$ with ode15s.

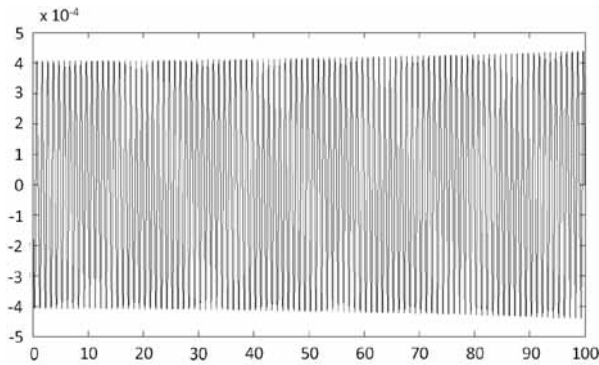


Figure 4. y -coordinate: Comparison Inexact BDF(3) + projection approach, $\Delta t = 0.00025$ with ode15s.

The BDF (3), however, seems to be more stable when it is used in a long term test with a maximum duration of one hour. Its tendency to run out of phase is more pronounced as in the case of the BDF(3), which generally steals a bit of system energy. Beyond this we see that the inexact approach with only one newton step and one projection step per time step performs very well for these typical HIL time step sizes.

Moreover it shows very good reduction rates near the third order in time. We also tried to increase the performance or accuracy of the BDF approach by preconditioning it with the result of the Adams-Bashforth method, but this did not work well.

In general both the explicit and the implicit approach are improved applying a single projection step using over-determined system.

5 Proposal for a Parallel Real-time Algorithm

Our survey demonstrates that a high order BDF or Adams-Bashforth approach combined with a single projection step is the best technique for the problem class discussed in this paper. Wherever we use an explicit or an implicit approach we need at least to assemble a Jacobi matrix and factorize it into a QR- or a LU-form, resp.

If we performed an assembling and a decomposition in every iteration step this would probably be hard to achieve in 1ms real-time. Thus, we intend to turn to a simplified Newton approach as described, for instance in [3] p. 52ff.

However, the shift to a simplified Newton approach will not lead to a real-time compatible technique. Beyond this one has to use the power of current multicore architectures also that tend to be the standard for HILs as well. With four cores it is possible to keep all I/O and OS aspects on one core, e.g. Core4, and use the remaining three cores for the computation. The basic architecture is displayed in Figure 5. As a result, the Jacobi matrix is only updated in the simplified Newton approach if the computation is finished. This procedure results in an algorithm whose turnaround is highly predictable time in real-time simulation.

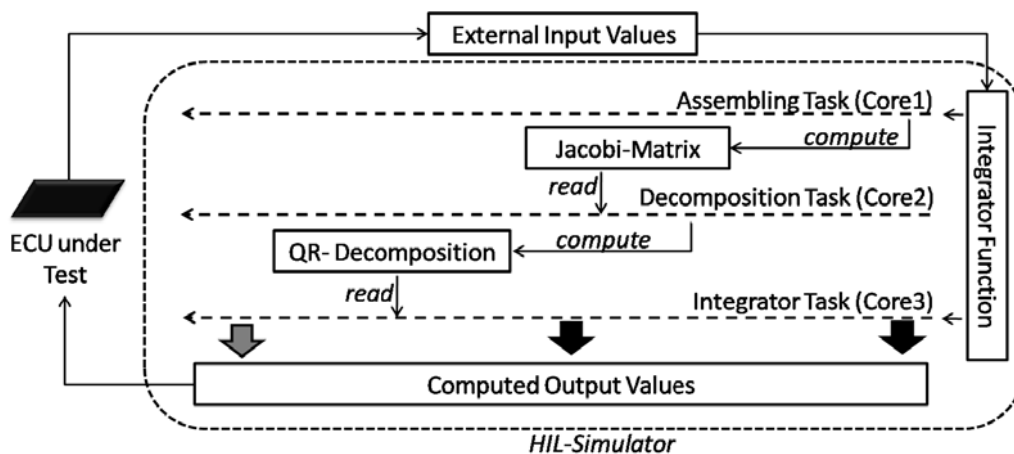


Figure 5. Tasks in a Parallel Real-time approach.

| Method | Newton Steps | Pro-jection | Δt or Tol | Δx | Δy | RES | ERR |
|--|--------------|-------------|---------------------|------------|------------|---------|--------|
| Results achieved with the methods discussed | | | | | | | |
| BDF(3) | 2 + 0 | No | $\Delta t = 1e-3$ | 1.8e-5 | 5.8e-6 | 3.7e-5 | 6.2e-5 |
| Precond-BDF3 | 1 + 0 | No | $\Delta t = 1e-3$ | 2.2e-5 | 1.2e-5 | 4.5e-5 | 8.0e-5 |
| Precond-BDF3 | 1 + 3 | Yes | $\Delta t = 1e-3$ | 1.1e-9 | 4.4e-5 | 2.8e-10 | 4.4e-5 |
| Precond-BDF3 | 1 + 3 | Yes | $\Delta t = 5e-4$ | 2.6e-11 | 5.8e-6 | 1.7e-11 | 5.8e-6 |
| BDF(3) | 1 + 0 | No | $\Delta t = 1e-3$ | 6.4e-5 | 7.6e-5 | 1.2e-4 | 2.7e-4 |
| BDF(3) | 1 + 0 | No | $\Delta t = 5e-4$ | 9.4e-6 | 1.1e-5 | 1.8e-5 | 3.9e-5 |
| BDF(3) | 1 + 0 | No | $\Delta t = 2.5e-4$ | 4.2e-6 | 5.7e-6 | 8.5e-6 | 1.8e-5 |
| Adams-Bash.(3) | 0 + 0 | No | $\Delta t = 1e-3$ | 2.1e-5 | 1.1e-5 | 4.2e-5 | 7.5e-5 |
| Adams-Bash.(3) | 0 + 0 | No | $\Delta t = 5e-4$ | 2.6e-6 | 9.6e-7 | 5.3e-6 | 8.9e-6 |
| Adams-Bash.(3) | 0 + 0 | No | $\Delta t = 2.5e-4$ | 3.3e-7 | 8.0e-8 | 6.6e-7 | 1.0e-6 |
| Adams-Bash.(3) | 0 + 1 | Yes | $\Delta t = 1e-3$ | 2.4e-9 | 7.1e-5 | 2.9e-10 | 7.1e-5 |
| Adams-Bash.(3) | 0 + 1 | Yes | $\Delta t = 5e-4$ | 7.1e-11 | 1.2e-5 | 1.8e-11 | 1.2e-5 |
| Adams-Bash.(3) | 0 + 1 | Yes | $\Delta t = 2.5e-4$ | 2.1e-12 | 2.3e-6 | 1.1e-12 | 2.3e-6 |
| BDF(3) | 1 + 1 | Yes | $\Delta t = 1e-3$ | 1.0e-9 | 4.4e-5 | 1.7e-10 | 4.4e-5 |
| BDF(3) | 1 + 1 | Yes | $\Delta t = 5e-4$ | 2.2e-11 | 5.8e-6 | 1.0e-11 | 5.8e-6 |
| BDF(3) | 1 + 1 | Yes | $\Delta t = 2.5e-4$ | 5.9e-13 | 7.2e-7 | 6.7e-13 | 7.2e-7 |
| Results from [5] with a different projection approach | | | | | | | |
| MKS-DAEOL | - | Yes | $Tol = 1e-5$ | 2.4e-8 | 2.2e-4 | 7.2e-10 | 2.2e-4 |
| MKS-DAEOL | - | Yes | $Tol = 1e-5$ | 9.7e-9 | 1.4e-4 | 5.9e-11 | 1.4e-4 |

Table 1. Results from discussed DAE Solvers.

6 Conclusion and Future Prospects

We gave an outlook on how we hope to produce an algorithm that can use the power of multi-core architectures to simulate industrial size DAE systems in realtime without making too much use of the unreliable approaches discussed in Section 2.1 with their drawbacks and uncertainties. Of course, for problems with a low structural index of 0 or 1 it will always be more ent to use the approaches of Section 2.1. One of our future prospects is to verify that accuracy and stability can be retained using this parallel approach.

Moreover, a deeper analysis of the process will be considered in future publications. We have shown that by using common multistep methods, explicit and implicit ones, it is even possible to simulate high order DAE with a very limited number of iterative steps. A single projection step can improve the results significantly. But beyond this, the presented approaches might support a least square modeling paradigma. When ODE integrators are regarded as preconditioners for an overdetermined system, this obviously leads to a direct modeling of overdetermined systems, which may be quite useful in some applications.

References

- [1] U. M. Ascher, H. Chin, S. Reich. *Stabilization of daes and invariant manifolds*. Numer. Math, 67:131-149, 1993.
- [2] F. Cellier, E. Kofman. *Continuous system simulation*. New York, Springer, 2006.
- [3] P. Deuflhard. *Newton methods for nonlinear problems. Affine invariance and adaptive algorithms*. Berlin: Springer, 2004.
- [4] dSPACE GmbH. *dSPACE FAQ 242 Handling Overrun Situations*. 2011.
- [5] E. Eich. *Convergence results for a coordinate projection method applied to mechanical systems with algebraic constraints*. SIAM J. Numer. Anal., 30(5):1467-1482, 1993.
- [6] E. Hairer, S. P. NNorsett, G. Wanner. *Solving ordinary differential equations. I: Nonstiff problems*. Berlin: Springer, 2010.
- [7] E. Hairer, G. Wanner. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*. Berlin: Springer, 2010.
- [8] A. Hindmarsh, P. Brown, K. Grant, R. S. S.L. Lee, D. Shumaker, C. Woodward. *SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers*. ACM Trans. on Mathematical Software, 31(3):363-396, 2005.
- [9] The MathWorks, Inc. MATLAB 2011a Documentation, Simscape, 2011.
- [10] S. E. Mattsson, G. Soederlind. *Index reduction in differential-algebraic equations using dummy derivatives*. SIAM J. Sci. Comput., 14(3):677-692, 1993.
- [11] Modelica Association. *Modelica Language Specification - Version 3.2*. 2010.
- [12] C. C. Pantelides. *The consistent initialization of differential algebraic systems*. SIAM J. Sci. Stat. Comput., 9(2):213-231, 1988.
- [13] G. Reissig, W. S. Martinson, P. I. Barton. *Differential-algebraic equations of index 1 may have an arbitrarily high structural index*. SIAM J. Sci. Comput., 21(6):1987-1990, 2000.
- [14] G. D. Wood, D. C. Kennedy. *Simulating mechanical systems in Simulink with SimMechanics*. Technical Report 91124v00, 2003.

Submitted: September 2011 (ASIM STS Winterthur)

Accepted: February 15, 2012

An Actor-oriented Approach to ARGESIM Benchmark C05 'Two State Model' using Berkeley Ptolemy II

Patrick Einzinger

dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria; patrick.einzinger@drahtwarenhandlung.at

Simulator. The Ptolemy II project (from the Center for Hybrid and Embedded Software Systems in the Department of Electrical Engineering and Computer Sciences of the University of California in Berkeley) studies modelling and simulation of concurrent, real-time, embedded systems which combine the continuous dynamics of physical systems with discrete mode changes (for example from a digital controller). Models in Ptolemy II consist of so-called actors - visually represented in block diagrams - which are able to communicate with each other by sending messages over ports and channels.

Actors and their connections describe an abstract syntax of the model; additionally one has to specify a 'model of computation' (like continuous time or discrete event) which gives operational rules for the execution of the model. Ptolemy II supports heterogeneous and hierarchical mixtures of different models of computation. It is constructed in Java and the source code is freely available (<http://ptolemy.eecs.berkeley.edu/>).

Modelling. The top-level of the model (Figure 1) consists of one 'Modal Model' actor called 'TwoStateModel' with output ports for y_1 , y_2 and state changes (the icon for the modal model hides the port for y_2). A modal model implements a finite state machine whose states (or modes) can have refinements, i.e. submodels that execute when the state machine is in the corresponding state. Therefore this actor type is clearly well suited for the Two State Model.

The modal model actor has connections to various sinks: a timed plotter for plotting the output signal y_1 , a periodic sampler (period 5) with a display to show the final value for y_1 on screen and an expression writer with a clock (triggered by the output port for state changes) for writing the time values of the state events to a file. With these sinks all necessary information for accomplishment of the tasks is available.

A 'director' block defines the model of computation. In the top-level a continuous director governs the execution. It implements a differential equation solver, however as there is no differential equation at the top-level

just the continuous directors in the refinements of the states control its step size.

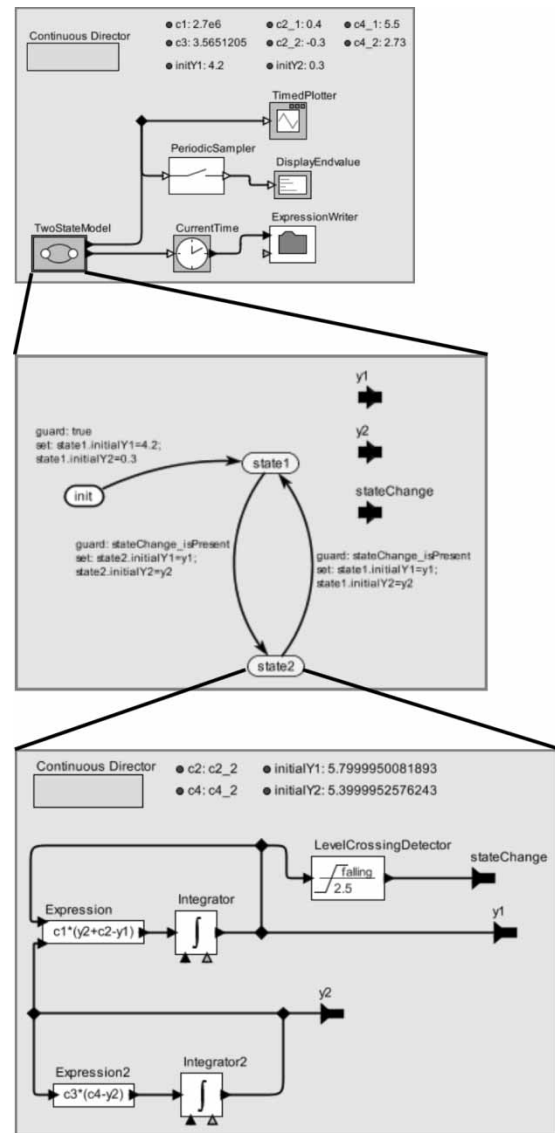


Figure 1. Three hierarchical levels of the model: The top level with the modal model actor 'TwoStateModel' inside of which there is a finite state machine with 'init' state for initialization and the two states of the system. Each state has a refinement. The refinement for state 2 is shown here. Note the 'LevelCrossingDetector' actor which triggers the event of hitting the pin.

Inside the modal model actor ‘TwoStateModel’ is a finite state machine (therefore it implicitly has an FSM director) with an initial state ‘init’ used for initializing the model with the corrector initial values for y_1 and y_2 and the two states ‘state1’ and ‘state2’. The port ‘stateChange’ triggers the transitions between the states, as it receives an output token each time y_1 crosses the specific thresholds. At the time of a state change the output ports for y_1 and y_2 give the new initial values of the state the model transits to.

The refinements of both state1 and state2 implement the differential equations. The block structure is similar to well-known block diagrams (as for example in Simulink) and has two integrator actors, one for y_1 and one for y_2 . Both are connected to the corresponding output ports. A ‘LevelCrossingDetector’ actor detects the crossing of the threshold (in rising direction in the case of state1 and in falling direction in state2). At a crossing it sends a token to the output port ‘stateChange’, activating the guard of the state transition. Both refinements have a continuous director and get the values of c_2 and c_4 from the corresponding parameters defined at top-level.

A-Task: Simulation Time Domain. The version of Ptolemy II used does not include a solver appropriate for stiff systems. The available solvers are explicit RK23 and explicit RK45. The simulation with RK45 was inefficient: Even with an error tolerance (for the solver and for the level crossing detectors) of 10^{-6} it took several hours (on a laptop with Intel Core 2 Duo T8300, 2GB of Ram and Windows Vista), because the solver had to choose very small step sizes. Figure 2 shows the plot for y_1 with error tolerance set to 10^{-10} .

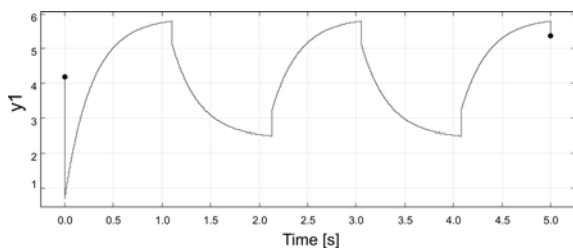


Figure 2. Plot of y_1 over time, all error tolerances set to 10^{-10}

B & C-Tasks: Event Times and Final Values. Table 1 shows the time values of the state transitions for all error tolerances set to 10^{-6} , 10^{-10} and 10^{-14} . With error tolerance 10^{-6} an additional sixth event takes place and the final value of y_1 is completely wrong. With more stringent error tolerances all events and the final value are detected nearly exactly.

| tol | 1.E-06 | 1.E-10 | 1.E-14 |
|-------|-----------------|-----------------|-----------------|
| t_1 | 1.1082628547607 | 1.1083061678763 | 1.1083061677712 |
| t_2 | 2.091871335981 | 2.1296853555014 | 2.1296853551551 |
| t_3 | 2.7319550555404 | 3.0541529075528 | 3.0541529069963 |
| t_4 | 3.2977498326525 | 4.0755320954016 | 4.0755320943802 |
| t_5 | 4.1092310397903 | 4.9999996474117 | 4.9999996462213 |
| t_6 | 4.9718927771635 | | |
| y_f | 3.5041594536956 | 5.3701798572355 | 5.3693128433754 |

Table 1. Time values of the state events with error tolerances set to 1.E-6, 1.E-10 and 1.E-14, final values y_f .

D-Task: Frequent Events. With new parameter values, the system exhibits high frequent oscillatory behaviour after the first state transition from state1 to state2.

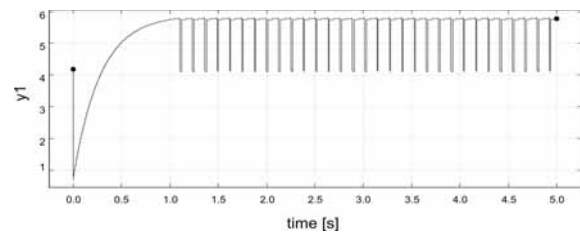


Figure 3. Plot of y_1 over time for the parameter values of task d (error tolerances set to 10^{-11}).

| | |
|----------------------------|----------------------------|
| $t_1 = 1.1083061677768$ | $t_{61} = 4.9230401079777$ |
| $t_2 = 1.1217299678991$ | $t_{62} = 4.9364639081008$ |
| $t_{60} = 4.8093061100968$ | $y_f = 5.7804025201735$ |

Table 1. First two and last three time values of the state events and final state, error tolerances; Task d.

Altogether there are 62 state events. Figure 3 shows the time plot of y_1 , and Table 2 gives the time values of the first and last discontinuities as well as the final value for an error tolerance of 10^{-11} .

Resume. The philosophy of the Ptolemy Project is appealing, as it does not express heterogeneous models in terms of one general model of computation. Instead of that it uses an abstract semantics which every actor has to fulfil, and a director block in each actor implements the concrete rules of execution. This allows a rigorous coupling in hybrid models.

Furthermore Ptolemy II is an open source software and freely available. Therefore it is possible to study the internals of the simulator and to do research on hybrid systems without expensive software. Still missing are advanced ODE or DAE solvers (2011).

Submitted: August 2011

Revised: October 2011

Accepted: February 1, 2012

A Discrete-event Modeling of ARGESIM Benchmark C20 'Complex Production System' using Flexsim Simulation Software

Eike Schulz^{*}, Josef Paul, Sebastian Schreiber, Alexander Fay

Helmut-Schmidt-University / University of the Federal Armed Forces Hamburg, Institute of Automation Technology, Holstenhofweg 85, D-22043 Hamburg, Germany; eike.schulz@hsu-hh.de

Abstract. ARGESIM proposes a new benchmark description ARGESIM C20 'Complex Production System'. This article describes a first implementation of this benchmark by use of the simulation software *Flexsim*. The simulation model is controlled by a heterarchical control algorithm implemented in *Flexsim*, which covers routing and scheduling and is based on reservations. The results are presented according to the requirements of the benchmark description.

Introduction

In [1], Schreiber and Fay propose an advanced reference system for the benchmarking of manufacturing control systems. As a first implementation a simulation model of the presented manufacturing system has been modeled by use of the simulation software *Flexsim* [2]. In addition, a heterarchical control algorithm has been developed that can cope with the challenges of the benchmark description.

This algorithm has also been implemented in the *Flexsim* model. Furthermore, interfaces have been implemented which allow the subsequent integration of other internal and external control algorithms into the model. Another implemented part of the simulation model are routines for the generation of output data for external analysis of each simulation run.

In this article, a brief introduction into the functionality and capabilities of *Flexsim* is given in Section 1. The designed simulation model is described in Section 2 the developed control algorithm in Section 3. Results of the validation of model and algorithm are presented in Section 4.

1 *Flexsim* Simulation Software

Flexsim is an object-oriented, discrete-event simulation software distributed by Flexsim Software Products Inc. (Orem, Utah, USA). Although declared as general purpose software, it is especially suitable for production and intra-logistic tasks. It offers a graphical user interface with a 2D or 3D view of the model and several standard library objects that can be included into the model via drag & drop. In addition to the 2D and 3D view, *Flexsim* offers a tree-view which shows all objects, tools and variables which the model consists of as a hierarchically structured tree. By creating a separate tree node for each component of the model, *Flexsim* assures its object-orientation.

All library objects can be edited by the user and complex logic can be placed in their event-triggers by using *Flexscript*, an integrated programming language. Logic can also be added by use of C++, but this code needs to be compiled before the model can be run. In addition, it is possible to bind DLLs to add user-specific functionality.

Generally *Flexsim* can process numerical and string values. These can be stored in global variables or tables or locally as subnodes of individual objects (so called 'labels'). Communication inside the model is possible by sending messages (which can contain up to three numeric values) from one object to another or by directly referencing another node. An interface to Excel is used for import and export of Excel/CSV files. For further information on the capabilities of *Flexsim* see [3].

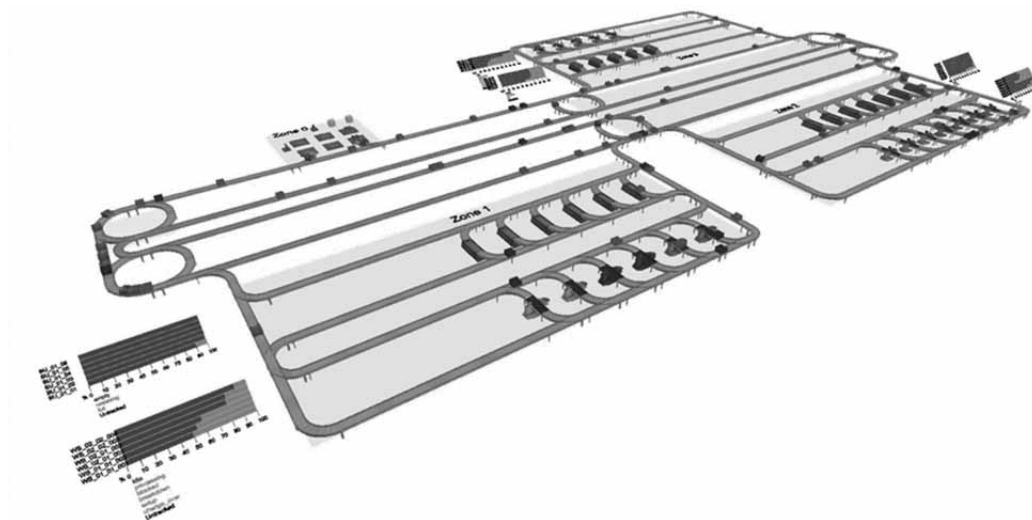


Figure 1. 3D view of the simulation model.

2 Reference System Model

A special demand imposed on this implementation of the ‘Complex Production System’ benchmark had been to provide the possibility to control it by different control algorithms, both control algorithms implemented in *Flexsim* (by means of the above-mentioned programming languages) and external ones, which communicate with the simulation model via OPC interfaces.

To fulfill the requirements regarding flexibility and adaptability, all parts of the model have been based on standard library objects: work stations (WS) on the standard *processor* object, the conveyors on the standard *conveyor* object, and the FIFO-buffers (BU) on the standard *queue* object. Furthermore, in each instance of an object class (e.g. WS or conveyor) the corresponding triggers have the identical code. The only individual difference is the instance’s name. This name follows a defined nomenclature which for example in case of the WS looks like ‘WS_xx_yy_zzz’, where the abbreviation ‘WS’ characterizes the object class; the first two numbers ‘xx’ show the type of WS (i.e. I-V); the two numbers ‘yy’ show the generation of the WS; the last three numbers ‘zzz’ are an instance specific number for this specific type of WS.

Based on this nomenclature, all other individual parameters, e.g. tool capacity and operation capabilities, are stored in global tables and variables and are retrieved on reset of the model. During runtime they are stored locally in labels on each object. By this means, the user can on the one hand easily change parameters

for several objects by just editing one global table, and on the other hand quickly create new objects by just cloning an existing one, editing its name and adding a line for the new parameters in the corresponding global table. This enhances flexibility and adaptability of the simulation model. The WS setup and the layout of the conveyors follow the description in [1]. A picture of the production system is shown in Figure 1.

Although *Flexsim* is not designed to make a separation between system’s behaviour and control system itself, such a separation is advisable to run different control algorithms in combination with the simulation model of the production system. Therefore, basic functions (i.e. import of order books and retrieval of object capabilities) have been separated from functions that are used specifically by some control algorithms (e.g. machine and buffer scheduling and routing decisions).

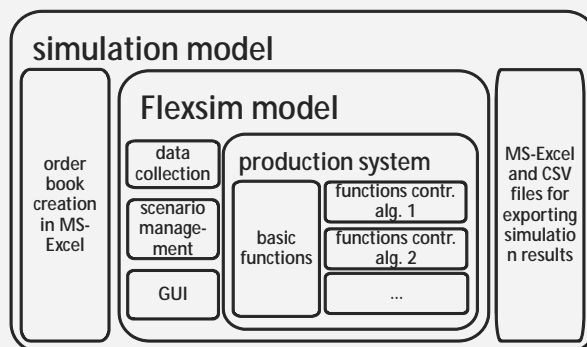


Figure 2: Extent of the simulation model.

The code of the latter kind of functions is encapsulated within “case select” blocks, where a global variable indicates the currently active control algorithm. During runtime, only the code of the corresponding “case” is executed. To enable the simulation model to be used with other control algorithms, only a new “case” needs to be defined and implemented.

Another part of the model are the routines for importing order books and exporting validation data. To make later comparison to other implementations of the ARGESIM C20 Benchmark possible, the order books as well as the validation data are stored as Excel/CSV files. Before the start of a simulation, the selected order book is imported into a global table. During the simulation run a variety of data is collected (e.g. state changes of all WS and buffers, entry and exit times of all products into WS and buffers). At the end of each simulation run, this data is aggregated and exported into Excel files which already include formulas to calculate the criteria proposed in [1].

There are two main reasons for not using the *Flexsim* internal analysis tools. The first is that Excel/CSV files can easier be accessed and retraced by others, the second is the possibility to use the add-in RExcel for further data analysis. The general structure of the simulation model as described in this paragraph is shown in Figure 2.

3 Heterarchical Control Algorithm

The first control algorithm that has been implemented for the ARGESIM C2E benchmark is of heterarchical nature and has been implemented directly in *Flexsim*. The control algorithm has to ensure that all items of the order book are produced according to their operation sequences and that all production constraints are met. It also has to manage tool replacements of WS type II and the operation changes of WS type V. Therefore the key elements of the control algorithm are its abilities regarding the scheduling of WS and buffers and the routing of the items between them. The scheduling is done by means of reservation routines for WS and BU slots. These routines are the most complex ones in the model and are the core of the algorithm.

As a pre-condition for the routing, all objects that can be selected as a target (i.e. WS, buffers, sink) and all split junctions have a unique ID that is assigned on reset of the model. The split junctions, which are called “decision points” (DPs) in the model, are the points

where the routing decisions take place. When an item with a target ID enters a DP, it is checked which of the exit ports leads to the shortest path to the target. Because of the fixed layout of the transport system, the shortest paths can be calculated already at reset of the model. This is done by use of the ‘Floyd-Wharshall all-pairs shortest-path’ algorithm.

The results are stored in global tables and can be accessed by each DP. For Scenarios 10 to 12, shortest-path calculation can be repeated after the breakdown or repair of a conveyor to update the distance-table and to allow rerouting of elements. In addition to the search for the shortest path, there are some fail-safe procedures implemented in the DPs. These prevent system to get into a block or deadlock state.

Every time an item enters a DP, a procedure called *Check_Item* is executed before the routing decision starts (see Figure 3). In this procedure it is first checked whether the item has a reservation or not. If not, the reservation sequence is initialized, and the check is completed. If the item already has a reservation, it is checked if there are any production constraints connected to this reservation and if they are still met. It is also checked if the item can still arrive on schedule at the designated WS. If there are any problems recognized with the existing reservation, the necessary steps are taken to cancel it and initialize the reservation sequence again. The *Check_Item* procedure is designed as a user defined function (called ‘user command’ in *Flexsim*) that can be automatically executed from every DP.

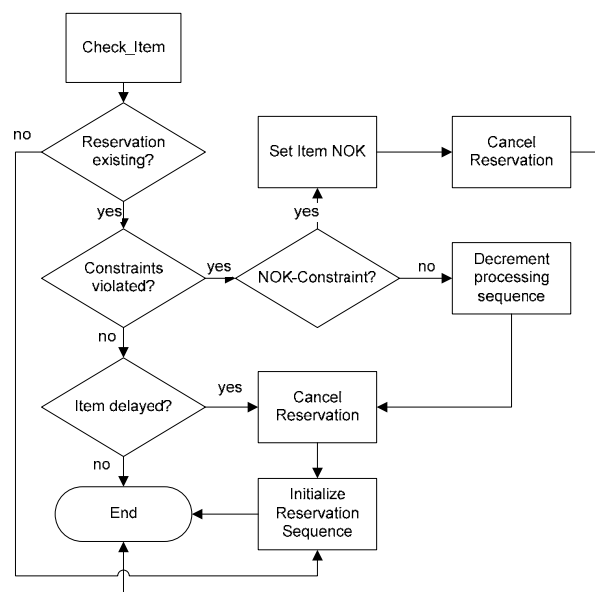


Figure 3. Check_item procedure at each decision point.

| Sc. | Task A makespan[h] | | Task B throughput time [s] | | Task C | | | | Task D No of NOK-1 | | Task D No of NOK-2 | |
|-----|-----------------------|------|-------------------------------|--------|------------------|-------|--------------------------|--------|-----------------------|-------|-----------------------|-------|
| | Median | IQR | Median | IQR | No of tardy jobs | | total weighted tard. [h] | | Median | IQR | Median | IQR |
| | | | | | Median | IQR | Median | IQR | | | | |
| 1 | 21.38 | 0.00 | 914.4 | 445.8 | 626.5 | 0.0 | 25.03 | 0.00 | 13.0 | 00.00 | 18.0 | 0.00 |
| 2 | | | | | 669.5 | 0.0 | 27.99 | 0.00 | | | | |
| 3 | | | | | 745.5 | 0.0 | 33.69 | 0.00 | | | | |
| 4 | 21.47 | 0.16 | 982.1 | 511.0 | 953.5 | 28.3 | 76.09 | 5.47 | 25.5 | 13.00 | 30.0 | 8.50 |
| 5 | | | | | 1019.5 | 46.5 | 80.05 | 5.73 | | | | |
| 6 | | | | | 1086.5 | 33.3 | 92.82 | 10.42 | | | | |
| 7 | 21.90 | 0.25 | 1214.0 | 919.4 | 2425.0 | 98.8 | 478.44 | 41.60 | 151.5 | 23.75 | 174.5 | 40.25 |
| 8 | | | | | 2516.0 | 66.0 | 508.58 | 58.86 | | | | |
| 9 | | | | | 2584.5 | 81.8 | 536.99 | 60.21 | | | | |
| 10 | 21.88 | 0.13 | 1260.0 | 1005.3 | 2744.5 | 135.8 | 647.85 | 125.91 | 178.5 | 38.25 | 221.0 | 46.25 |
| 11 | | | | | 2780.0 | 130 | 637.26 | 141.89 | | | | |
| 12 | | | | | 2801.5 | 134.8 | 669.69 | 100.43 | | | | |

Table 1: Results of tasks A-D for 30 (90) independent simulation runs of one order book of 5,000 items (OB_1000).

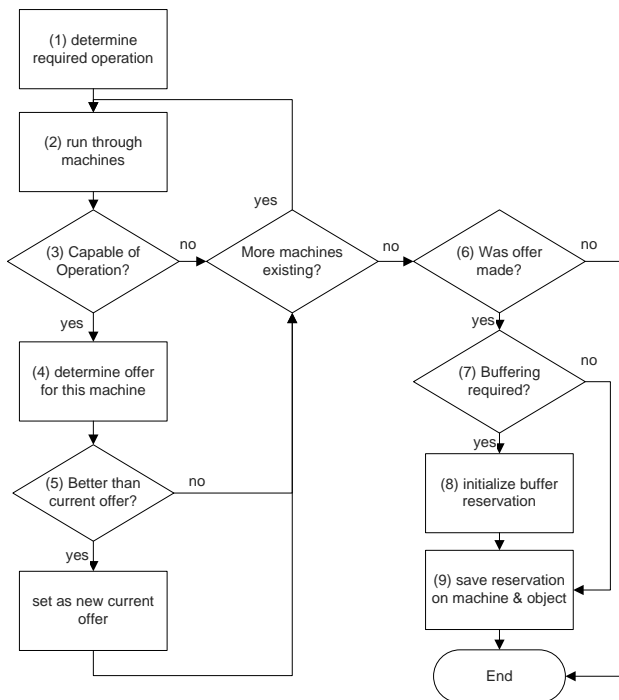


Figure 4. Reservation process.

The reservation process itself works as shown in Figure 4. It is distributed among the three user defined functions *Reserve_WS*, *Reserve_BU* (for buffer reservations) and *Cancel_Res* which are called directly from the *Check_Item* procedure. At the beginning of the process, the required process step is determined and all machines are considered to find the best possible offer for this item and operation. In this algorithm, the offer with the earliest possible start is selected. No optimization rules like “earliest due date” or “total weighted tardiness” are used.

Each item tries individually to find the best offer for itself. If an offer can be made it is checked if the item can be sent directly to the WS or if buffering is required. In the latter case, the function *Reserve_BU* is started, and in case a buffer place was found, the reservations are stored on the item and on the corresponding buffer or WS. The ID of the buffer or WS that has been reserved is then used as target ID for this item.

Although an even more heterarchical way of decision making would have been preferable, the workaround with the DPs had to be made because flowitems cannot start logic on their own in *Flexsim* but only activate triggers in other fixed resources.

Other important parts of the control algorithm are the user defined functions which are responsible for the reservation of time slots for tool replacement and operation changes. These functions are called directly from the workstations when the tool abrasion limit is undercut or a certain number of operation change requests is reached.

By making reservations for these changeover times, it is possible to already plan normal item reservations after a tool or operation change. With this no reservation has to be cancelled because of tool abrasion or a required operation change.

4 Results

According to [1] results are presented for the four criteria makespan (Task A), throughput time (Task B), tardiness (Task C) and number of violated constraints (Task D) by use of the statistics median and interquartile range (IQR). After validation of model and algorithm and finalizing all parameters, simulation runs with three order books were made.

The order books OB1000, OB1001 and OB1002, each consisting of a set of 5,000 orders, were tested with 30 independent simulation runs per order book. Results for OB1000 are presented in Table 1. Results for the other two order books can be found in Appendix A.

Because the control algorithm does not account any due dates and does not make any reservations before an order's release date, it works exactly the same in all three operational scenarios. Due to this the results (except for tardiness) could be aggregated for each of the four manufacturing scenarios. The median and IQR for tasks A, B and D, are hence based on 90 independent values.

For PSc#1 (deterministic, scenarios 1-3) the control algorithm provides deterministic results, as expected. Only the throughput time has a positive IQR because all product types are mapped in one value. The results for all Tasks show the biggest leap from PSc#2 (some disruptions, scenarios 4-6) to PSc#3 (many disruptions, scenarios 7-9). The additional disruptions of the transport system in PSc#4 (scenarios 10-12) have in contrast rather moderate impact on the values.

References

- [1] S. Schreiber, A. Fay. *ARGESIM Benchmark C20 'Complex Production System' – Definition and Call*. SNE Simulation Notes Europe, vol. 21 (3-4), 2011. doi: 10.11128/sne.21.bn20.10095
- [2] *Flexsim*. <http://www.flexsim.com/products/flexsim>.
- [3] M. Beaverstock, A. Greenwood, E. Lavery, W. Nordgren. *Applying Simulation with Flexsim*. Orem, Utah, USA: Flexsim Simulation Software, 2010.

Remarks. This work is based on the first author's master thesis "*Implementation and validation of a benchmark for a production system by use of a heterarchical control architecture*" (in German), written at and accepted by the Helmut-Schmidt-University, Institute of Automation Techn. 2011.

Submitted: February 10, 2012

Accepted: March 15, 2012

Appendix A - Further Results

| Sc. | Task A makespan[h] | | Task B throughput time [s] | | Task C | | | | Task D No of NOK-1 | | Task D No of NOK-2 | |
|-----|-----------------------|------|-------------------------------|-------|------------------|-------|--------------------------|--------|-----------------------|-------|-----------------------|-------|
| | Median | IQR | Median | IQR | No of tardy jobs | | total weighted tard. [h] | | Median | IQR | Median | IQR |
| | | | | | Median | IQR | Median | IQR | | | | |
| 1 | 21.20 | 0.00 | 942.4 | 468.9 | 725 | 0.0 | 31.18 | 0.00 | 35.0 | 0.00 | 18.0 | 0.00 |
| 2 | | | | | 785 | 0.0 | 35.72 | 0.00 | | | | |
| 3 | | | | | 863 | 0.0 | 44.20 | 0.00 | | | | |
| 4 | 21.23 | 0.14 | 993.5 | 524.0 | 1069.5 | 45.3 | 88.41 | 11.28 | 39.0 | 13.25 | 28.0 | 9.25 |
| 5 | | | | | 1115.5 | 43.3 | 96.83 | 8.39 | | | | |
| 6 | | | | | 1210.5 | 34.3 | 104.19 | 15.75 | | | | |
| 7 | 22.19 | 0.64 | 1204.0 | 925.3 | 2532.5 | 52.0 | 715.56 | 115.38 | 163.5 | 17.25 | 180.5 | 20.75 |
| 8 | | | | | 2570.5 | 66.0 | 703.60 | 83.58 | | | | |
| 9 | | | | | 2624.5 | 44.0 | 731.19 | 135.36 | | | | |
| 10 | 22.49 | 0.99 | 1247.0 | 993.9 | 2721.0 | 122.3 | 876.86 | 203.88 | 178.0 | 18.50 | 205.0 | 34.5 |
| 11 | | | | | 2779.5 | 88.5 | 865.58 | 162.33 | | | | |
| 12 | | | | | 2859.5 | 121.3 | 983.91 | 199.74 | | | | |

Table 2: Results of tasks A-D for 30 (90) independent simulation runs of one order book of 5,000 items (OB_1001).

| Sc. | Task A makespan[h] | | Task B throughput time [s] | | Task C | | | | Task D No of NOK-1 | | Task D No of NOK-2 | |
|-----|-----------------------|------|-------------------------------|--------|------------------|-------|--------------------------|--------|-----------------------|------|-----------------------|-------|
| | Median | IQR | Median | IQR | No of tardy jobs | | total weighted tard. [h] | | Median | IQR | Median | IQR |
| | | | | | Median | IQR | Median | IQR | | | | |
| 1 | 20.98 | 0.00 | 908.6 | 429.4 | 676.0 | 0.00 | 30.04 | 0.00 | 21.0 | 0.0 | 14.0 | 0.00 |
| 2 | | | | | 721.0 | 0.00 | 32.55 | 0.00 | | | | |
| 3 | | | | | 801.0 | 0.00 | 38.61 | 0.00 | | | | |
| 4 | 20.91 | 0.12 | 990.2 | 510.0 | 1054.5 | 44.0 | 80.65 | 6.94 | 30.0 | 15.5 | 32.5 | 7.50 |
| 5 | | | | | 1106.5 | 28.8 | 87.25 | 5.99 | | | | |
| 6 | | | | | 1184.0 | 43.5 | 96.53 | 10.78 | | | | |
| 7 | 21.24 | 0.25 | 1255.0 | 986.6 | 2768.5 | 115.3 | 631.56 | 106.99 | 184.5 | 25.0 | 248.5 | 49.25 |
| 8 | | | | | 2839.0 | 91.8 | 648.38 | 86.52 | | | | |
| 9 | | | | | 2883.0 | 109.3 | 669.96 | 141.72 | | | | |
| 10 | 21.23 | 0.32 | 1306.0 | 1085.2 | 3064.5 | 198.3 | 829.57 | 222.02 | 221.0 | 50.0 | 295.0 | 68.25 |
| 11 | | | | | 3115. | 277.3 | 849.98 | 437.49 | | | | |
| 12 | | | | | 3197.0 | 176.8 | 1065.43 | 421.21 | | | | |

Table 3: Results of tasks A-D for 30 (90) independent simulation runs of one order book of 5,000 items (OB_1002).

SNE Simulation News

EUROSIM Data and Quick Info



EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

www.eurosim2013.info

Contents

| | |
|---|-------|
| Info EUROSIM | 2 |
| Info EUROSIM Societies | 3 - 7 |
| Info ASIM, CROSSIM | 3 |
| Info CSSS, HSS, DBSS, FRANCOSIM | 4 |
| Info ISCS, PSCS, SIMS, SLOSIM | 5 |
| Info UKSIM, LSS, CAE-SMSG, ROMSIM | 6 |
| Info RNSS, LIOPHANT Info SNE | 7 |
| News EUROSIM, ASIM, SLOSIM | 9 |
| Introduction & News RNSS – Russian Sim. Society | 10 |

Simulation Notes Europe SNE is the official membership journal of EUROSIM and distributed / available to members of the EUROSIM Societies as part of the membership benefits. **SNE** is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With Online **SNE** the publisher ARGESIM follows the Open Access strategy for basic **SNE** contributions. Since 2011 Online **SNE** contributions are identified by DOI 10.11128/sne.xx.nnnnn. for better web availability and indexing.

Print **SNE**, high-resolution Online **SNE**, and additional **SNE** contributions are available for subscription via membership in a EUROSIM society.

This *EUROSIM Data & Quick Info* compiles data from EUROSIM societies and groups: addresses, weblinks, officers of societies with function and email, to be published regularly in **SNE** issues.

SNE Reports Editorial Board

EUROSIM Khalid Al-Begain, kbegain@glam.ac.uk
Borut Zupančič, borut.zupancic@fe.uni-lj.si
Felix Breitenecker, Felix.Breitenecker@tuwien.ac.at
ASIM Thorsten Pawletta, pawel@mb.hs-wismar.de
CROSSIM Vesna Dušak, vdusak@foi.hr
CSSS Mikuláš Alexík, alexik@frtk.utc.sk
DBSS A. Heemink, a.w.heemink@its.tudelft.nl
FRANCOSIM Karim Djouani, djouani@u-pec.fr
HSS András Jávör, javor@eik.bme.hu
ISCS M. Savastano, mario.savastano@unina.it
PSCS Zenon Sosnowski, zenon@ii.pb.bialystok.pl
SIMS Esko Juuso, esko.juuso@oulu.fi
SLOSIM Rihard Karba, rihard.karba@fe.uni-lj.si
UKSIM Richard Zobel, r.zobel@ntlworld.com
CAE-SMSG Emilio Jiminez, emilio.jiminez@unirioja.es
LSS Yuri Merkurjev, merkur@itl.rtu.lv
ROMSIM Florin Stanciulescu, sflorin@ici.ro
RNSS Y. Senichenkov, sneyb@dcn.infos.ru
LIOPHANT F. Longo, f.longo@unicat.it

SNE Editorial Office /ARGESIM

→ www.sne-journal.org, www.eurosim.info

✉ office@sne-journal.org

Felix Breitenecker, eic@sne-journal.org
Anna Mathe, Anna.Mathe@tuwien.ac.at, office@sne-journal.org
Nikolas Popper, Niki.Popper@drahtwarenhandlung.at

If you have any information, announcement, etc. you want to see published, please contact a member of the editorial board in your country or the editorial office.



EUROSIM
Federation of European
Simulation Societies

General Information. EUROSIM, the Federation of European Simulation Societies, was set up in 1989. The purpose of EUROSIM is to provide a European forum for regional and national simulation societies to promote the advancement of modelling and simulation in industry, research, and development.

→ www.eurosim.info

Member Societies. EUROSIM members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation. At present EUROSIM has thirteen full members and three observer member:

| | |
|------------|---|
| ASIM | Arbeitsgemeinschaft Simulation <i>Austria, Germany, Switzerland</i> |
| CEA-SMSG | Spanish Modelling and Simulation Group <i>Spain</i> |
| CROSSIM | Croatian Society for Simulation Modeling <i>Croatia</i> |
| CSSS | Czech and Slovak Simulation Society <i>Czech Republic, Slovak Republic</i> |
| DBSS | Dutch Benelux Simulation Society <i>Belgium, Netherlands</i> |
| FRANCO-SIM | Société Francophone de Simulation <i>Belgium, France</i> |
| HSS | Hungarian Simulation Society <i>Hungary</i> |
| ISCS | Italian Society for Computer Simulation <i>Italy</i> |
| LSS | Latvian Simulation Society <i>Latvia</i> |
| PSCS | Polish Society for Computer Simulation <i>Poland</i> |
| SIMS | Simulation Society of Scandinavia <i>Denmark, Finland, Norway, Sweden</i> |
| SLOSIM | Slovenian Simulation Society <i>Slovenia</i> |
| UKSIM | United Kingdom Simulation Society <i>UK, Ireland</i> |
| ROMSIM | Romanian Society for Modelling and Simulation, <i>Romania, Observer Member</i> |
| RNSS | Russian National Simulation Society <i>Russian Federation, Observer Member</i> |
| LIOPHANT | LIOPHANT Simulation Club <i>Italy & Interbational, Observer Member</i> |

EUROSIM Board / Officers. EUROSIM is governed by a board consisting of one representative of each member society, president and past president, and representatives for SNE Simulation notes Europe. The President is nominated by the society organising the next EUROSIM Congress. Secretary and Treasurer are elected out of members of the Board.

| | |
|----------------|---|
| President | Khalid Al.Begain <i>kbegain@glam.ac.uk</i> |
| Past president | Mikuláš Alexík (CSSS), <i>alexik@frtk.fri.utc.sk</i> |
| Secretary | Borut Zupančič (SLOSIM) <i>borut.zupancic@fe.uni-lj.si</i> |
| Treasurer | Felix Breitenecker (ASIM) <i>felix.breitenecker@tuwien.ac.at</i> |
| SNE Repres. | Felix Breitenecker <i>felix.breitenecker@tuwien.ac.at</i> |

SNE – Simulation Notes Europe. SNE is a scientific journal with reviewed contributions as well as a membership newsletter for EUROSIM with information from the societies in the *News Section*. EUROSIM societies are offered to distribute to their members the journal SNE as official membership journal. SNE Publishers are EUROSIM, ARGESIM and ASIM.

| | |
|-----------------|--|
| Editor-in-chief | Felix Breitenecker <i>felix.breitenecker@tuwien.ac.at</i> |
|-----------------|--|

→ www.sne-journal.org,

✉ office@sne-journal.org

EUROSIM Congress. EUROSIM is running the triennial conference series EUROSIM Congress. The congress is organised by one of the EUROSIM societies.

EUROSIM 2013 will be organised by UKSIM in Cardiff, Wales, UK, September 10-13, 2013.

Chairs / Team EUROSIM 2010

Khalid Al.Begain, *kbegain@glam.ac.uk*
Richard Zobel, *r.zobel@ntlworld.com*
David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Alessandra Orsoni, *a.orsoni@kingston.ac.uk*
Richard Cant, *richard.cant@ntu.ac.uk*

→ www.eurosim2013.info



ASIM German Simulation Society Arbeitsgemeinschaft Simulation

ASIM (Arbeitsgemeinschaft Simulation) is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria. ASIM was founded in 1981 and has now about 700 individual members, and 30 institutional or industrial members. Furthermore, ASIM counts about 300 affiliated members.

→ www.asim-gi.org with members' area

✉ info@asim-gi.org, admin@asim-gi.org

✉ ASIM – Inst. f. Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

ASIM Officers

| | |
|--------------------------------------|---|
| President | Felix Breitenecker felix.breitenecker@tuwien.ac.at |
| Vice presidents | Sigrid Wenzel, s.wenzel@uni-kassel.de T. Pawletta, pawel@mb.hs-wismar.de |
| Secretary | Anna Mathe, anna.mathe@tuwien.ac.at |
| Treasurer | I. Bausch-Gall, Ingrid@Bausch-Gall.de |
| Membership affairs | S. Wenzel, s.wenzel@uni-kassel.de W. Maurer, werner.maurer@zhwin.ch I. Bausch-Gall, Ingrid@Bausch-Gall.de F. Breitenecker, felix.breitenecker@tuwien.ac.at |
| Universities / Research Inst. | S. Wenzel, s.wenzel@uni-kassel.de W. Wiechert, W.Wiechert@fz-juelich.de J. Haase, Joachim.Haase@eas.iis.fraunhofer.de Katharina Nöh, k.noeh@fz-juelich.de |
| Industry | S. Wenzel, s.wenzel@uni-kassel.de K. Panreck, Klaus.Panreck@hella.com |
| Conferences | Klaus Panreck Klaus.Panreck@hella.com A. Gnauck, albrecht.gnauck@tu-cottbus.de |
| Publications | Th. Pawletta, pawel@mb.hs-wismar.de Christina Deatcu, christina.deatcu@hs-wismar.de F. Breitenecker, felix.breitenecker@tuwien.ac.at |
| Repr. EuroSIM | F. Breitenecker, felix.breitenecker@tuwien.ac.at N. Popper, niki.popper@drahtwarenhandlung.at |
| Education / Teaching | Ch. Deatcu, christina.deatcu@hs-wismar.de N. Popper, niki.popper@drahtwarenhandlung.at Katharina Nöh, k.noeh@fz-juelich.de |
| International Affairs | H. Szczerbicka, hsz@sim.uni-hannover.de O. Rose, Oliver.Rose@tu-dresden.de |
| Editorial Board SNE | T. Pawletta, pawel@mb.hs-wismar.de Ch. Deatcu, christina.deatcu@hs-wismar.de |
| Web EuroSIM | Anna Mathe, anna.mathe@tuwien.ac.at |

Last data update March 2011

ASIM Working Committee. ASIM, part of GI - Gesellschaft für Informatik, is organised in Working Committees, dealing with applications and comprehensive subjects in modelling and simulation:

ASIM Working Committee

| | |
|-------------|--|
| GMMS | Methods in Modelling and Simulation Th. Pawletta, pawel@mb.hs-wismar.de |
| SUG | Simulation in Environmental Systems Wittmann, wittmann@informatik.uni-hamburg.de |
| STS | Simulation of Technical Systems H.T.Mammen, Heinz-Theo.Mammen@hella.com |
| SPL | Simulation in Production and Logistics Sigrid Wenzel, s.wenzel@uni-kassel.de |
| EDU | Simulation in Education/Education in Simulation N. Popper, niki.popper@drahtwarenhandlung.at |

Working Groups for Simulation in Business Administration, in Traffic Systems, for Standardisation, for Validation, etc.

CROSSIM – Croatian Society for Simulation Modelling

CROSSIM-Croatian Society for Simulation Modelling was founded in 1992 as a non-profit society with the goal to promote knowledge and use of simulation methods and techniques and development of education. CROSSIM is a full member of EUROSIM since 1997.

→ www.eurosim.info

✉ vdusak@foi.hr

✉ CROSSIM / Vesna Dušak
Faculty of Organization and Informatics Varaždin, University of Zagreb
Pavlinska 2, HR-42000 Varaždin, Croatia

CROSSIM Officers

| | |
|--------------------------------|--|
| President | Jadranka Božikov, jbozikov@snz.hr |
| Vice president | Vesna Dušak, vdusak@foi.hr |
| Secretary | Vesna Bosilj-Vukšić, vbosilj@efzg.hr |
| Executive board members | Vlatko Čerić, vceric@efzg.hr Tarzan Legović, legovic@irb.hr |
| Repr. EuroSIM | Jadranka Božikov, jbozikov@snz.hr |
| Edit. Board SNE | Vesna Dušak, vdusak@foi.hr |
| Web EuroSIM | Jadranka Božikov, jbozikov@snz.hr |

Last data update March 2011



CSSS – Czech and Slovak Simulation Society

CSSS -The *Czech and Slovak Simulation Society* has about 150 members working in Czech and Slovak national scientific and technical societies (*Czech Society for Applied Cybernetics and Informatics, Slovak Society for Applied Cybernetics and Informatics*). The main objectives of the society are: development of education and training in the field of modelling and simulation, organising professional workshops and conferences, disseminating information about modelling and simulation activities in Europe. Since 1992, CSSS is full member of EUROSIM.

→ www.fit.vutbr.cz/CSSS

✉ snorek@fel.cvut.cz

✉ CSSS / Miroslav Šnorek, CTU Prague
FEE, Dept. Computer Science and Engineering,
Karlovo nám. 13, 121 35 Praha 2, Czech Republic

CSSS Officers

| | |
|-------------------------|--|
| President | Miroslav Šnorek, snorek@fel.cvut.cz |
| Vice president | Mikuláš Alexik, alexik@frtk.fri.utc.sk |
| Treasurer | Evžen Kindler, ekindler@centrum.cz |
| Scientific Secr. | A. Kavička, Antonin.Kavicka@upce.cz |
| Repr. EUROSIM | Miroslav Šnorek, snorek@fel.cvut.cz |
| Deputy | Mikuláš Alexik, alexik@frtk.fri.utc.sk |
| Edit. Board SNE | Mikuláš Alexik, alexik@frtk.fri.utc.sk |
| Web EUROSIM | Petr Peringer, peringer@fit.vutbr.cz |

Last data update December 2009

FRANCOSIM – Société Francophone de Simulation

FRANCOSIM was founded in 1991 and aims to the promotion of simulation and research, in industry and academic fields. Francosim operates two poles.

- Pole Modelling and simulation of discrete event systems. Pole Contact: *Henri Pierreval*, pierre-va@imfa.fr
- Pole Modelling and simulation of continuous systems. Pole Contact: *Yskandar Hamam*, y.hamam@esiee.fr

→ www.eurosim.info

✉ y.hamam@esiee.fr

✉ FRANCOSIM / Yskandar Hamam
Groupe ESIEE, Cité Descartes,
BP 99, 2 Bd. Blaise Pascal,
93162 Noisy le Grand CEDEX, France

FRANCOSIM Officers

| | |
|------------------------|---|
| President | Yskandar Hamam, y.hamam@esiee.fr |
| Treasurer | François Rocaries, f.rocaries@esiee.fr |
| Repr. EUROSIM | Yskandar Hamam, y.hamam@esiee.fr |
| Edit. Board SNE | Yskandar Hamam, y.hamam@esiee.fr |

Last data update April 2006

DBSS – Dutch Benelux Simulation Society

The Dutch Benelux Simulation Society (DBSS) was founded in July 1986 in order to create an organisation of simulation professionals within the Dutch language area. DBSS has actively promoted creation of similar organisations in other language areas. DBSS is a member of EUROSIM and works in close cooperation with its members and with affiliated societies.

→ www.eurosim.info

✉ a.w.heemink@its.tudelft.nl

✉ DBSS / A. W. Heemink
Delft University of Technology, ITS - twi,
Mekelweg 4, 2628 CD Delft, The Netherlands

DBSS Officers

| | |
|------------------------|--|
| President | A. Heemink, a.w.heemink@its.tudelft.nl |
| Vice president | W. Smit, smitnet@wxs.nl |
| Treasurer | W. Smit, smitnet@wxs.nl |
| Secretary | W. Smit, smitnet@wxs.nl |
| Repr. EUROSIM | A. Heemink, a.w.heemink@its.tudelft.nl |
| Deputy | W. Smit, smitnet@wxs.nl |
| Edit. Board SNE | A. Heemink, a.w.heemink@its.tudelft.nl |

Last data update April 2006

HSS – Hungarian Simulation Society

The Hungarian Member Society of EUROSIM was established in 1981 as an association promoting the exchange of information within the community of people involved in research, development, application and education of simulation in Hungary and also contributing to the enhancement of exchanging information between the Hungarian simulation community and the simulation communities abroad. HSS deals with the organization of lectures, exhibitions, demonstrations, and conferences.

→ www.eurosim.info

✉ javor@eik.bme.hu

✉ HSS / András Jávör,
Budapest Univ. of Technology and Economics,
Sztoczek u. 4, 1111 Budapest, Hungary

**HSS Officers**

| | |
|------------------------|--|
| President | András Jávör, javor@eik.bme.hu |
| Vice president | Gábor Szűcs, szucs@itm.bme.hu |
| Secretary | Ágnes Vigh, vigh@itm.bme.hu |
| Repr. EUROSIM | András Jávör, javor@eik.bme.hu |
| Deputy | Gábor Szűcs, szucs@itm.bme.hu |
| Edit. Board SNE | András Jávör, javor@eik.bme.hu |
| Web EUROSIM | Gábor Szűcs, szucs@itm.bme.hu |

Last data update March 2008

PSCS – Polish Society for Computer Simulation - update

PSCS was founded in 1993 in Warsaw. PSCS is a scientific, non-profit association of members from universities, research institutes and industry in Poland with common interests in variety of methods of computer simulations and its applications. At present PSCS counts 257 members.

→ www.ptsk.man.bialystok.pl✉ leon@ibib.waw.pl✉ PSCS / Leon Bobrowski, c/o IBIB PAN,
ul. Trojdena 4 (p.416), 02-109 Warszawa, Poland**PSCS Officers**

| | |
|---------------------------|---|
| President | Leon Bobrowski, leon@ibib.waw.pl |
| Vice president | Andrzej Grzyb, Tadeusz Nowicki |
| Treasurer | Z. Sosnowski, zenon@ii.pb.bialystok.pl |
| Secretary | Zdzisław Galkowski, Zdzislaw.Galkowski@simr.pw.edu.pl |
| Repr. EUROSIM | Leon Bobrowski, leon@ibib.waw.pl |
| Deputy | A.Chudzikiewicz, ach@it.pw.edu.pl |
| Edit. Board SNE | Z.Sosnowski, zenon@ii.pb.bialystok.pl |
| PSCS Board Members | R. Bogacz, Z. Strzyzakowski Andrzej Tylikowski |

Last data update March 2009

ISCS – Italian Society for Computer Simulation

The Italian Society for Computer Simulation (ISCS) is a scientific non-profit association of members from industry, university, education and several public and research institutions with common interest in all fields of computer simulation.

→ www.eurosim.info✉ Mario.savastano@uniina.it✉ ISCS / Mario Savastano,
c/o CNR - IRSIP,
Via Claudio 21, 80125 Napoli, Italy**ISCS Officers**

| | |
|------------------------|--|
| President | M. Savastano, mario.savastano@uniina.it |
| Vice president | F. Maceri, Franco.Maceri@uniroma2.it |
| Repr. EUROSIM | F. Maceri, Franco.Maceri@uniroma2.it |
| Edit. Board SNE | M. Savastano, mario.savastano@uniina.it |

Last data update April 2005

SIMS – Scandinavian Simulation Society

SIMS is the *Scandinavian Simulation Society* with members from the four Nordic countries Denmark, Finland, Norway and Sweden. The SIMS history goes back to 1959. SIMS practical matters are taken care of by the SIMS board consisting of two representatives from each Nordic country. Iceland will be represented by one board member.

SIMS Structure. SIMS is organised as federation of regional societies. There are FinSim (Finnish Simulation Forum), DKSIM (Dansk Simuleringsforening) and NFA (Norsk Forening for Automatisering).

→ www.scansims.org✉ esko.juuso@oulu.fi

✉ SIMS / Esko Juuso, Department of Process and Environmental Engineering, 90014 Univ.Oulu, Finland

SIMS Officers

| | |
|------------------------|---|
| President | Esko Juuso, esko.juuso@oulu.fi |
| Treasurer | Vadim Engelson, vadim.engelson@mathcore.com |
| Repr. EUROSIM | Esko Juuso, esko.juuso@oulu.fi |
| Edit. Board SNE | Esko Juuso, esko.juuso@oulu.fi |
| Web EUROSIM | Vadim Engelson |

Last data update December March 2011

SLOSIM – Slovenian Society for Simulation and Modelling

SLOSIM - Slovenian Society for Simulation and Modelling was established in 1994 and became the full member of EUROSIM in 1996. Currently it has 69



members from both slovenian universities, institutes, and industry. It promotes modelling and simulation approaches to problem solving in industrial as well as in academic environments by establishing communication and cooperation among corresponding teams.

→ www.slosim.si✉ slosim@fe.uni-lj.si✉ SLOSIM / Rihard Karba, Faculty of Electrical Engineering, University of Ljubljana,
Tržaška 25, 1000 Ljubljana, Slovenia



SLOSIM Officers

| | |
|------------------------|---|
| President | B. Zupančič, borut.zupancic@fe.uni-lj.si |
| Vice president | Leon Žlajpah, leon.zlajpah@ijs.si |
| Secretary | Vito Logar, vito.logar@fe.uni-lj.si |
| Treasurer | Milan Simčič, milan.simcic@fe.uni-lj.si |
| Repr. EUROSIM | B. Zupančič, borut.zupancic@fe.uni-lj.si |
| Deputy | Rihard Karba, rihard.karba@fe.uni-lj.si |
| Edit. Board SNE | Rihard Karba, rihard.karba@fe.uni-lj.si |
| Web EUROSIM | Vito Logar, vito.logar@fe.uni-lj.si |

Last data update March 2011

UKSIM – United Kingdom Simulation Society

UKSIM has more than 100 members throughout the UK from universities and industry. It is active in all areas of simulation and it holds a biennial conference as well as regular meetings and workshops.

→ www.uksim.org.uk

✉ david.al-dabass@ntu.ac.uk

- ✉ UKSIM / Prof. David Al-Dabass
Computing & Informatics,
Nottingham Trent University
Clifton lane, Nottingham, NG11 8NS
United Kingdom

UKSIM Officers

| | |
|----------------------------|--|
| President | David Al-Dabass, david.al-dabass@ntu.ac.uk |
| Secretary | A. Orsoni, A.Orsoni@kingston.ac.uk |
| Treasurer | B. Thompson, barry@bjtcon.ndo.co.uk |
| Membership chair | K. Al-Begain, kbegain@glam.ac.uk |
| Univ. liaison chair | R. Cheng, rchc@maths.soton.ac.uk |
| Repr. EUROSIM | Richard Zobel, r.zobel@ntlworld.com |
| Edit. Board SNE | Richard Zobel, r.zobel@ntlworld.com |

Last data update March 2009 (partially)

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control. In order to improve the efficiency and to deep into the different fields of automation, the association is divided into thematic groups, one of them is named 'Modelling and Simulation', constituting the group.

→ www.cea-ifac.es/wwwgrupos/simulacion

→ simulacion@cea-ifac.es

- ✉ CEA-SMSG / María Jesús de la Fuente,
System Engineering and Automatic Control department,
University of Valladolid,
Real de Burgos s/n., 47011 Valladolid, SPAIN

CAE - SMSG Officers

| | |
|------------------------|--|
| President | María J. la Fuente, maria@autom.uva.es |
| Repr. EUROSIM | Emilio Jimenez, emilio.jimenez@unirioja.es |
| Edit. Board SNE | Emilio Jimenez, emilio.jimenez@unirioja.es |

Last data update March 2009

LSS – Latvian Simulation Society

The Latvian Simulation Society (LSS) has been founded in 1990 as the first professional simulation organisation in the field of Modelling and simulation in the post-Soviet area. Its members represent the main simulation centres in Latvia, including both academic and industrial sectors.

→ briedis.itl.rtu.lv/imb/

✉ merkur@itl.rtu.lv

- ✉ LSS / Yuri Merkuryev, Dept. of Modelling and Simulation Riga Technical University
Kalku street 1, Riga, LV-1658, LATVIA

LSS Officers

| | |
|------------------------|--|
| President | Yuri Merkuryev, merkur@itl.rtu.lv |
| Repr. EUROSIM | Yuri Merkuryev, merkur@itl.rtu.lv |
| Edit. Board SNE | Yuri Merkuryev, merkur@itl.rtu.lv |

Last data update December 2008

ROMSIM – Romanian Modelling and Simulation Society

ROMSIM has been founded in 1990 as a non-profit society, devoted to both theoretical and applied aspects of modelling and simulation of systems. ROMSIM currently has about 100 members from both Romania and Republic of Moldavia.

→ www.ici.ro/romsim/

✉ sflorin@ici.ro

- ✉ ROMSIM / Florin Stanciulescu,
National Institute for Research in Informatics, Averișcu
Av. 8 – 10, 71316 Bucharest, Romania

ROMSIM Officers

| | |
|------------------------|---|
| President | Florin Stanciulescu, sflorin@ici.ro |
| Vice president | Florin Hartescu, flory@ici.ro Marius Radulescu, mrادulescu@ici.ro |
| Secretary | Zoe Radulescu, radulescu@ici.ro |
| Repr. EUROSIM | Florin Stanciulescu, sflorin@ici.ro |
| Deputy | Florin Hartescu, flory@ici.ro |
| Edit. Board SNE | Florin Stanciulescu, sflorin@ici.ro |

Last data update March 2009

RNSS – Russian Simulation Society

NSS - The Russian National Simulation Society (Национальное Общество Имитационного Моделирования – НОИМ) was officially registered in Russian Federation on February 11, 2011. In February 2012 NSS has been accepted as an observer member of EUROSIM.

→ www.simulation.su

✉ yusupov@ias.spb.su

✉ RNSS / R. M. Yusupov,
St. Petersburg Institute of Informatics and Automation
RAS, 199178, St. Petersburg, 14th lin. V.O, 39

RNSS Officers

| | |
|------------------|--|
| President | R. M. Yusupov, yusupov@ias.spb.su |
| Chair Man. Board | A. Plotnikov, plotnikov@sstc.spb.ru |
| Secretary | M. Dolmatov, dolmatov@simulation.su |
| Repr. EUROSIM | R. M. Yusupov, yusupov@ias.spb.su |
| Deputy | B. Sokolov, sokol@ias.spb.su |
| Edit. Board SNE | Y. Senichenkov, sneyb@dcn.infos.ru |

Last data update February 2012

LIOPHANT Simulation

Liophant Simulation is a non-profit association born in order to be a trait-d'union among simulation developers and users; Liophant is devoted to promote and diffuse the simulation techniques and methodologies; the Association promotes exchange of students, sabbatical years, organization of International Conferences, organization of courses and stages in companies to apply the simulation to real problems.

→ www.liophant.org

✉ info@liophant.org

✉ LIOPHANT Simulation, c/o Agostino G. Bruzzone,
DIME, University of Genoa, Polo Savonese,
via Molinero 1, 17100 Savona (SV), Italy



ROMSIM Officers

| | |
|-----------------|--|
| President | A.G. Bruzzone, agostino@itim.unige.it |
| Director | E. Bocca, enrico.bocca@liophant.org |
| Secretary | A. Devoti, devoti.a@iveco.com |
| Repr. EUROSIM | A.G. Bruzzone, agostino@itim.unige.it |
| Deputy | F. Longo, f.longo@unica.it |
| Edit. Board SNE | F. Longo, f.longo@unica.it |

Last data update March 2012

SNE – Simulation Notes Europe

Simulation Notes Europe publishes peer reviewed *Technical Notes*, *Short Notes* and *Overview Notes* on developments and trends in modelling and simulation in various areas and in application and theory. Furthermore SNE documents the ARGESIM Benchmarks on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions (*Benchmark Notes*). Special *Educational Notes* present the use of modelling and simulation in and for education and for e-learning.

SNE is the official membership journal of EUROSIM, the Federation of European Simulation Societies. A News Section in SNE provides information for EUROSIM Simulation Societies and Simulation Groups.

SNE is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With Online SNE the publisher ARGESIM follows the Open Access strategy, allowing download of published contributions for free. Since 2011 Online SNE contributions are identified by an DOI (Digital Object Identifier) assigned to the publisher ARGESIM (DOI prefix 10.11128). Print SNE, high-resolution Online SNE, source codes of the *Benchmarks* and other additional sources are available for subscription via membership in a EUROSIM society.

Authors Information. Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the SNE Editorial Office. SNE distinguishes different types of contributions (*Notes*):

- *Overview Note* – State-of-the-Art report in a specific area, up to 14 pages, only upon invitation
- *Technical Note* – scientific publication on specific topic in modelling and simulation, 6 – 8 (10) pages
- *Education Note* – modelling and simulation in / for education and e-learning; max. 6 pages
- *Short Note* – recent development on specific topic, max. 4 pages
- *Software Note* – specific implementation with scientific analysis, max 4 pages
- *Benchmark Note* – Solution to an ARGESIM Benchmark; basic solution 2 pages, extended and commented solution 4 pages, comparative solutions on invitation

Interested authors may find further information at SNE's website → www.sne-journal.org (layout templates for *Notes*, requirements for benchmark solutions, etc.).



ASIM



ASIM



ASIM



ASIM - Buchreihen / ASIM Book Series

Fortschritte in der Simulationstechnik / Frontiers in Simulation Monographs - Conference Proceedings

Simulation und Optimierung in Produktion und Logistik.

L. März, W. Krug, O. Rose, G. Weigert, G. (eds.), ISBN 978-3-642-14535-3, Springer, 2011

Integrationsaspekte der Simulation: Technik, Organisation und Personal.

Zülch, G., Stock, P. (eds.), ISBN 978-3-86644-558-1, KIT Scientific Publishing, Karlsruhe, 2010

Verifikation und Validierung für die Simulation in Produktion und Logistik, Vorgehensmodelle und Techniken.

M. Rabe, S. Spieckermann, S. Wenzel (eds.); ISBN: 978-3-540-35281-5, Springer, Berlin, 2008

Advances in Simulation for Production and Logistics Applications.

M. Rabe (ed.), ISBN 978-3-8167-7798-4, Fraunhofer IRB-Verlag, Stuttgart, 2008

Modellierung, Regelung und Simulation in Automotive und Prozessautomation

- Proceedings 5. ASIM-Workshop Wismar 2011. -

C. Deatcu, P. Dünow, T. Pawletta, S. Pawletta (eds.), ISBN 978-3-901608-36-0, ASIM/ARGESIM, Wien, 2011.

Reihe Fortschrittsberichte Simulation / Series Advances in Simulation

Ch. Steinbrecher: **Ein Beitrag zur prädiktiven Regelung verbrennungsmotorischer Prozesse**

FBS 18, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-68-1, ARGESIM Report 68

O. Hagendorf: **Simulation-based Parameter and Structure Optimisation of Discrete Event Systems**

FBS 17, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-67-4, ARGESIM Report 67

D. Leitner: **Simulation of Arterial Blood Flow with the Lattice Boltzmann Method**

FBS16, ASIM/ARGESIM Vienna, 2009; ISBN 978-3-901608-66-7, ARGESIM Report 16

Th. Löscher: **Optimisation of Scheduling Problems Based on Timed Petri Nets.**

FBS 15, ASIM/ARGESIM Vienna, 2009; ISBN 978-3-901608-65-0, ARGESIM Report 15

J. Wöckl: **Hybrider Modellbildungszugang für biologische Abwasserreinigungsprozesse.**

FBS 14, ASIM/ARGESIM Vienna, ISBN 3-901608-64-8, 2006, ARGESIM Report 14,

M. Gyimesi: **Simulation Service Providing als Webservice zur Simulation Diskreter Prozesse.**

FBS 13, ASIM/ARGESIM Vienna, ISBN 3-901-608-63-X, 2006, ARGESIM Report 13

R. Fink: **Untersuchungen zur Parallelverarbeitung mit wissenschaftlich-technischen Berechnungsumgebungen.**

FBS 12, ASIM/ARGESIM Vienna, 2008; ISBN 978-3-901608-62-9, ARGESIM Report 12

H. Ecker: **Suppression of Self-excited Vibrations in Mechanical Systems by Parametric Stiffness Excitation.**

FBS 11, ASIM/ARGESIM Vienna, ISBN 3-901-608-61-3, 2006, ARGESIM Report 11

Orders:

ASIM/ARGESIM Office Germany, Hochschule Wismar, PF 1210, 23952 Wismar, Germany

ASIM/ARGESIM Geschäftsstelle Österreich, c/o DWH, Neustiftgasse 57, 1040 Vienna, Austria

Order and Download via ASIM webpage in preparation


Info: www.asim-gi.org, info@asim-gi.org



REPORTS



REPORTS

EUROSIM Federation of European Simulation Societies

New Members. EUROSIM got two new members: on occasion of the board meeting in February 2012, Vienna, the membership applications of two new societies have been approved:

- **NSS** - Russian National Simulation Society
- **LIOPHANT Simulation**, Italy.

At present, both societies have status *Observer Member*, application for full membership are in preparation. Detailed information on the new member societies are published in this and in the next regular SNE issue.



Khalid al Begain, EUROSIM president, welcoming Yuri Senichenkov of NSS (left) and Agostino Bruzzone (right) of LIOPHANT Simulation as new members

Other important topics of the board meeting were EUROSIM membership potentials (news from BULSIM, SRBISIM, Kosovo), information providing for the EUROSIM webpages, planning of EUROSIM events between EUROSIM congresses, and programme for the EUROSIM Congress 2013 in Cardiff.

BULSIM, a Bulgarian simulation society, will apply for the membership, as soon as the society will reach all conditions. Edmond Hajrizi, representant of the Kosovo Simuzlation Society, presented the society's activities and was invited to submit a membership application.



Participants at Vienna board meeting:
F. Maceri, F. Breitenacker, K. Al Begain, B. Zupancic,
Y. Senichenkov, L. Bobrowski, A. Bruzzone, E. Jiminez,
M. Alexik, E. Juuso, E. Hajrizi.



ASIM German Simulation Society Arbeitsgemeinschaft Simulation

Coming Events. As in December 2012 the well-known WINTERSIM conference is organised for the first time in Europe (Berlin), ASIM has postponed the autumn 2012 conference ASIM SPL *Simulation in Production and Logistics* to autumn 2013 (**ASIM SPL 2013**). ASIM members are organising contributions and special sessions for WINTERSIM 2012, Berlin.

WINTERSIM 2012

Winter Simulation Conference

December 9-12, 2012, Berlin, Germany

www.wintersim.org

SLOSIM – Slovenian Society for Simulation and Modelling



Past Events. From February 14 – 17, 2012, SLOSIM members attended **MATHMOD 2012** Conference in Vienna, Austria. The conference was again a big success with more than 400 participants, with high quality contributions, with many tutorials, plenary lectures, a poster session and an outstanding social programme.

Traditionally, SLOSIM society took an active part in conference organization, where our members organized two tutorials (G. Mušič), two minisymposia (G. Mušič and M. Atanasijević-Kunc) and submitted ten papers. 15 society members attended the conference. Our researchers M. Blazinšek and M. Arh under mentorship of I. Škrjanc won the 2nd place in the poster competition, with the contribution *Comparison between the auto-tuning of PID, PFC and modified-PFC controllers*.

Coming Events. For **I3M 2012 - 9th International Multidisciplinary Modelling & Simulation Multiconference** (Vienna Sept. 2012), with e.g. **EMSS 2012** and **IWISH 2012 International Workshop on Innovative Simulation for Health Care**, ASIM and SLOSIM simulationists are preparing the following special sessions:

- Modelling and Simulation in and for Education
- Math. Modeling and Health Technology Assessment
- Modelling and Simulation in Physiology and Medicine
- Discrete and Combined Simulation



RNSS Russian Simulation Society

NSS (RNSS) - the Russian National Simulation Society (Национальное Общество Имитационного Моделирования – НОИМ) was officially registered in Russian Federation on February 11, 2011. In February 2012 NSS has been accepted as an observer member of EUROSIM.

RNSS was officially registered in Russian Federation on February 11, 2011. Russian companies such as Elina-Computer Ltd. (www.gpss.ru, Kazan), XJ Technologies Ltd. (Saint Petersburg www.xjtek.com), CGSC Transas Technologies (www.transas.ru, Saint Petersburg) and the employees of the Federal State Budgetary Institution of Science, St. Petersburg, Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS) (www.spiiras.nw.ru), JSC “Shipbuilding and Ship Repair Technology Center” (www.sstc.spb.ru) and Saint Petersburg State Polytechnic University are the founders of the Society (more details about founder at website).

→ www.simulation.su

✉ yusupov@iias.spb.su

✉ RNSS / R. M. Yusupov,
St. Petersburg Institute of Informatics and Automation
RAS, 199178, St. Petersburg, 14th lin. V.O, 39

RNSS General Information

President of the Society. Yusupov Raphael Midhatovich, Doctor of Engineering Science, Corresponding Member of the Russian Academy of Sciences, Director of St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, Saint Petersburg; yusupov@iias.spb.su.



Chairman of Management Board– Plotnikov Alexander Mikhailovich, Chief of CAD/CAM branch of Joint-Stock Corporation Shipbuilding & Ship Repair Technology Center, Saint-Petersburg plotnikov@sstc.spb.ru.



The main objectives of the society are:

- development of ideas, methods, instrumental and software means of simulation and their application in Russia;
- extension of application areas and promotion of simulation in science, industry, on transport, in communication, defense, in social and other areas;
- formation and carrying out of the coordinated scientific and technical policy in the field of simulation in Russia;
- coordination of activities of society members in the field of development of simulation methods as well as instrumental and software means when carrying out large state projects in the field of modeling;
- assistance to training in the simulation area, to improvement of professional skills of experts and an exchange of experience between the society members;
- assistance in mobilization of financial and other resources for obtaining and execution of projects in the simulation field;
- development of the international cooperation in the simulation field, including involvement in international symposiums, conferences, seminars and exhibitions.

RNSS Activities

Currently, an active preparation for reception of new members to the structure of NC “NSS” is conducted. In the coming months the society website (www.simulation.su) will be created. In the “Applied Informatics” Journal No.4 (34), 2011, the conversation with Prof. R. M. Yusupov about the purposes and the immediate tasks of the society is published (the interview text can also be accessed online at www.gpss.ru).

The society’s primary objective is to establish relations between experts in the field of simulation in Russia, to define a circle of the industrial organizations which are using simulation as means of research and designing and to understand how widely the simulation and simulation tools are used in education.

We also want to work in tight and effective cooperation with foreign colleagues and we will be glad to establish any new contacts with the societies of simulation promoting development of the theory and practice of simulation.



The immediate task of the National Simulation Society is to become an official member of the European Society for Simulation Modeling (EUROSIM). Also it is planned to establish close connections with other international and national communities, directing the representatives to the international conferences organised in Europe.

Past Events

The necessity of creation of a simulation society has been realized a long time ago. In our country, great attention has always been paid to the theory and practice of simulation. Nowadays the role of simulation in science, industry, education is ever increasing. We have managed to overcome organizational difficulties, and even could arrange a number of joint events with foreign participants.

On June 29-30, 2011 within the International Maritime Defence Show IMDS-2011, St. Petersburg, the International Scientific-Practical Conference «**Simulation and Complex Modeling in Marine Engineering and Marine Transporting Systems**» – SCM MEMTS 2011 was held. This conference was an official action of the International Maritime Defence Show. The Organizers of the conference were JSC “Shipbuilding and Ship Repair Technology Center” (JSC “SSTC”) and the National Simulation Society. The international co-organizers of the conference were: United Kingdom Simulation Society (UKSIM), Latvian Simulation Society (LSS), International Mediterranean & Latin American Council of Simulation (IM & LACS).

The conference topics have covered various simulation application areas concerned with maritime activities such as: designing, building and maintenance of vessels and the sea engineering; sea transport systems, sea cargo transportation and logistics; ship-building manufactures and port facilities. The speakers and listeners of the conference represented 19 organizations, enterprises, firms and higher educational institutions of Russia, Italy, Latvia and Brazil. Within the conference, a round-table was held.

The Fifth (Anniversary) All-Russia Scientific-Practical Conference on Simulation Modeling and its Application in Science and the Industry „Simulation Modeling. The Theory and Practice” **SIMMOD-2011** was held in St.-Petersburg on October 19-21, 2011.

The Conference has been organized by JSC “Shipbuilding and Ship Repair Technology Center”, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIA RAS) and the National Simulation Society.

The conference program consisted of the reports concerned, first of all, with the area of simulation systems with the discrete events and time. Papers on simulation of the continuous systems on the basis of differential equations were only accepted if they contained the results related to the general methodological and program-technical problems of simulation modeling.

Upcoming Events

The Second International Scientific-Practical Conference “Simulation and Complex Modelling in Marine Engineering and Marine Transportation Systems” – SCM MEMTS 2013 will be held within the International Maritime Defence Show (IMDS-2013, St. Petersburg, www.navalshow.ru) on June 26-27, 2013. The Conference is an official action of the International Maritime Defence Show. The Organizers of the conference are JSC “SSTC”, SPIIA RAS and the National Simulation Society.

The conference topics will cover the following areas of maritime activity:

- designing, building and maintenance of vessels and the sea engineering;
- sea transport systems, sea cargo transportation and logistics;

SHIP-BUILDING MANUFACTURES AND PORT FACILITIES. SCM MEMTS 2013

2nd International Scientific-Practical Conference
«Simulation and Complex Modeling in Marine Engineering and Marine Transportation Systems»
June 26 – 27, 2013, Saint-Petersburg, Russia
www.navalshow.ru

The Sixth All-Russia Scientific-Practical Conference on Simulation and its Application in Science and Industry „Simulation. The Theory and Practice”, SIMMOD-2013 will take place in Kazan from October 16-18, 2013. The Organizers of the conference are JSC “SSTC”, SPIIA RAS and the National Simulation Society.



Topics covered by SIMMOD-2013 will include but are not limited to:

- Theoretical foundations and methodology of simulation;
- Methods of model quality estimation;
- Methods and systems of the distributed simulation;
- Simulation of global processes;
- Tools for simulation automation and visualization;
- System dynamics (with mandatory presence of an imitative component);
- Practical application of simulation and tools for automation of simulation; decision making on the basis of modeling results;
- Simulation in training and education.

SIMMOD 2013

6th All-Russia Scientific-Practical Conference on Simulation Modeling and its Application in Science and Industry

«Simulation Modeling. The Theory and Practice»

October 16– 18, 2013, Kazan, Russia

www.gpss.ru

In 2012, the 11th conference “Computer Simulation of Complex Systems”, CSCS – 2012 will take place in St. Petersburg. The conference is mainly targeted at lecturers, postgraduate students and students of Russian universities.

The conference sections are: “The Visual Environments of Simulation of Complex Dynamic Systems”, “Computer Models of Complex Dynamic Systems”, “Computer Tools in Education” and “Computer Models of Students and Postgraduate Students”. The conference organiser is St. Petersburg State Polytechnic University (Fac. of Eng. Cybernetics, <http://dcn.ftk.spbstu.ru>).

CSCS 2012

11th Conference

«Computer Simulation of Complex Systems»

September, 2012, Saint Petersburg, Russia

Miscellaneous

In Russia, the simulation both as a tool of scientific research and a tool for designing new technical systems has a rich history. It suffices to mention works of N. N. Janenko, N. P. Buslenko, V. M. Glushkov, N. N. Moiseev, A. A. Samarsky, P. S. Krasnoshchekov, Yu. N. Pavlovsky, G. I. Savin and B. N. Chetverushkin. The “Mathematical Modeling” journal is available to European readers online at <http://www.imamod.ru/journal/>.

The development teams of the environments of SITE simulation, Simulation in Technical Equipment (<http://energy.power.bmstu.ru/mvtu/>), AnyLogic (www.xjtek.com), MvStadium (www.mvstadium.com) and RMD (www.rand-service.com) are already known to the readers of SNE and participants of MATHMOD Conference. One of our tasks is to introduce the European colleagues to other interesting simulation tools created by the teams of scientists from Moscow, St. Petersburg, Ekaterinburg, Novosibirsk, Nizhni Novgorod, Omsk, Perm, Kazan and other cities of Russia.

The main results of activity of the Russian companies and the enterprises over the last ten years are generally presented in works of the All-Russia Conference on Simulation, SIMMOD (2003, 2005, 2007, 2009 and 2011; www.gpss.ru).

The analysis of works of five SIMMOD conferences shows that interest in simulation teaching, theoretical research and application-oriented development in the field of simulation and complex modeling has been increasing in Russia lately. Nowadays more than 10 thousand experts familiar with essentials of computer simulation graduate annually from 150 high schools of Russia. Similar situation can be observed in CIS countries. Simulation has been integrated not only into computer science curriculum but also into some economics courses.

Y. Senichenkov, sneyb@dcn.infos.ru

*515.000.000 KM, 380.000 SIMULATIONEN
UND KEIN EINZIGER TESTFLUG.*

DAS IST MODEL-BASED DESIGN.

Nachdem der Endabstieg der beiden Mars Rover unter Tausenden von atmosphärischen Bedingungen simuliert wurde, entwickelte und testete das Ingenieur-Team ein ausfallsicheres Bremsraketen-System, um eine zuverlässige Landung zu garantieren. Das Resultat – zwei erfolgreiche autonome Landungen, die exakt gemäß der Simulation erfolgten. Mehr hierzu erfahren Sie unter: www.mathworks.de/mbd

**MATLAB[®]
& SIMULINK[®]**



EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013



EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM2013, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics

The EUROSIM 2013 Congress will include invited talks, parallel, special and the poster sessions. The Congress topics of interest include, but are not limited to:

Intelligent Systems and Applications
Hybrid and Soft Computing
Communication Systems and Networks
Case Studies, Emergent Technologies
Workflow Modelling and Simulation
Web-based Simulation
Security Modelling and Simulation
Computer Games and Simulation
Neural Networks, Fuzzy Systems &
Evolutionary Computation
Autonomous Mental Development
Bioinformatics and Bioengineering
Circuits, Sensors and Devices

e-Science and e-Systems
Image, Speech & Signal Processing
Human Factors and Social Issues
Industry, Business, Management
Virtual Reality, Visualization and
Computer Games
Internet Modelling, Semantic Web
and Ontologies
Computational Finance & Economics
Systems Intelligence and
Intelligence Systems
Adaptive Dynamic Programming and
Reinforcement Learning

Methodologies, Tools and
Operations Research
Discrete Event /RT Systems
Mobile/Ad hoc wireless
networks, mobicast, sensor
placement, target tracking
Control of Intelligent Systems
and Control Intelligence
Robotics, Cybernetics, Control
Engineering, & Manufacturing
Energy, Power, Transport,
Logistics, Harbour, Shipping
and Marine Simulation
Semantic & Data Mining

Congress Venue / Social Events

The Congress will be held in the historic and magnificent City Hall in the heart of Cardiff, the capital city of Wales. The Gala Dinner will be held in the main hall of the National Museum of Wales. Social activities include visits to Cardiff Castle and Caerphilly Castle.

Congress Team: K. Al-Begain, A. Orsoni, R. Zobel, R. Cant, D. Al-Dabass; kbegain@glam.ac.uk

Info: www.eurosim2013.info