

SNE SIMULATION NOTES EUROPE



Volume 23 No.3-4 December2013

doi: 10.11128/sne.23.34.1020



Journal on Developments and
Trends in Modelling and Simulation

Membership Journal for Simulation
Societies and Groups in EUROSIM

Print ISSN 2305-9974
Online ISSN 2306-0271





EUROSIM 2016

9th EUROSIM Congress on Modelling and Simulation

City of Oulu, Finland, September 12 – 16, 2016



EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM 2016, 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 2016 Congress will include invited talks, parallel, special and poster sessions, exhibition and versatile technical and social tours. The Congress topics of interest include, but are not limited to:

Intelligent Systems and Applications
Hybrid and Soft Computing
Data & Semantic Mining
Neural Networks, Fuzzy Systems & Evolutionary Computation
Image, Speech & Signal Processing
Systems Intelligence and Intelligence Systems
Autonomous Systems
Energy and Power Systems
Mining and Metal Industry
Forest Industry
Buildings and Construction
Communication Systems
Circuits, Sensors and Devices
Security Modelling and Simulation

Bioinformatics, Medicine, Pharmacy and Bioengineering
Water and Wastewater Treatment, Sludge Management and Biogas Production
Condition monitoring, Mechatronics and maintenance
Automotive applications
e-Science and e-Systems
Industry, Business, Management, Human Factors and Social Issues
Virtual Reality, Visualization, Computer Art and Games
Internet Modelling, Semantic Web and Ontologies
Computational Finance & Economics

Simulation Methodologies and Tools
Parallel and Distributed Architectures and Systems
Operations Research
Discrete Event Systems
Manufacturing and Workflows
Adaptive Dynamic Programming and Reinforcement Learning
Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
Control of Intelligent Systems
Robotics, Cybernetics, Control Engineering, & Manufacturing
Transport, Logistics, Harbour, Shipping and Marine Simulation

Congress Venue / Social Events The Congress will be held in the City of Oulu, Capital of Northern Scandinavia. The main venue and the exhibition site is the Oulu City Theatre in the city centre. Pre and Post Congress Tours include Arctic Circle, Santa Claus visits and hiking on the unique routes in Oulanka National Park.

Congress Team: The Congress is organised by SIMS - Scandinavian Simulation Society, FinSim - Finnish Simulation Forum, Finnish Society of Automation, and University of Oulu. Esko Juuso EUROSIM President, Erik Dahlquist SIMS President, Kauko Leiviskä EUROSIM 2016 Chair

Info: www.eurosim.info, office@automaatioseura.fi

Editorial

Dear Readers – This third issue of SNE Volume 23 again combines individually submitted contributions and post-conference publications submitted from EUROSIM societies' conferences. In SNE 23(3-4), the post-conference publications have been submitted from SIMS Conference 2012 (SIMS – Scandinavian Simulation Society), from MATHMOD 2012 (Vienna Conference on Mathematical Modelling, Vienna, Austria), and from ERK 2013 Conference (Portoroz, Slovenia; SLOSIM - Slovenian Simulation Society's track on modelling and simulation track. Again the broad spectrum of modelling and simulation is reflected in the contributions, from modelling analysis to computational case studies. We are also glad, that we can continue our series on ARGESIM Benchmarks on 'Modelling Approaches and Simulation Implementations' with two solutions – one being a classical two-page solution, the other one being an extended comparative solution – which will be the general trend for our benchmark solutions.

And this issue also concludes the series of three front cover graphics, which Vlatko Ceric provided for SNE Volume 23.

I would like to thank all authors for their contributions, and the organizers of the EUROSIM conferences for co-operation in post-conference publication, and the ARGESIM SNE staff for helping to manage the SNE administration and the improved SNE layout and extended templates for submissions (now also tex), And especially Vlatko Ceric for his graphics..

Felix Breiteneker, SNE Editor-in-Chief, eic@sne-journal.org; felix.breiteneker@tuwien.ac.at

Contents SNE 23(3-4)

SNE doi: 10.11128/sne.22.34.1020

An Approximate Differentiation Method of Inverse Simulation based on a Continuous System Simulation Approach. <i>D. J. Murray-Smith</i>	111
Diffusion Approximation in a Stochastic Cellular Automaton Model for Epidemics. <i>M. Bicher</i>	117
Mathematical Framework for Graphical Simulation Models of Dynamical Systems in SIMULINK. <i>A. Körner</i>	123
Exploring the Possibilities of Co-Simulation with CATIA V6 Dynamic Behavior Modeling. <i>P. Smolek, B. Heinz, H. Ecker, F. Breiteneker</i>	129
Dynamic Properties of the Non-linear Cholesterol-level Control in the Cell. <i>A. Belić</i>	133
Comparison Open Modelica–Dymola for Power Plant Simulation. <i>E. T. Aparicio, E. Dahlquist, H. Li, P. Fritzson, P. Östlund</i>	139
A Case Study on Object-oriented Modelling and Simulation of Machine Tools. <i>B. Heinzl, M. Landsiedl, F. Dür, A.-A. Dimitriou, W. Kastner, F. Breiteneker</i>	147
A Case Study On Coupling Thermo-dynamical and Electro-mechanical Models in Modelica-based Simulators. <i>M. Rößler</i>	155
Scripting Modelica Models using Python. <i>B. Lie, F. Haugen</i>	161
Acausal Modeling and Simulation of the Standalone Solar Power Systems as Hybrid DAEs. <i>A. M. Dizqah, K. Busawon, P. Fritzson</i>	171
Hydro Power Systems: Scripting Modelica Models for Operational Studies in Education. <i>D. Winkler, B. Lie</i>	179
Simulation of Heat Radiation Asymmetry with Maple. <i>I. Perjési-Hámori</i>	185
Short Remark on Lateral Vibration of Functionally Graded Beams. <i>G. Maróti</i>	189
An Agent-based Approach to ARGESIM Benchmark C6 'Emergency Department – Followup Treatment' using Anylogic. <i>K. Ölsböck, M. Noya, P. Einzinger</i>	193
Comparison of MATLAB, Simulink and AnyLogic Approach to ARGESIM Benchmark C9 'Fuzzy Control of a Two Tank System'. <i>S. Hödlmoser, F. Kitzler</i>	195
EUROSIM Societies Info & News	N1-N8

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, with main topics being simulation aspects and interdisciplinarity.

Individual submission of scientific papers are welcome, as well as post-conference publications of contributions from conferences of EUROSIM societies.

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 2012 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 members of EUROSIM societies.

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

SNE Issue 23(3-4) Dec. 2013 doi: 10.11128/sne.22.34.1020

→ www.sne-journal.org

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

✉ 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:

- David Al-Dabass, david.al-dabass@ntu.ac.uk
Nottingham Trent University, UK
- Felix Breitenecker, Felix.Breitenecker@tuwien.ac.at
Vienna Univ. of Technology, Austria, Editor-in-chief
- Maja Atanasijevic-Kunc, maja.atanasijevic@fe.uni-lj.si
Univ. of Ljubljana, Lab. Modelling & Control, Slovenia
- Aleš Belič, ales.belic@sandoz.com
Sandoz / National Inst. f. Chemistry, Slovenia
- Peter Breedveld, P.C.Breedveld@el.utwente.nl
University of Twente, Netherlands
- Agostino Bruzzone, agostino@itim.unige.it
Università degli Studi di Genova, Italy
- Francois Cellier, fcellier@inf.ethz.ch
ETH Zurich, Switzerland
- Vlatko Čerić, vceric@efzg.hr
Univ. Zagreb, Croatia
- Russell Cheng, rhc@maths.soton.ac.uk
University of Southampton, UK
- Eric Dahlquist, erik.dahlquist@mdh.se, Mälardalen Univ., Sweden
- Horst Ecker, Horst.Ecker@tuwien.ac.at
Vienna Univ. of Technology, Inst. f. Mechanics, Austria
- Vadim Engelson, vadim.engelson@mathcore.com
MathCore Engineering, Linköping, Sweden
- 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
- Yuri Merkuryev, merkur@itl.rtu.lv, Riga Technical Univ.
- 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@dwh.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. Furthermore, SNE invites organizers of EUROSIM conferences to provide post-conference publication for the authors of their conference (with peer review).

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 p.
- *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 4-8 pages

Further info and templates (doc, tex) at SNE's website.

SNE Contact & Info

→ www.sne-journal.org

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

✉ SNE Editorial Office, ARGESIM / dwh Simulation Services,
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: J. Tanzler, F. Preysler, T. Wobruha;
C. Wytrzens, R. Leskovar et al.; Math. Modelling and Simulation Group, Vienna Univ. of Technology, Wiedner Hauptstrasse 8-10, 1040 Vienna, ✉ office@sne-journal.org

Print SNE: Grafisches Zentrum, TU Vienna,
Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria

Online SNE: ARGESIM / ASIM, c.o. dwh Simulation Services,
Neustiftgasse 57-59, 1070 Vienna, Austria

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 2013

An Approximate Differentiation Method of Inverse Simulation based on a Continuous System Simulation Approach

David J. Murray-Smith

University of Glasgow, School of Engineering, Rankine Building, Oakfield Avenue, Glasgow G12 8LT,
United Kingdom; *david.murray-smith@glasgow.ac.uk*

Simulation Notes Europe SNE 23(3-4), 2013, 111 - 116
DOI: 10.11128/sne.23.tn.10201
Received: October 5, 2013;
Accepted: November 30, 2013;

Abstract. Inverse simulation techniques allow inverse solutions to be found for a range of problems involving dynamic systems described by sets of linear or nonlinear ordinary differential equations. Techniques in common use generally involve iterative solutions based on discretised descriptions but continuous system simulation tools can also provide solutions and are often simpler to apply and computationally more efficient. This paper presents a method of inverse simulation which involves use of a very simple, but effective, approximation for derivative terms within the model. Discussion of results for linear and non-linear examples leads to the conclusion that this technique can be applied to a wide range of dynamic models that are of practical importance for engineering applications.

Introduction

Conventional modelling and simulation involves a process of finding a model ‘output’ for a given set of initial conditions and time history of ‘inputs’, whereas inverse modelling and simulation is a process in which ‘inputs’ are found that will produce prescribed model ‘outputs’. Inverse simulation methods provide an alternative to the use of mathematical techniques for the inversion of dynamic models, particularly in the nonlinear case. Not only do they avoid the complexities of the mathematical approaches for nonlinear models but they also provide insight that may not otherwise be so readily available from conventional simulation techniques.

A number of established inverse simulation methods involve discretisation of continuous models. Examples include the so-called differentiation approach [1], [2], the widely-used integration based approaches [3], [4] and optimisation-based methods [5], [6]. A paper by Thomson and Bradley provides a useful review [7] of some of these techniques, which are essentially iterative in nature and were developed, initially, for aeronautical applications.

Other techniques are based on continuous system simulation models and include the use of differential algebraic equation (DAE) solvers such as those available in the Modelica[®] environment (see, e.g. [8]) but DAE methods do not yet appear to have been applied routinely to large and complex models of the type that typically arise in engineering applications. A useful and proven alternative involves the use of feedback principles and continuous system simulation tools (see, e.g. [9-13]).

The approximate differentiation method of inverse simulation outlined in this paper provides yet another approach which involves continuous system simulation methods and may be simpler to apply in some cases. In common with other methods of inverse simulation it has obvious limitations, but can be used for a range of model structures of importance for engineering applications.

1 The Approximate Differentiation Method

This is a continuous simulation equivalent of the discrete ‘differentiation method’. The basic idea is to rearrange the given model in state space form so that the inputs of interest appear on the left hand side of the equations.

Derivatives of state variables appearing on the right hand side can then be approximated using a simple continuous representation based on the use of an integrator block and feedback pathway, as shown in the block diagram of Figure 1. This may be seen from the first order equation defining that system which is given by:

$$\frac{dw}{dt} = \frac{1}{T}(v(t) - w(t)) \quad (1)$$

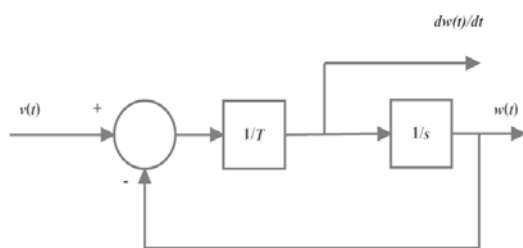


Figure 1: Block diagram of the approximate differentiator. The block $1/s$ represents the operation of integration in terms of the Laplace variable s .

If the variable $v(t)$ in Figure 1 is replaced by $x_{id}(t)$ which represents the desired time history for a state variable x_i then, provided the time constant T is very small in relation to the dynamics of the given model, the variable $w(t)$ in Figure 1 is a close approximation to the desired variable $x_{id}(t)$ and to the state variable $x_i(t)$. The quantity found at the input to the integrator block in Figure 1 is given by $\frac{1}{T}(x_{id}(t) - x_i(t))$ and is thus an approximation to the derivative \dot{x}_i . Hence a derivative of a state variable \dot{x}_i within a given state-space model may be replaced by a quantity $\frac{1}{T}(x_{id}(t) - x_i(t))$ where $x_{id}(t)$ is the desired time history.

The approach, which was mentioned in an invited keynote lecture at the 8th EUROSIM Congress in September 2013 [14], is best presented through a simple illustrative example which has also been used in investigations of other inverse simulation methods [6], [15]. Consider a linear single-input single-output system model of the form:

$$\dot{x} = Ax + Bu \quad (2)$$

$$y = Cx + Du \quad (3)$$

$$\text{where } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}, B = \begin{bmatrix} 1 \\ -5 \\ 69 \end{bmatrix},$$

$$C = [1 \ 0 \ 0], D = 0$$

Simple linear analysis shows that this linear single-input single-output (SISO) system model has poles at positions $s = -1$ rad/s, $s = -2$ rad/s and $s = -3$ rad/s and zeros at $s = -0.5000 \pm j7.0534$ rad/s. The range of frequencies of interest for this model is from 0 to 30 rad/s. The set of ordinary differential equations for the system as given above is:

$$\dot{x}_1 = x_2 + u \quad (4)$$

$$\dot{x}_2 = x_3 - 5u \quad (5)$$

$$\dot{x}_3 = -6x_1 - 11x_2 - 6x_3 + 69u \quad (6)$$

$$y = x_1 \quad (7)$$

Let the desired output be denoted by x_{1d} . The derivative \dot{x}_1 may then be approximated by $\frac{1}{T}(x_{1d} - x_1)$ and we have a new 'output equation' of the form:

$$u = x_2 - \dot{x}_1 = x_2 - \frac{1}{T}(x_{1d} - x_1) \quad (8)$$

We now have a modified set of equations of the form:

$$\dot{x}_1 = -\frac{1}{T}x_1 + \frac{1}{T}x_{1d} \quad (9)$$

$$\dot{x}_2 = \frac{5}{T}x_1 + 5x_2 + x_3 - \frac{5}{T}x_{1d} \quad (10)$$

$$\dot{x}_3 = -(6 + \frac{69}{T})x_1 - 80x_2 - 6x_3 + \frac{69}{T}x_{1d} \quad (11)$$

$$u = -\frac{1}{T}x_1 - x_2 + \frac{1}{T}x_{1d} \quad (12)$$

This set of equations has zeros at $s = -1$ rad/s, $s = -2$ rad/s, $s = -3$ rad/s and poles at $s = -0.5 \pm j7.0534$ rad/s and at $s = -1/T$ rad/s so, clearly, the zeros of the inverse simulation model are at the same positions as the poles of the given model and the poles lie at the positions of the zeros of that model, apart from an additional pole at $s = -1/T$. Provided the time constant T can be made very small, this additional pole of the inverse simulation can be positioned at a point in the s -plane far from the other poles and zeros, where its effect is insignificant. For example, a value of T of 0.001s would give an additional pole at $s = -1000$ rad/s, and this is so far removed from all the other poles and zeros that it would have a negligible influence on the dynamic behaviour of the inverse simulation. One could, of course, make the time constant T even smaller but this would tend to increase the stiffness of the inverse simulation and there is a clear trade-off between the overall accuracy of the inverse simulation and computational efficiency.

Consider the specific case of an desired time history for the state variable $x_1(t)$ which takes the form of a triangular waveform involving a negative going ramp, starting from zero at time $t = 0$ with a gradient -0.5 units/s, changing to a positive slope of 0.5 units/s at time $t = 1.5$ s and then repeating this pattern at time $t = 3$ s.

Using MATLAB[®] software to implement the inverse simulation based on Equations (9)–(12) for this desired form of output waveform and a value of T of 0.001s, we find that the required input has the form shown in the upper trace of Figure 2. The lower traces show the required output, together with the output found from a forward simulation for the given model when the waveform obtained from the inverse simulation is applied as input. It can be seen from this that the two plots almost coincide, so the input found from inverse simulation generates the required output almost exactly for the chosen value of T .

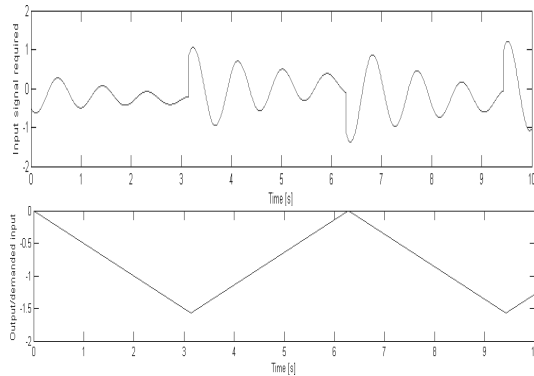


Figure 2: The upper trace shows the input found from inverse simulation of the linear SISO system of Equations (2) and (3). The lower traces (superimposed) show the demanded output together with the output obtained from application of the input found from inverse simulation to the conventional forward simulation model for this system.

2 An Example Involving a Nonlinear Model

Consider a mathematical model of a coupled-tanks system shown in schematic form in Figure 3, involving two interconnected tanks of liquid, (Tank 1 and Tank 2) with input flow rates, Q_{i1} and Q_{i2} respectively. These inputs are from electrically driven variable-speed pumps. There is a single outlet flow Q_{23} from the second tank which may be adjusted manually by means of a tap. Both tanks are equipped with sensors that can detect the level of liquid and provide a proportional output as an electrical voltage signal.

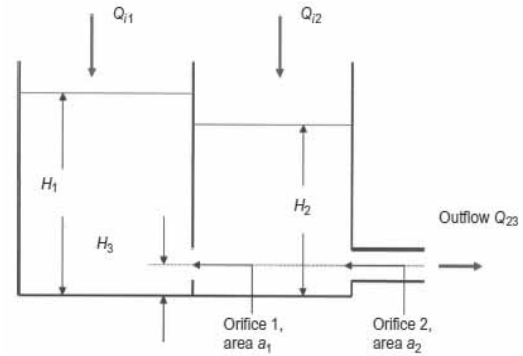


Figure 3: Schematic diagram of the coupled-tanks system.

If the levels of liquid in the two tanks (H_1 and H_2) are regarded as output quantities and the flow rates from the pumps as inputs, a two-input two-output second-order nonlinear state-space description may be developed for this system using simple physical principles based on the fact that the rates of change of volume of liquid in each tank must be equal to the difference between the total flow rate into that tank and the total flow rate out (see e.g. [13], [16]).

For situations in which the liquid level in Tank 1 is greater than the level in Tank 2 the equations are as follows:

$$\frac{dH_1}{dt} = \frac{Q_{i1}(t)}{A_1} - \frac{C_{d1}a_1}{A_1} \sqrt{2g(H_1(t) - H_2(t))} \quad (13)$$

$$\frac{dH_2}{dt} = \frac{Q_{i2}(t)}{A_2} + \frac{C_{d1}a_1}{A_2} \sqrt{2g(H_1(t) - H_2(t))} - \frac{C_{d2}a_2}{A_2} \sqrt{2g(H_2(t) - H_3)} \quad (14)$$

Values of parameters are defined below for a specific laboratory-scale system [13], [16]:

- Cross-sectional areas of tanks $A_1 = A_2 = 9.7 \times 10^{-3} \text{ m}^2$;
- Cross-sectional area of orifice 1 $a_1 = 3.956 \times 10^{-5} \text{ m}^2$;
- Cross-sectional area of orifice 2 $a_2 = 3.85 \times 10^{-5} \text{ m}^2$;
- Coefficient of discharge of orifice 1 $C_{d1} = 0.63$;
- Coefficient of discharge of orifice 2 $C_{d2} = 0.58$;
- Height of outlet above base of tank $H_3 = 0.03 \text{ m}$;
- Gravitational constant $g = 9.81 \text{ m/s}^2$;
- Maximum flow rate $Q_{i1\max} = Q_{i2\max} = 5 \times 10^{-5} \text{ m}^3/\text{s}$;
- Minimum flow rate $Q_{i1\min} = Q_{i2\min} = 0 \text{ m}^3/\text{s}$;
- Maximum liquid level $H_{1\max} = H_{2\max} = 0.3 \text{ m}$.

This model is nonlinear in structure because of the nonlinear relationship between the liquid levels in the Tanks 1 and 2 and the flow between them and also because of the nonlinear relationship between the output flow rate and the liquid level in Tank 2.

The inverse simulation developed from the application of the approximate differentiation method is readily obtained using the principles outlined above, and involves the following set of equations:

$$\frac{dH_1}{dt} = \frac{1}{T} [H_{1req}(t) - H_1(t)] \quad (15)$$

$$\frac{dH_2}{dt} = \frac{1}{T} [H_{2req}(t) - H_2(t)] \quad (16)$$

$$Q_{i1}(t) = \frac{A_1}{T} [H_{1req}(t) - H_1(t)] + C_{d1}a_1\sqrt{2g(H_1(t) - H_2(t))} \quad (17)$$

$$Q_{i2}(t) = \frac{A_2}{T} [H_{2req}(t) - H_2(t)] - C_{d1}a_1\sqrt{2g(H_1(t) - H_2(t))} - C_{d2}a_2\sqrt{2g(H_2(t) - H_3)} \quad (18)$$

where $H_{1req}(t)$ and $H_{2req}(t)$ are the required time histories of liquid levels in Tanks 1 and 2 respectively.

Figure 5 and Figure 6 show results obtained using MATLAB® for the method outlined above for a case involving the required time histories of level changes shown in Figure 4. The value of the time constant T used in the equations for the inverse simulation is 0.1s, which is very small in relation to the dynamic characteristics of the system. The results from the inverse simulation for the desired level changes of Figure 4 are given in Figure 5 and these show that the pattern of input flow rate changes do not cause either of the inputs to reach the maximum flow rate of $5 \times 10^{-5} \text{ m}^3/\text{s}$ or to drop to zero flow rate (the minimum allowed) at any time in the simulated test. The results do, however, show strong interactions between the two tanks, especially during the initial phase of the test where the input flow to Tank 2 falls rapidly to compensate for the rising level in Tank 1 (and thus allows the required level in Tank 2 to be maintained). When the input flow patterns shown in Figure 5 are applied to a forward simulation of the coupled-tanks system, the levels of liquid in the two tanks are very close to the required levels in Figure 4, as is shown by the results of Figure 6 which show maximum differences of the order of $1.4 \times 10^{-17} \text{ m}$ between the levels found from forward simulation results and the desired levels.

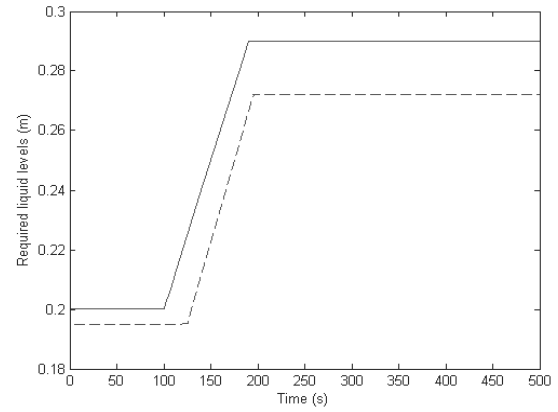


Figure 4: Required liquid levels for the first case considered.

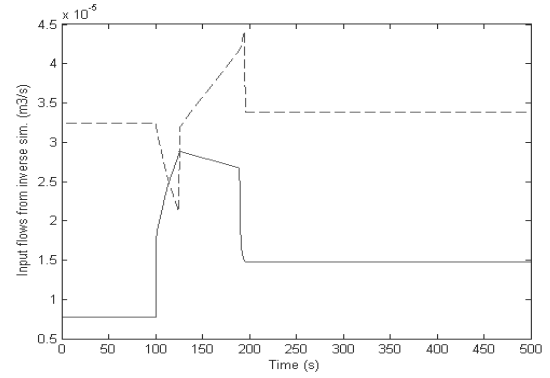


Figure 5: Flow rates determined by inverse simulation for the first case considered.

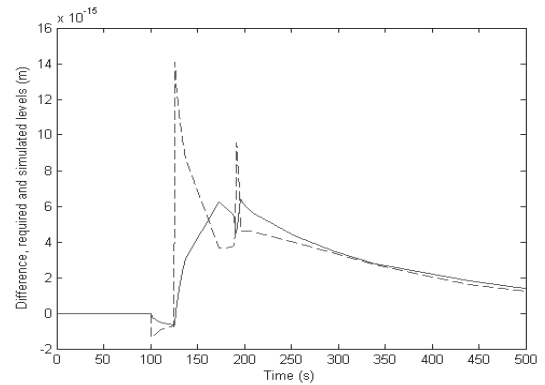


Figure 6: Differences between reference liquid level time histories and liquid levels found by applying inputs from inverse simulation to the forward simulation model.

The pattern of demanded reference levels shown in Figure 7 relates to a case in which the required level changes result in one of the input flow rates reaching its maximum value of $5 \times 10^{-5} \text{ m}^3/\text{s}$. The input flow rate to Tank 1 remains at that maximum value for a period of about 40s and then falls to a new constant value of about $4 \times 10^{-5} \text{ m}^3/\text{s}$, while the input flow to Tank 2 drops to zero, as is shown in Figure 8. In this case, if the input flow rates obtained from inverse simulation are applied to the forward simulation model, significant differences are found between the levels achieved and the desired levels, as shown in Figure 9. The inverse simulation thus shows, in this case, that for the given system configuration the required pattern of liquid level changes cannot be achieved. Hence, this might suggest that design changes would be required within the system if this pattern of level changes was an essential requirement for some specific application.

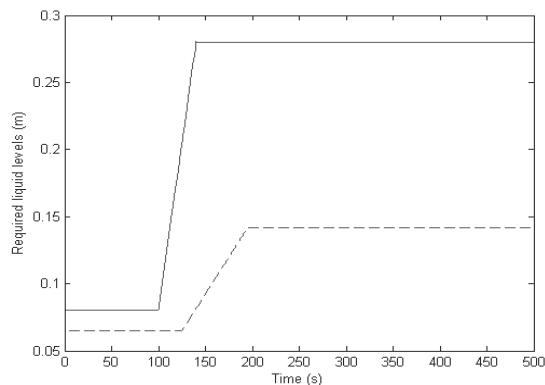


Figure 7: Required liquid levels for second case involving a larger difference between final levels in Tank 1 and Tank 2 resulting in a required flow rate for Tank 1 that exceeds the maximum.

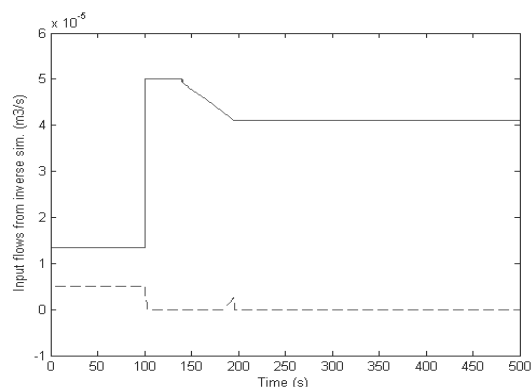


Figure 8: Input flow rates found using inverse simulation for level variations defined in Figure 7.

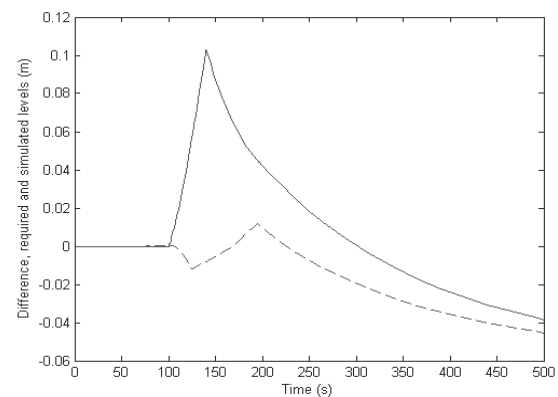


Figure 9: Differences between the desired levels and the levels found from forward simulation of the coupled-tanks model for the case defined by the inputs of Figure 8.

3 Discussion and Conclusions

The linear example involving a single-input single-output model defined in state-variable form shows that the approximate differentiation approach to inverse simulation can give results in which poles of the inverse simulation match the zeros of the given model. The approximation is equivalent to adding one pole at a point in the s -plane which is located on the negative real axis far from all the poles and zeros of the model. The effect of this additional pole resulting from the approximation can thus be made negligibly small by choosing a small value for the time constant associated with the differentiation process.

In the case of a multi-input system, if all the inputs are to be found by inverse simulation, the number of additional time constants would be equal to the number of inputs, but the time constants would again have negligible effect if they had appropriate small values.

The success of the approximate differentiation approach for the case of a model involving nonlinear equations has been demonstrated through use of the coupled-tanks example. In this case, provided the model input variables are within their limiting values, the inverse simulation gives an accurate prediction of the inputs required to produce desired model output time histories.

The method also provides the user with a clear indication of the effects of input limits when they arise, thus providing insight about the reasons why a specific desired time history of outputs is not achievable. Depending on the context in which inverse simulation is being applied, such situations may lead to modification of the desired pattern of outputs or to a change in the design of the system represented by the model.

As mentioned in the introductory section of this note, another commonly-used approach to inverse simulation which uses continuous system simulation methods is based upon feedback techniques (see e.g. [13]). This has been found to be a powerful approach and has been applied to a wide range of practical systems. However, that method requires the design of a feedback structure around the given simulation model, which can be time-consuming and difficult for those with little experience of closed-loop system design. It can also present significant problems if limit cycle oscillations arise. The approximate differentiation method thus provides an interesting alternative approach which avoids such difficulties but is also based on continuous system simulation principles.

References

- [1] Kato O, Sugiura I. An interpretation of airplane motion and control as an inverse problem. *Journal of Guidance Control and Dynamics*. 1986; 9:198 - 204
- [2] Thomson DG, Bradley R. Recent developments in the calculation of inverse solutions of the helicopter equations of motion. In: Proceedings of UK Simulation Council Conference, University College of North Wales, 9-11 Sept. 1987. UKSC, Ghent, Belgium, 1987, 227 - 234.
- [3] Hess RA, Gao C, Wang SH. A generalized technique for inverse simulation applied to aircraft maneuvers. *Journal of Guidance Control and Dynamics*. 1991; 14:920 - 926
- [4] Rutherford S, Thomson DG. Improved methodologies for inverse simulation. *Aeronautical Journal*. 1996; 100:79 - 86.
- [5] Celi R. Optimization-based inverse simulation of a helicopter slalom manoeuvre. *Journal of Guidance, Control and Dynamics*. 2000; 23:289 - 297
- [6] Lu L, Murray-Smith DJ, Thomson DG. Issues of numerical accuracy and stability in inverse simulation. *Simulation Modelling Practice and Theory*. 2008; 16:1350 - 1364
- [7] Thomson D, Bradley R. Inverse simulation as a tool for flight dynamics research – Principles and applications. *Progress in Aerospace Sciences*. 2006; 42:174 - 210
- [8] Thümmel M, Looye G, Kurze M, Otter M, Bals J. Non-linear inverse models for control. In: Schmitz G, ed., Proceedings of the 4th International Modelica Conference, Hamburg, March 7-8, 2005, Modelica Association, Linköping, Sweden, 2005, 267 - 279. [Online] <http://www.Modelica.org/events/Conference2005/>
- [9] Hamel PG. Aerospace vehicle modelling requirements for high bandwidth flight control. In: Cook MV, Rycroft MJ, eds., *Aerospace Vehicle Dynamics and Control*, Clarendon Press, Oxford, UK, 1994, 1 - 31.
- [10] Gray GJ, von Grünhagen W. An investigation of open-loop and inverse simulation as nonlinear model validation tools for helicopter flight mechanics. *Mathematical and Computer Modelling of Dynamical Systems*. 1998; 4:32 - 57
- [11] Buchholz JJ, von Grünhagen W. Inversion Impossible? Technical Report, University of Applied Sciences Bremen, Germany, Sept. 2004.
- [12] Tagawa Y, Fukui K. Inverse dynamics calculation of nonlinear model using low sensitivity compensator. In: Proceedings of Dynamics and Design Conference, 1994, 185 - 188.
- [13] Murray-Smith DJ. Feedback methods for inverse simulation of dynamic models for engineering systems. *Mathematical and Computer Modelling of Dynamical Systems*. 2011; 17:515 - 541
- [14] Murray-Smith DJ. Looking at problems the other way round: Engineering applications of inverse simulation based on continuous system simulation methods. Invited Keynote Lecture, 8th EUROSIM Congress, Cardiff, Wales, UK, 10-12 Sept. 2013. [Online] Summary and slides (Keynote Lecture 4) <http://eurosims2013.info/>
- [15] Lu L. Inverse Modelling and Inverse Simulation for Engineering Applications, Lambert, Saarbrücken, Germany, 2010.
- [16] Gong M, Murray-Smith DJ. A practical exercise in simulation model validation. *Mathematical and Computer Modelling of Dynamical Systems*. 1998; 4:100 - 117

Diffusion Approximation in a Stochastic Cellular Automaton Model for Epidemics

Martin Bicher^{1,2}

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

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

Simulation Notes Europe SNE 23(3-4), 2013, 117 - 122
DOI: 10.11128/sne.23.tn.10202
Received: September 10, 2013; Revised: November 10, 2013;
Accepted: November 30, 2013;

Abstract. Although microscopic models are nowadays getting more and more popular among, still the modelling approach lacks of appropriate mathematical theory to confidently rely on the outputs of the derived models. Especially unexpected chaotic group behaviour and the inability to validate and parametrise the model often leads to unusable simulations. The investigated test-case, a simple cellular automaton (CA) simulating the temporal development of a SIR (Susceptible-Infected-Recovered) type epidemic, shows a field of application for so called complexity theory. In order to explain and analyse the aggregated simulation results of the CA, certain methods usually used in Markov theory for quantum mechanics, basically extensions of so called diffusion approximation [1], are applied. Finally, already suspected, correlations to the solutions of the famous SIR differential equations, formerly derived by Kermack and McKendrick [2], can be proven with analytical methods and extended by convergence results and qualitative error estimations.

Introduction

Due to tough limited resources usage of modelling and simulation to support strategic planning has nowadays become an indispensable part of management. Especially the increasing number of simulations for emerging problems within so called soft-sciences like medicine, biology or sociology can be observed. Main reason for this development is the exponentially increase of computational resources (compare Moore's law [3]) making it possible to simulate very complex, individual-based

models, which, correctly validated, produce reliable results. Nevertheless the validation process for these models is very difficult, requires lots of data and heuristic parameter-sweeps for sensitivity analysis. The usage of microscopic models always involves the danger, that maybe unpredictable chaotic group behaviour distorts the results. Most of the microscopic modelling methods, like agent-based models and cellular automata, somehow lack of necessary mathematical basis.

Nevertheless compared to classic macroscopic modelling methods some of the advantages and disadvantages of so called microscopic or individual-based models can be summarized in Table 1.

Advantages	Disadvantages
Lower abstraction level compared to reality	Difficult to validate
Easy to understand for non-experts	High computational efforts
Very flexible regarding system changes	Difficult to document regarding reproducibility
More suitable for eye-catching visualisations	Sometimes unpredictable and chaotic results

Table 1: Some advantages and disadvantages of microscopic models.

Within epidemiology so called SIR (Susceptible – Infected - Recovered) strategy already poses the base for lots of flexible microscopic models for diseases and vaccine strategies (see e.g. [4],[5]). Hereby the spread of one single serotype in a certain environment among certain individuals is studied, wherein the individuals are divided into the aforementioned three subclasses. Infected individuals forward the spread of the disease among the susceptible individuals. Recovered individuals are meant to stay immune against the serotype for the further progress of the disease (died individuals are included here).

Several years ago a team of the AMSDM group (Applied Modelling, Simulation and Decision Making), a cooperation of the “dwh GmbH” and the group of Prof. Felix Breiteneker at Vienna University of Technology, created lots of epidemics-related teaching material for modelling lectures. This material was developed in the context of a huge project with the Federation of Austrian Social Insurance analysing the effect of different vaccine strategies against influenza viruses on the Austrian social system (population, financial aspects, etc.) supported by theoretical models ([6], [7]). Due to interesting results especially one of them attracted special attention. The emphasis is laid on a stochastic cellular automaton (short CA), in detail described in chapter 1, which can be used to simulate simple epidemics with the aforementioned SIR strategy. For more information the reader is referred to [18]. In addition the influence of vaccinations before the breakout of the disease can be investigated confirming the flexibility of agent-based models compared to most macroscopic models.

1 Comparison of two Modelling Approaches

Some resulting curves of the mentioned cellular automaton model can be seen in Figure 1. The three graphs located at the lower part of the figure are calculated by the number of cells sharing the same state. The green line shows the number of “susceptible” cells, the red and blue line illustrate the number of “infected” and “immune” cells. The upper part of the figure shows the state of the cellular automaton at a certain time during the simulation run.

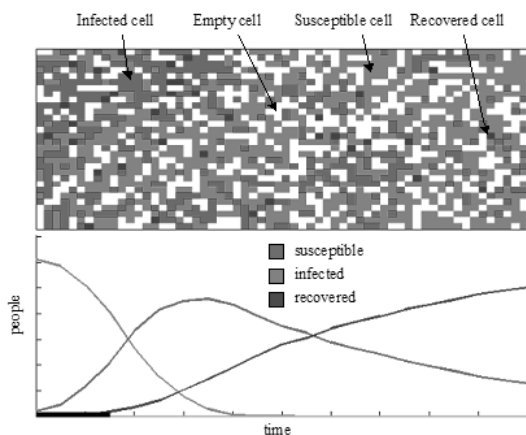


Figure 1: Example Result of the Cellular Automaton.

Studies (see [8]) have shown, that a comparison between the aggregated CA-results and the famous SIR differential equation model (short ODE model) by Kermack and McKendrick, 1927 [2], seen in (1) is justified.

$$\begin{aligned}\frac{dS}{dt} &= -\alpha IS \\ \frac{dI}{dt} &= \alpha IS - \beta I \\ \frac{dR}{dt} &= \beta I\end{aligned}\quad (1)$$

This nonlinear system of differential equations can only be solved using numerical integration algorithms, as it does not have any nontrivial analytical solutions. A MATLAB generated plot of the Runge-Kutta-Fehlberg (4th order with step-size control) approximation of the solution is shown in Figure 2.

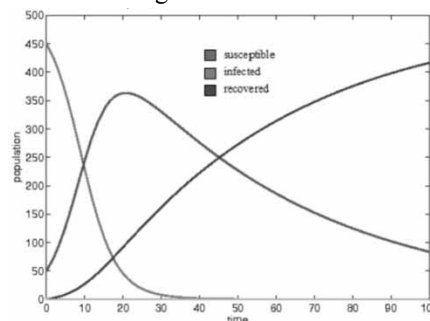


Figure 2: Example Result of Classic SIR Differential Equations.

It is undeniable that the solution curves of both models look very similar. Further comparative studies (also [9]) showed that there is a correlation between the parameters of the ODE model (Infection rate α , Regeneration rate β) and the stochastic CA model (movement rules, regeneration probability, neighbourhood, population-density and infection-probability) in form of closed equations. Unfortunately these were mostly derived via basic stochastics and experiments, without any statements regarding convergence and errors between those two approaches.

A possible way to compare these two completely different modelling ideas with mathematical techniques is presented in the following chapters. One must not forget that a stochastic, time and space discrete model is hereby compared with a deterministic, completely continuous model posing a big challenge. Starting to analyse the CA model by a series of transformations finally the ODE formulation will be derived. Convergence results and error estimations are going to appear as by-products of these.

2 Cellular Automaton – Definition

Although usually presented and implemented as a cellular automaton the most comprehensible description of this model is claimed in form of an agent-based model. The transition rules for the cellular automaton can be derived analogously. Although the underlying modelling concept is completely different in this simple case both modelling approaches (agent-based and CA) end up with the same model.

2.1 Space

Let Ω be a discrete two-dimensional rectangular grid with $M = M_x \cdot M_y$ aligned cells $c_{i,j}, i \in \{0, \dots, M_x\}, j \in \{0, \dots, M_y\}$. Each cell itself is partitioned into four cell-fractions: $\{c_{i,j,1,1}, c_{i,j,1,2}, c_{i,j,2,1}, c_{i,j,2,2}\}$.

2.2 Agents (non-empty cells)

A number $N \leq 4M$ of agents $a_n(t), n \in \{0, \dots, N\}$ are placed onto the grid. Each agent is assigned exactly one cell-fraction: $a_n(t)_1 \in \{i, j, k, l\}$. Furthermore each agent has one of three states ('susceptible', 'infected' or 'recovered'): $a_n(t)_2 \in \{1, 2, 3\}$. To simplify the speech and to support a pictorial representation, the state of the agent is usually called as a prefix of the word agent: e.g. 'infected agent'.

2.3 Simulation

The simulation is performed time discrete with equidistant steps $\{1, \dots, t_{end}\}$. Each time step is split into two phases. First of all the so called infection-phase is performed wherein new-infections and regenerations are calculated. During this phase each agent is addressed once and basically two cases can lead to a state-change:

- If the agent is susceptible and shares a cell with an infected agent, there is a certain probability $P(1,2)$ that the agent gets infected too.
- If the agent is already infected, there is a certain probability $P(2,3)$ that the agent recovers and becomes immune against the disease.

All state changes are done simultaneously.

After this a series of movement rules are applied during so called movement-phase. Hereby all agents are shifted corresponding to certain laws to achieve new arrangements. Basically they are inspired by movement rules of so called lattice gas cellular automata (compare FHP model [10]) and will not be explained here in detail.

A summary of all these ideas is found in Figure 3. It can be seen that even this rather simple model (it is usually very well understood if a picture similar to Figure 3 is given) is very difficult to be described in a formal language especially regarding mathematical formulas, functions and equations.

3 Analysis and Transformations

In order to achieve convergence results and error estimation several stochastic methods known from Markov theory can be used. Concrete a three-dimensional version of the so called diffusionapproximation can be applied. Therefore it is, first of all, necessary to convert spatial influences like neighbourhood and transition conditions into probabilities.

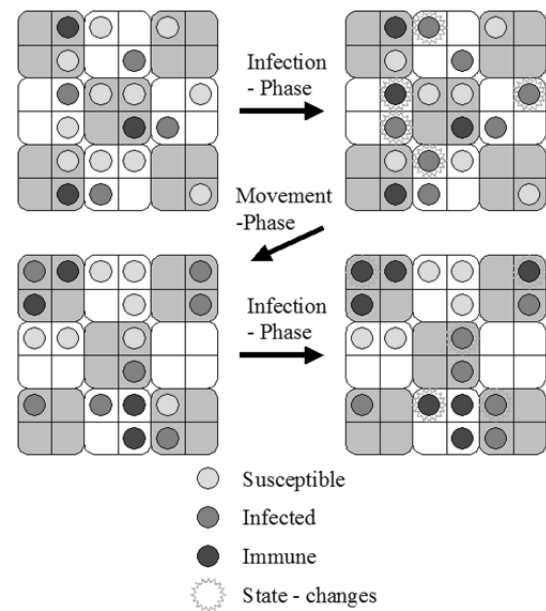


Figure 3: Visualisation of the rules for the SIR Cellular Automaton.

3.1 Extracting probabilities

Defining movement rules in general is basically related to two main ideas: First the motion needs to guarantee a good mixture among the individuals. A bad systematic motion could lead to (usually) unwanted loops or clustering. From this point of view, the ideal movement rule to guarantee a perfect mixture would be a complete stochastic re-arrangement of all agents at the end of each time-step:

$$a_n(t)_1 = (i, j, k, l) \Rightarrow a_n(t+1)_1 = (X, Y, U, V).$$

Though this is not compatible with the second basic aspect of movement, namely that agents cannot cover an infinite range (usually only 1-2 cells) during one time step, a closer look at this aspect is taken. Defining the agitation of all agents completely randomly, basically two independent random processes are responsible for the state of each agent, hence the model is cleanly stochastic and can be described by transition probabilities then. For a fixed index k the probability of agent k changing its state from susceptible to infected can be calculated by multiplying the probability of being placed next to an infected agent during movement-phase and the probability of getting infected by this agent. By simple uniform distribution argument this probability can be calculated to:

$$P(a_k(t+1)_2 = 2 | a_k(t)_2 = 1) = \frac{3\rho\#\{a_k(t)_2 = 2\}}{N} P(1,2).$$

Hereby ρ denotes the population-density $\rho = \frac{N}{4M}$ and model parameter $P(1,2)$ is a fixed infection probability (as described earlier). Analysis of the original model using the original, lattice gas automaton inspired movement rules, shows that a random initial placement of the agents is enough to ensure at least (2) holds.

$$\omega_{1,2} := P(a_k(t+1)_2 = 2 | a_k(t)_2 = 1) = \frac{3\rho\#\{a_k(t)_2 = 2\}}{N} P(1,2) + O(N^{-1}) \quad (2)$$

The Landau-symbol O indicates the asymptotical order of the expression. In this case the asymptotical expression is a result of accidentally clustering of infected or immune agents and strange distributions close to the borders which can lead to other contact probabilities. In both cases they can asymptotically be neglected regarding big numbers of agents (with constant density).

Obviously the result of (2) can, in case of high densities and high numbers of infected agents, end up with a value higher than 1, which in terms of probability theory cannot be correct. Reason for this observation is a basically wrong ansatz wherein the contact probability is sloppily calculated via a simple multiplication of the number of neighboured cells (3) times the probability of observing an infected agent ($p_i := \frac{\rho\#\{a_k(t)_2=2\}}{N}$). Nevertheless the correlation between the correct calculation, seen on the left hand side in (3) and the used calculation on the right hand side is very small and vanishes second order for small densities.

$$1 - (1 - p_i)^3 = 3p_i + O(\rho^2) \quad (3)$$

Surely the probability of regeneration is completely independent of the agents' position:

$$\omega_{2,3} := P(a_k(t+1)_2 = 3 | a_k(t)_2 = 2) = P(2,3).$$

Thus, eliminating the influence of the spatial grid, the agent-vector can be described by an N -dimensional Markov-chain with transition tensor (4).

$$\Omega_i := \begin{pmatrix} 1 - \omega_{1,2} & \omega_{1,2} & 0 \\ 0 & 1 - \omega_{2,3}(\vec{a}) & \omega_{2,3}(\vec{a}) \\ 0 & 0 & 0 \end{pmatrix} \quad (4)$$

$i \in \{0, \dots, N\}.$

As the transition probabilities depend on the sum of all agents sharing the same state the agents itself cannot be described by single Markov-chains which poses the main difference to classic microscopic Markov-models.

Furthermore Markov-theory is going to pave the way to overcome the obstacles between discrete (CA model) and continuous (ODE model).

3.2 Time discrete to continuous

Suppose a given regular, homogeneous, time continuous but space discrete Markov-process $X(t), t \in [0, t_{end}]$ with transition matrix R and three possible states the Kolmogorov equation (5) holds which is in case of sufficient regularity solved by (6).

$$P(X(t) = i | X(0) = j)_{i \in \{1,2,3\}}' = R \cdot P(X(t) = i | X(0) = j)_{i \in \{1,2,3\}} \quad (5)$$

$$P(X(t) = i | X(0) = j) = \exp(Rt) \cdot \vec{e}_j. \quad (6)$$

Therefore a time continuous Markov-process according to $\Omega = \exp(R \cdot 1) \Rightarrow R := \log(\Omega)$ seems to be an appropriate choice to approximate the time discrete one. Surprisingly the first order Taylor approximation: $R := R_i := \Omega_i - \text{Id}, i \in \{0, \dots, N\}$ turns out to be the better choice in our case conserving mean and variance. Errors regarding this approximation can e.g. be calculated using the Taylor series remainder.

3.3 Spatial discrete to continuous

Key observation for the transformations is definitely that the so called observable vector defined by (7) 'counting' all Markov-processes sharing the same state is also a time continuous Markov-process with transition rates ϕ of which only two do not vanish.

$$\vec{o}(t) := \frac{1}{N} \begin{pmatrix} \sum_{i=1}^N \delta_{X_i(t),1}(t) \\ \sum_{i=1}^N \delta_{X_i(t),2}(t) \\ \sum_{i=1}^N \delta_{X_i(t),3}(t) \end{pmatrix} \quad (7)$$

The two non-zero rates are given in (8) and (9).

$$\phi_{N^{-1} \binom{i}{j} \binom{i}{k} + \binom{-1}{1} \binom{0}{0}} = iR_{1,2} = i\omega_{1,2} \quad (8)$$

$$\phi_{N^{-1} \binom{i}{j} \binom{i}{k} + \binom{0}{-1} \binom{0}{1}} = jR_{2,3} = j\omega_{2,3}. \quad (9)$$

These two transition rates, also called jump rates denote the rate for jumps of single agents from state 1 to 2 respectively from state 2 to 3. As the process is continuous and regular no more than one agent can change its state during an infinitesimal time-interval. Therefore all other rates vanish.

Summarizing the performed transformations Figure 4 is given.

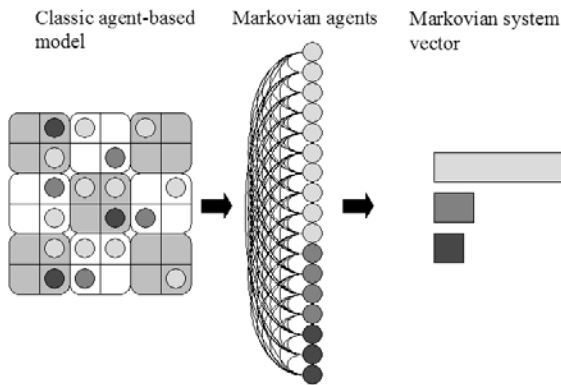


Figure 4: Transformation from CA to Markov Process.

For big numbers of N the observable can take any value between 0 and 1 and thus the probability function $p(t, \vec{o}|\vec{o}_0)$ can be approximated with a continuous and differentiable function taking account interpolation errors. As the observable vector is still markovian the so called diffusion approximation can be performed.

4 Diffusion Approximation

The Kolmogorov equation respectively the closely related master-equation poses the base for a lot of analytical transformations known within physicists under the name “diffusion approximation”. This technique, first time published 1983 by N.G. Van Kampen ([1], [11]), is

based on Taylor-series expansion (named after Kramers and Moyal [12]) and the variable substitution $\vec{o} = \langle \vec{o} \rangle(t) + N^{-1/2} \vec{\xi}(t)$ (compare Itô [13]). This technique is commonly used to describe the temporal development of particle-probabilities. Here only the results of these technique are presented which are valid with respect to asymptotic errors of order $O(\sqrt{N^{-1}})$.

Key observation of these estimations is the derivation of an ODE for the mean value of the observable vector:

$$(\langle \vec{o} \rangle(t))' = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} \langle \vec{o} \rangle_1(t) \omega_{1,2} + \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \langle \vec{o} \rangle_2(t) \omega_{2,3},$$

which, resubstituted, leads to

$$(\langle \vec{o} \rangle(t))' = \begin{pmatrix} -\langle \vec{o} \rangle_1(t) \langle \vec{o} \rangle_2(t) \frac{3\rho}{N} P(1,2) \\ \langle \vec{o} \rangle_1(t) \langle \vec{o} \rangle_2(t) \frac{3\rho}{N} P(1,2) - \langle \vec{o} \rangle_2(t) P(2,3) \\ \langle \vec{o} \rangle_2(t) P(2,3) \end{pmatrix}. \quad (10)$$

Equation (10) can be determined to match the SIR ODE by Kermack and McKendrick. Variance of the observable can be calculated to vanish for $N \rightarrow \infty$ with square-root order. Applied time-scaling $\tau = t/c$ in addition to inverse scaling of the rates $\theta := c\omega$ finally leads to convergence of the aggregated results of the stochastic CA towards the solution of the ODE system when $c \rightarrow 0$ and $\rho \rightarrow 0$. Although the sloppily calculated probability for a state change from susceptible to infected is slightly wrong the error does not disturb the results. Nevertheless the better fitting ODE curves would be (compare with (1)):

$$\begin{pmatrix} S \\ I \\ R \end{pmatrix}' = \begin{pmatrix} -\alpha S(1 - (1 - I)^3) \\ \alpha S(1 - (1 - I)^3) - \beta I \\ \beta I \end{pmatrix}. \quad (11)$$

5 Conclusion

The presented strategy can without loss of generality be extended to other microscopic models and can help to improve understanding of unexpected group behavior in general. The diffusion approximation introduced by Van Kampen [1] was extended and used to show convergence between a classic ODE model and a time discrete Cellular Automaton. Hereby a similar strategy as introduced on the example on economic models in Aoki [14] and on the example of queuing processes in Dohse [15] was used to achieve the transformation from discrete to continuous.

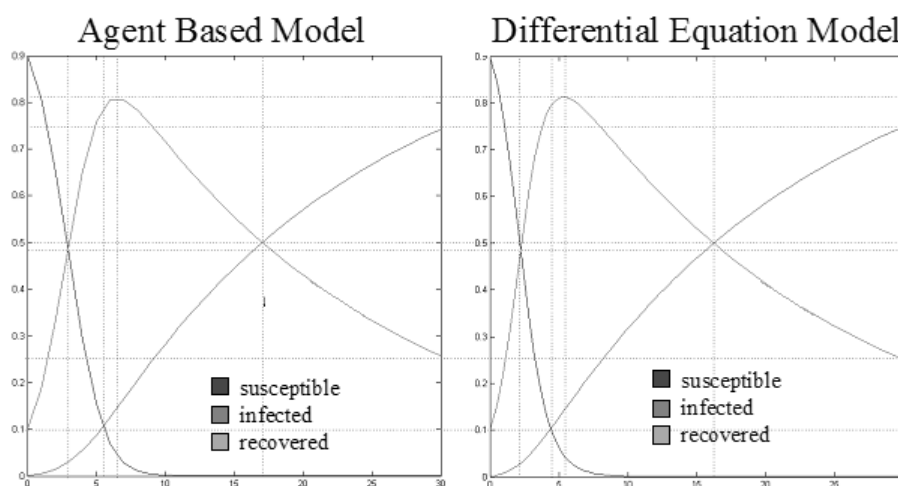


Figure 5: Direct comparison between ODE and CA model.

Figure 5 affirms the success of the explained technique with respect to small errors.

Though the restrictions for analysis of microscopic models using this strategy are very sharp, the validity of the central limit theorem for somehow weak dependent random variables/processes (see e.g. strong mixing [16],[17]), which in general poses the basis for the analysis of aggregated numbers, makes hope for the success of further analysis of aggregated observable vectors of individual-based models. Hopefully further theoretical research can help developing and validating new models on the one hand benefiting from the great flexibility of time discrete microscopic models and on the other hand profiting from fast computation properties and good validation methods of macroscopic models.

References

- [1] van Kampen NG. *The Diffusion Approximation for Markov Processes*. Walter de Gruyter and Co., 1982.
- [2] Kermack WO, McKendrick AG. A Contribution to the Mathematical Theory of Epidemics. *Proc. R. Soc. Math. Phys. Eng. Sci.* 1927; 115(772):700–721
- [3] Moore GE. Cramming more components into integrated circuits *Electronics*. 1965; 38(8):114–117
- [4] Coburn BJ, Wagner BG, Blower S. Modeling influenza epidemics and pandemics: insights into the future of swine flu (H1N1). *BMC Med.* 2009; 7(30)
- [5] Katriel G, Stone L. Pandemic Dynamics and the Breakdown of Herd Immunity. *PLoS ONE*. 2010; 5(3):e9565
- [6] Miksch F, Popper N, Zauner G, Schiller-Frühwirth I, and Endel G. Modelling Spread of Pneumococcal Diseases in Austria: Long Term Behavior and Impact of Vaccination. In *ASIM-Workshop 2009 in Dresden mit integrierter DASS'2009 - Workshop Beiträge*, Dresden, Germany, 2009, vol. Tagungsband (der Fullpaper). P.147–152
- [7] Miksch F, Breiteneker F, Endel G, Popper N, Urach C, Zauner G. Modeling Various Vaccination Strategies against Streptococcus Pneumoniae. *Med. Decis. Making*. 2011; 31(1): E43
- [8] Breiteneker F, Hötendorfer H, Popper N. Modelling and Simulation of SIR-type Epidemic - Classical ODE Approach versus Cellular Automata Approach. In *Proceedings of the 14th International Electrotechnical and Computer Science Conference ERK 2005*, Portoroz, Slovenia, 2005, vol. A, P. 301–306.
- [9] Miksch F, Haim C, Schneckenreither G, Breiteneker F. Comparison of Differential Equations and Cellular Automata for Epidemic Simulation. presented at the ERK - International Electrotechnical and Computer Science Conference, Portoroz, Slovenia, 2013, P. 4
- [10] Hardy J, Pomeau Y, de Pazzis O. Time Evolution of a Two-Dimensional Classical Lattice System. *Phys. Rev. Lett.* 1973; 31(5):276–279
- [11] van Kampen NG. *Stochastic Processes in Physics and Chemistry*, 3rd ed. Amsterdam ; Boston: Elsevier, 2007.
- [12] Moyal JE. Stochastic Processes and Statistical Physics. *J. R. Stat. Soc.* 1949; B(11):150–210
- [13] Gardiner CW. *Stochastic methods: a handbook for the natural and social sciences*. 4th ed. Berlin: Springer, 2009.
- [14] Aoki M. *Modeling aggregate behavior and fluctuations in economics: stochastic views of interacting agents*. New York: Cambridge University Press, 2002.
- [15] Dohse L. *Diffusionsapproximation von Warteschlangensystemen mit Gruppenbedienung*. 2005.
- [16] Rosenblatt M. A Central Limit Theorem and a Strong Mixing Condition. *Proc. Natl. Acad. Sci.* 1956; 42(1):43–47
- [17] Bradley RC. Basic properties of strong mixing conditions. A survey and some open questions. *Probab. Surv.* 2005; 2:107–144
- [18] Schneckenreiter G. Developing Mathematical Formalisms for Cellular Automata in Modelling and Simulation. *Master Thesis, Inst. f. Analysis und Scientific Computing, Vienna University of Technology*, Okt. 2014.

Mathematical Framework for Graphical Simulation Models of Dynamical Systems in SIMULINK

Andreas Körner

Institute for Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; andreas.koerner@tuwien.ac.at

Simulation Notes Europe SNE 23(3-4), 2013, 123 - 128
DOI: 10.11128/sne.23.tn.10203
Received: September 15, 2013; Revised: November 15, 2013;
Accepted: November 30, 2013;

Abstract. This article is about the mathematical description, analysis and background of the simulation environment SIMULINK. This simulation environment is a well-known tool in applied mathematics and a wide range of fields in engineering, mainly control engineering. SIMULINK is driven by control engineering which is recognisable in the block oriented structure as well as in the notation and characterisation of simulation models. This article will give an overview about the mathematics behind the simulation models in SIMULINK and discuss several items. The structure of the article starts in the beginning with the mathematical definitions and context. A relation between the mathematical aspects and the situation in the simulation environment is given.

Introduction

As an introduction of the article the setup of dynamical systems will be given and discussed. Mainly dynamical systems are linked for many people with ordinary differential equations, initial value problems or partial differential equation in more complicated modelling approaches. This link is wrong in general, a dynamical system can be defined without a differential equation. This allows to consider a wider range of dynamical systems, e.g. time continuous and time discrete. In the case of time continuous systems the connection to differential equations can be established as shown in the following steps.

Dynamical System. Assume sets T and $X \neq \emptyset$ with $T \in \{\mathbb{N}_0, \mathbb{Z}, \mathbb{R}_0^+, \mathbb{R}\}$. Furthermore define a mapping

$g: T \times X \rightarrow X$ which satisfies the for $x \in X$ and $t_1, t_2 \in T$

1. $g(0, t) = x$,
2. $g(t_1, g(t_2, x)) = g(t_1 + t_2, x)$.

The triple (T, X, g) is named a dynamical system, in case of $T \in \{\mathbb{N}_0, \mathbb{Z}\}$ it is called time discrete, in case of $T \in \{\mathbb{R}_0^+, \mathbb{R}\}$ it is called time continuous. X is called the state space and g the flow. For easier notation g is redefined to

$$g_t(x) = g(t, x). \quad (1)$$

It is easy to proof that for $t, t_1, t_2 \in T$ the mapping g fulfills

1. $g_0 = \text{id}$,
2. $g_{t_1+t_2} = g_{t_1} \circ g_{t_2} = g_{t_2} \circ g_{t_1}$,
3. $g_t^{-1} = g_{-t}$.

Time Continuous Dynamical Systems. Assume $x: \mathbb{R} \rightarrow \mathbb{R}^n$, $x \in \mathcal{C}^1(\mathbb{R})$ and g continues differentiable. With $x(0) = x_0$ and $x(t) = g_t(x_0)$ let's consider

$$\begin{aligned} \frac{d}{dt}x(t) &= \lim_{h \rightarrow 0} \frac{1}{h} (g_{t+h}(x_0) - g_t(x_0)) = \\ &= \left(\left(\lim_{h \rightarrow 0} \frac{1}{h} (g_h - \text{id}) \right) \circ g_t \right) (x_0) = \\ &= \left(\frac{\partial}{\partial t} g_t \Big|_{t=0} \circ g_t \right) (x_0) = \\ &= \left(\frac{\partial}{\partial t} g_t \Big|_{t=0} \right) (x(t)). \end{aligned} \quad (2)$$

Time continuous dynamical systems with conditions regarding x and g assumed above can be described by

$$x'(t) = f(x(t)), \quad f(x(t)) = \left(\frac{\partial}{\partial t} g_t \Big|_{t=0} \right) (x(t)). \quad (3)$$

1 Model Structure in SIMULINK

Let's consider an arbitrary SIMULINK model, e.g. as illustrated in Figure 1.

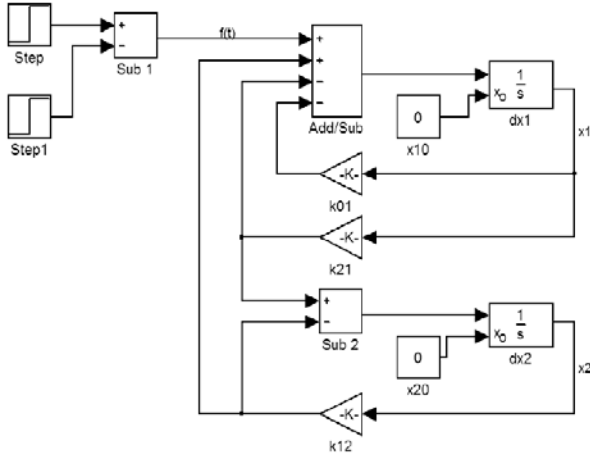


Figure 1: An arbitrary SIMULINK model to illustrate different relations between certain SIMULINK elements.

The model structure shows two issues:

1. input-output relation of each component
2. graph structure defines the topology of the component connection

The following sections will discuss these two aspects, the signal flow graphs as a mathematical environment to describe the interconnection and the input-output relation of the individual blocks to implement different behavior.

2 Graph Concept for SIMULINK Models

Oriented Graphs and SIMULINK-Models. Consider $V = V(G)$ as the set of nodes and $E = E(G)$ the set of edges. The tuple $G = (V, E)$ is called a graph and $e \in E(G)$ an edge. The edge is called oriented if e is represented as an ordered pair $e = \langle v_1, v_2 \rangle$ of nodes $v_1, v_2 \in V(G)$, the starting node v_1 and the end node v_2 . The edge is called not oriented if the edge is represented via a not oriented pair $e = (v_1, v_2)$ of nodes $v_1, v_2 \in V(G)$.

As illustrated in Figure 1 each SIMULINK-model has an orientation. For this purpose the oriented graph is a suitable mathematical environment to represent the

orientation in a SIMULINK-model. Next step is to express the mathematical and technical manipulations in the model, which are represented in a block with certain input and output signals or vectors of signals.

Weighted Graphs and Signal Flow Graphs.

Let's assume $G = (V, E)$ to be a graph and $w: E \rightarrow \mathbb{R}_0^+$. The triple (V, E, w) is called weighted graph and w the weighting function. Let (V, E, w) be a weighted graph and (e_k) a series of edges of the graph. The weight of the series of edges is defined as

$$w((e_k)) = \sum_{i=1}^k w(e_k). \quad (4)$$

Furthermore the distance between two nodes in the graph is defined as the minimum of weights through all series of nodes which are connecting the two discussed nodes. The illustration of a particle of a weighted graph is illustrated in Figure 2.

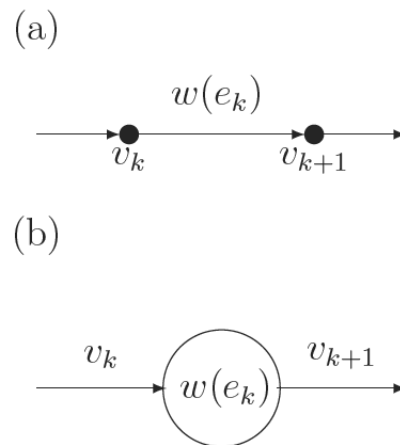


Figure 2: An illustration of a particle of a weighted graph, (a) classical representation in graph theory, (b) representation focusing on signal flow graphs.

The last step in the mathematical environment is the integration of signal flow in the graph concept.

Assume $G_w = (V, E, w)$ as an oriented and weighted graph with the set of nodes $V = \{v_1, \dots, v_k\}$, the set of edges $E = \{e_1, \dots, e_m\}$ and the weighting function $w: E \rightarrow \mathbb{R}_0^+$. Furthermore is $X = \{x_1, \dots, x_k\}$ with $x_i \in \mathbb{R}$ for $i = 1, \dots, k$ defined. The mapping

$$\mu: V \rightarrow X, v_i \mapsto x_i$$

relates each node v_i with a value x_i . The values $x_j = \mu(v_j)$, $x_\ell = \mu(v_\ell)$ of X are connected to w by the equation

$$x_\ell = x_j \cdot w(e_j), \quad (5)$$

if $e_j = \langle v_j, v_\ell \rangle$ is valid. If the node v_ℓ is the end of more than one edge e_j , e.g. $r \in \mathbb{N}$

$$e_j = \langle v_j, v_\ell \rangle, e_{j+1} = \langle v_{j+1}, v_\ell \rangle, \dots, e_{j+r} = \langle v_{j+r}, v_\ell \rangle,$$

than for x_ℓ hold

$$x_\ell = \sum_{s=0}^r x_{j+s} \cdot w(e_{j+s}).$$

The tuple $S = (G_w, \mu)$ is called a signal flow graph.

To have in mind that this mathematical environment is designed for a description for SIMULINK-models this definition of a signal flow graph is not generalized enough to represent more than linear models in SIMULINK. For this purpose the concept has to be re-designed to cover more general model structures.

Generalised Signal Flow Graphs. In the definition given up to now the relation between two nodes x_j and x_ℓ is given by the weighting function w along the edge e_j according to

$$x_\ell = x_j \cdot w(e_j).$$

This limitation in the mathematical description will not allow to include all SIMULINK-models, so the definition need a more general approach. The first step is to generalize the weighting.

Assume S as a signal flow graph. The mapping

$$w_{e_j}: X \rightarrow X, x_j \mapsto x_\ell = w_{e_j}(x_j) \quad (6)$$

defines the so-called generalised weighting function of the signal flow graph S .

This definition of a generalised weighting function allows to cover a wider range of relations between nodes in the signal flow graph. If there are no possibilities of misunderstanding the shorter notation

$$x_\ell = w(x_j)$$

can be applied.

The next step of the generalisation in the signal flow graph concept addresses the subject of signals. The definition up to now didn't cover the flow of signals, only

the relation was oriented which implied a flow. This will be improved in the following extension.

Assume a set $\Omega \subseteq \mathbb{R}$ and a function $f: \Omega \rightarrow \mathbb{R}$.

1. The function f is called a signal, if the characteristics over the time $t \in \Omega$ covers some information of a physical quantity.
2. A signal is called causal, if $f(t) = 0$ for all $t < 0$. More general also $f(t) = 0$ for all $t < t_0$ possible.
3. The set

$$L^2(\Omega) = \left\{ f: \Omega \rightarrow \mathbb{R}: \int_{\Omega} |f(t)|^2 dt < \infty \right\}$$

is called the set of quadratic integrable functions.

4. $(L^2(\Omega), +, \cdot)$ with the composition is a vector space. With

$$\langle f, g \rangle_{L^2} = \int_{\Omega} f(t)g(t) dt$$

a scalar product is defined and its induced norm

$$\|f\|_{L^2} = \sqrt{\langle f, f \rangle_{L^2}} = \sqrt{\int_{\Omega} |f(t)|^2 dt}.$$

For that reason the vector space $(L^2(\Omega), +, \cdot)$ is regarding $\|\cdot\|_{L^2}$ complete and so a Hilbert-space.

This mathematical excursion brings the definition to the second most important generalisation related to the signal space. The Hilbert-space $(L^2(\Omega), +, \cdot)$ with the scalar product $\langle \cdot, \cdot \rangle_{L^2}$, whose elements are signals, is called signal space. Also for this vector space the notation L^2 is used, for causal signals for example the notation would be $L^2(\mathbb{R}_0^+)$.

Finally the introduced concepts are combined in the definition of the generalised signal flow graph.

Assume $G_w = (V, E, w)$ as an oriented and weighted graph with the set of nodes $V = \{v_1, \dots, v_k\}$, the set of edges $E = \{e_1, \dots, e_m\}$ and the weighting function $w: E \rightarrow \mathbb{R}_0^+$. Moreover are $x_i \in L^2(I)$ for $i = 1, \dots, k$ and $I \subseteq \mathbb{R}$. The mapping

$$\mu: V \rightarrow X, v_i \mapsto x_i$$

relates each node v_i with a corresponding signal x_i . The values $x_j = \mu(v_j)$, $x_\ell = \mu(v_\ell)$ of the set X are associated via w referred to

$$x_\ell(t) = w_{e_j}(x_j(t)),$$

if $e_j = \langle v_j, v_\ell \rangle$ is valid.

Is v_ℓ the end node of more than one edge e_j , so for $r \in \mathbb{N}$

$$e_j = \langle v_j, v_\ell \rangle, e_{j+1} = \langle v_{j+1}, v_\ell \rangle, \dots, e_{j+r} = \langle v_{j+r}, v_\ell \rangle,$$

for x_ℓ the relation

$$x_\ell = \sum_{s=0}^r w_{e_{j+s}}(x_{j+s})$$

is valid. The graph $S = (G_w, L^2(I), \mu)$ is called a generalised signal flow graph.

In simulation environments the mathematical description has a disfigurement. In computer the L^2 formulation is not representable. By sampling of the signals the set X can be constructed and a purely discrete graph is used for the numerical simulation.

The interval $I = [a, b] \subset \mathbb{R}$ with $a < b$ and a signal $x \in L^2(I)$ will be considered. Moreover $\mathcal{Z} = \{t_1, t_2, \dots, t_{k-1}, t_k\}$ is a segmentation of I with $t_1 = a$ and $t_k = b$. Is $t_{i+1} - t_i = \Delta t$ for all $i = 1, \dots, k$ the segmentation is called equidistant. Via the determination

$$x(t_i) = x_i$$

it is defined a mapping $\delta_a: L^2(I) \rightarrow \mathbb{R}$, which is called the discretisation of I . This mapping is the connection point between the generalized signal flow graph - the continues mathematical framework - and the regular signal flow graph - the numerical setup for system simulation. The link between is given by the mapping

$$\delta_a: L^2(I) \rightarrow \mathbb{R}, x \mapsto x(t_i)$$

and the construction of the set X by

$$X = \{x_1 = x(t_1), x_2 = x(t_2), \dots, x_k = x(t_k)\}.$$

3 Input-Output Relation

SIMULINK is designed by definition via input-output relations. In linear cases the input-output scheme is mathematical well observed and there are several theories available for the model analysis. In the nonlinear case the mathematical toolbox is smaller or the available methods and theorems are not that global and general as it is in the linear domain. The following subsections will discuss this difference.

3.1 Linear Time Invariant Systems

Time Continuous Systems. Linear time invariant systems are described via a charming mathematical environment - the Laplace transform. This is an integral transform of the form

$$\mathcal{L}(f)(s) = \int_0^\infty f(t)e^{-st} dt. \quad (7)$$

Only signals of exponential order, this means that $s_0 > 0$ and $M > 0$ exist that satisfy for $T > 0$ the condition

$$|f(t)| \leq M \cdot e^{s_0 t} \quad (8)$$

for all $t > T$.

The standard description of LTI-systems in time domain is given by the state space description

$$\begin{aligned} x'(t) &= A x(t) + b u(t), & x(0) &= x_0 \\ y(t) &= c^T x(t) + d u(t), \end{aligned} \quad (9)$$

with $b, c, x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$ and $u, y, d \in \mathbb{R}$. This state space description is equivalent to a ordinary differential equation n -th order. The Laplace transform lead to a system description in the Laplacian domain.

Assume $U = \mathcal{L}(u)$ the Laplacian of the input signal and $Y = \mathcal{L}(y)$ the Laplacian of the output signal of a LTI-system with the state $x \in \mathbb{R}^n$ and $x_0 = 0$. The function $G: \mathbb{C} \rightarrow \mathbb{C}$ which fulfills the relation

$$Y(s) = G(s) \cdot U(s) \quad (10)$$

for each input signal is called the transfer function of the LTI-system.

This mathematical environment offers a structure which is called an algebra. Interpreted with the transfer-function, represented as blocks in the signal flow graph, a particular model can be built by using a certain topology of transfer function blocks. Some basic circuit arrangements are illustrated in Figure 3.

Discrete Time Systems. The discretisation introduced in the section above leads also to the description of discrete linear time invariant systems. The equivalent to the Laplace transform in the discrete time is the Z-Transform. This transformation is as well linear and is applied on series' instead of signals.

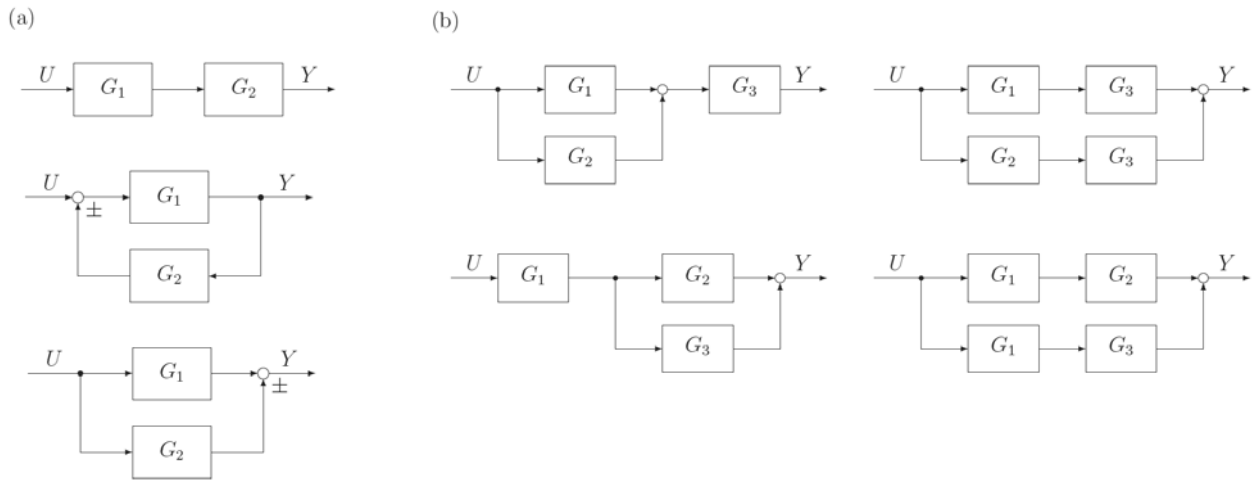


Figure 3: Illustration of Block Circuits: (a) Basic Topologies, (b) Algebra of Transfer Functions.

It is evident that this series are results from the sampling process, following denoted as $(f_n)_{n \in \mathbb{N}}$.

1. If the series $(f_n)_{n \in \mathbb{N}}$ satisfy the inequality

$$|f_n| \leq M\gamma^n, \quad \forall n \in \mathbb{N}$$

and for suitable $\gamma, M > 0$, (f_n) is called from exponential order.

2. $S(\mathbb{N}, \mathbb{R})$ denotes the set of all series which are from exponential order.

3. With

$$\mathcal{Z}(f_n)(z) = \sum_{n=0}^{\infty} f_n z^{-n}$$

a mapping \mathcal{Z} is defined, the so-called z -Transform.

Due to the correlation of the sampling process the z -Transform correspond with the Laplacian. Assume

$$\tilde{f}(t) = \sum_{n=0}^{\infty} f(nT) \delta(t - nT)$$

for $T > 0$ and $\delta: \mathbb{R} \rightarrow \mathbb{R} \cup \{\infty\}$, defined by

$$\delta(t) = \begin{cases} \infty & \text{for } t = 0, \\ 0 & \text{otherwise,} \end{cases}$$

and the condition $\int_{\mathbb{R}} \delta(t) dt = 1$.

Apply on \tilde{f} the Laplacian it results in

$$\begin{aligned} \mathcal{L}(\tilde{f})(s) &= \int_0^{\infty} \sum_{n=0}^{\infty} f(nT) \delta(t - nT) e^{-st} dt = \\ &= \sum_{n=0}^{\infty} f(nT) e^{-snT}. \end{aligned}$$

In summary for $f_n = f(nT)$ and $z = e^{sT}$ it results the formula of the z -Transform.

This relation between the two domains allows the definition of a sampling system by observing a continuous LTI-system with a discretisation and reconstruction interface as illustrated in Figure 4.

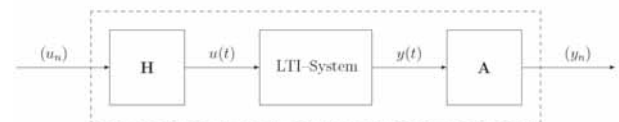


Figure 4: Split-up of a sampling system with sample and hold interface and a corresponding continuous LTI-system.

3.2 Nonlinear systems

Nonlinear Systems has as well a description which is oriented to an input-output relation. It is a generalisation of the state space description for linear time invariant systems, given by two mappings $f: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $g: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ for $m, n, p \in \mathbb{N}$ satisfying the form

$$\begin{aligned} x'(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)). \end{aligned} \quad (11)$$

Not only this systems are addressed when the term nonlinear is used. Also systems with a certain saturation or discretisation effect are covered in this field. Systems like Rate-Limiter, Saturation, Quantizer and Hit Crossing are counting to this systems as well. Nevertheless the general mathematical formulation fits also to this class of systems.

4 Simulation Models in SIMULINK

The introduced mathematical framework for SIMULINK is suitable to describe systems in a formal way. For simulation aspects this description is not helpful or offers optimizing opportunities. But in a modelling context this framework offers a new level in the coexistence of modelling and simulation. The common approach is that the modelling process provides a description of the model and the simulation environment runs the calculation.

The description of the model is normally given in a mathematical framework or an computer science approach. This produce the first problem, there is no common layer where the model can be compared or analysed. The introduced framework offers the possibility to separate a abstract mathematical model, graphical simulation model and the model implemented in the simulation environment by itself. In case of SIMULINK models this approach offers an abstract layer for model descriptions.

References

- [1] Angermann A., Beuschel M., Rau M., Wohlfarth U. *MATLAB–SIMULINK–Stateflow*. 6th Edition. Oldenburg; 2009.
- [2] Scherf H. *Modellbildung und Simulation dynamischer Systeme*. 4th Edition. Oldenburg; 2010.
- [3] Körner, A. *Analyse und Simulation dynamischer Systeme in Simulink mit Web-Interface* [master thesis]. [Institute for Analysis and Scientific Computing, (abrev. State)]. Vienna University of Technology; 2012.

Exploring the Possibilities of Co-Simulation with CATIA V6 Dynamic Behavior Modeling

Peter Smolek^{1*}, Bernhard Heinzl², Horst Ecker¹, Felix Breiteneker³

¹Vienna University of Technology, Institute of Mechanics and Mechatronics; **peter.smolek@tuwien.ac.at*

²dwh Simulation Services, Vienna, Austria

³Inst. of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria;

Simulation Notes Europe SNE 23(3-4), 2013, 129 - 132

DOI: 10.11128/sne.23.sn.10205

Received: Oct. 10, 2013 (Selected ERK 2013 Postconf. Publ.);

Revised Accepted: November 30, 2013;

Abstract. This study shows how to utilize the CATIA V6 Dynamic Behavior Modeling (DBM) software environment for the purpose of co-simulation of physical models. The implementation of a co-simulation using Building Controls Virtual Test Bed (BCVTB) is demonstrated and the pros and cons are discussed. Further the methods for FMI export and import in CATIA are explored with respect to implementing a co-simulation either in CATIA itself or other host programs. The two approaches are displayed by implementing simple examples. Ultimately possible applications for an advanced tool to link 3D geometric data and systems simulation, with the potential to perform co-simulation, are presented.

Introduction

The approach of using system simulations results in new opportunities for integrated virtual product development. Putting the product itself in the center of focus and building the different tools around it allows connecting the different disciplines of development, such as Computer Aided Design (CAD), Computer Aided Engineering (CAE) and system simulation. Co-simulation describes the process of using multiple programs working on the same simulation, sharing data with each other. This grants the flexibility to use specialized software to describe specific parts of the model, for example analytical, databased, static or dynamic components. This enables to examine complex problems like multidomain simulations, stiff systems and many more. In addition it allows the developers to use the software tool best suited for their field of study, collaborating simul-

taneously on the same model. Thereby using different model descriptions and multiple numerical algorithms can be used to study heterogeneous systems.

With the CATIA V6 release, an environment for systems simulation has been implemented – the Systems Dynamic Behavior Modeling (DBM) [1] –which in itself consists of an altered Dymola build. Therefore the CATIA DBM also uses the Modelica standard [2], and the subsequent libraries. The layout has been adjusted to fit the general CATIA logic, the functionality of the Dymola software remains for the most part unaltered. The main difference with respect to Dymola structure is that the DBM is integrated in the CATIA Product Lifecycle Management (PLM) solution, therefore the data is no longer stored locally but on a server. On this remote computer the files of multiple users are stored and the license management is performed. As a result, libraries need to be imported and propagated to the server before use. Modelica files can be imported as well as exported, however the common approach is to work with the DBM projects through the server.

As part of the CATIA PLM, combining physical data with the functionality of the model is realized through the Requirements, Functional, Logical and Physical (RFLP) environment, which can be accessed through the Functional and Logical Workbench seen in Figure 1. It is possible to define logical references, which interact through connections and can be arranged hierarchically. The references in turn can be given a DBM model. At the top of Figure 1 an example of a logical hierarchy is shown. The small icon in the lower right corner of the logical references indicates that a DBM model is implemented. Further, a geometric representation can be linked to the logical reference to represent the physical properties of the logical component, as seen at the bottom of Figure 1.

As it stands, it is not possible to use general 3DParts in the RFLP environment but special 3DRepresentations. This however is announced to be resolved in future releases. The whole simulation can then be run from the Functional and Logical Workbench where all the underlying DBM models are executed and data is exchanged according to the logical connections.

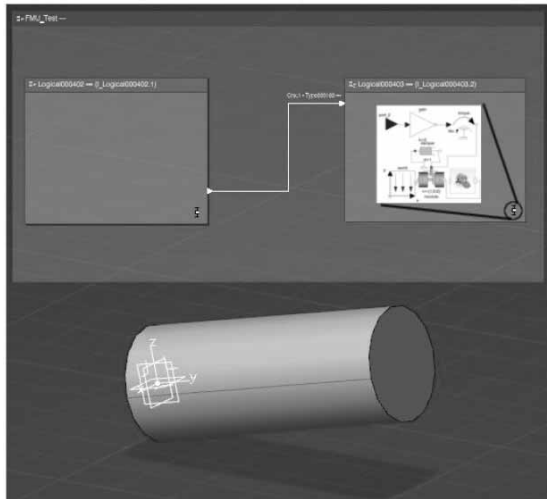


Figure 1: CATIA Functional and Logical Workbench with logical references containing a DBM model and a 3DRepresentation.

1 Co-simulation with BCVTB

To prepare CATIA DBM for co-simulation, a software solution that is compatible with Dymola is chosen. The Building Controls Virtual Test Bed (BCVTB) [3] delivers that in combination with an interface provided in the Modelica Buildings Library [4]. The BCVTB was originally used for buildings simulation, but the ability to link to the general purpose tools Dymola and MATLAB enables the use in other domains. Since this is a Modelica library, importing it into the DBM environment is easily achieved. Models can be built very similarly as in Dymola, since the functionality of the two is basically the same. BCVTB controls the communication between the backbone and CATIA and relies on BSD sockets [5] for runtime data exchange. To minimize startup time, an existing CATIA instance is accessed. The running instance can be contacted through a COM server, which CATIA already runs natively. This can be achieved via a Visual Basic script since CATIA handles macros internally with the same language. Therefore further control over the CATIA instance is possible, especially remote execution of simulation runs is practical.

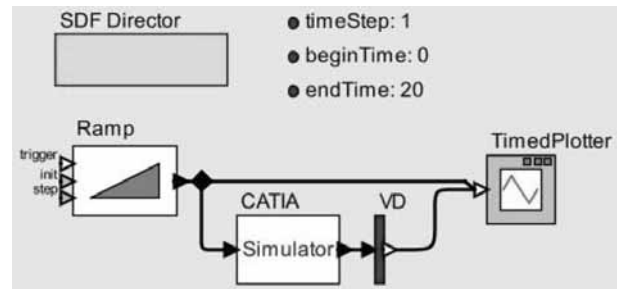


Figure 2: Model in BCVTB; the CATIA simulator block is responsible for the communication with CATIA.

To demonstrate the basic data exchange, a simple model was implemented consisting of a ramp signal, defined in the BCVTB environment. The data signal is then sent to the simulator actor block, which implements the BSD socket communication with the CATIA DBM environment, as seen in Figure 2. The DBM model simply consists of a gain element, which gets the data value from BCVTB, multiplies it by two and returns the result to the backbone. The model is shown in Figure 3, Figure 4 is demonstrating an example simulation run and depicts the input and output signal (timed plotter in Figure 2). The red graph represents the ramp output, the blue is the returned data of CATIA, which is zero per default during the initial data exchange. This simple example is solely supposed to demonstrate the communication between BCVTB and CATIA, no differential equations are solved and the connection to a second simulation software is missing.

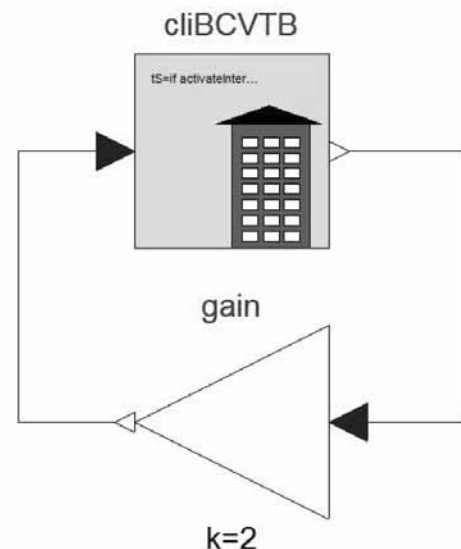


Figure 3: Modelica model implemented in CATIA DBM with the BCVTB Modelica block; the data is received from BCVTB, amplified and returned to BCVTB.

With this method of employing co-simulation, coupling models implemented in different simulation tools with CATIA is achieved. In order to take full advantage of what system simulation in combination with a CAD software offers, it is necessary to couple BCVTB not with the DBM, but with the Functional and Logical Workbench. Unfortunately, with the current release of CATIA V6 that could not be achieved. The RFLP environment cannot execute a simulation successfully with the BCVTB block as a part of it, even though the underlying DBM model in itself can be run. The problem seems to be, that the linked C-functions are not called properly. A solution could not yet be found since CATIA does not allow insight into the internal processes. It is to be expected, that this issue will be resolved in the future, enabling real advantages for co-simulation with CATIA.

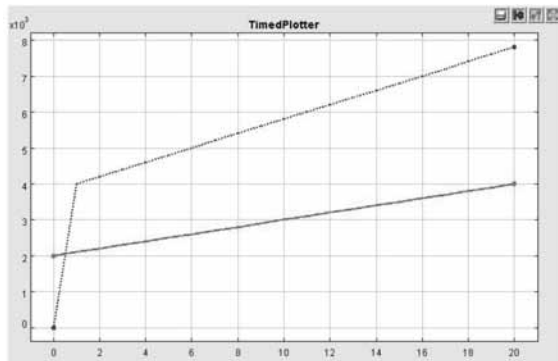


Figure 4: Output of the timed plotter of BCVTB; red: the ramp, blue and dotted: simulator output.

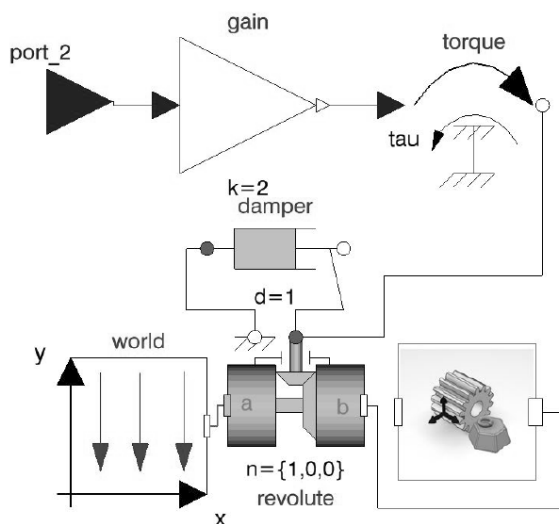


Figure 5: DBM model of a revolvable cylinder; the data from the FMU is received through the input signal port, driving the rotation.

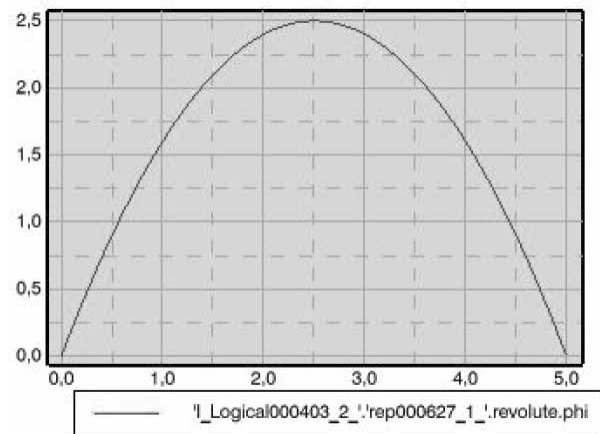


Figure 6: Plotting window of CATIA in the Functional and Logical Workbench; red: the angle of the cylinder over time.

2 Co-simulation with FMI

Another interesting possibility to realize co-simulation with CATIA V6 is by employing the Functional Mockup Interface (FMI) [6]. This standardized interface allows exporting models as Functional Mockup Units (FMU), which in turn can be imported into other simulation tools. It is possible to either export the model alone – the solver of the host then calculates the imported model – or the solver algorithm can be exported with the model, therefore allowing a co-simulation with a modelica host, which is further discussed.

CATIA supports the export and import of FMUs in the DBM as well as the RFLP environment. For this study, the FMUs to be imported are created with Dymola 2014. First of, the DBM Workbench offers the same functionality as does Dymola in its current release. It is possible to import FMUs, link them via appropriate interfaces to the DBM model and run the simulation. The export of a model, as well, works with exceptions unproblematic. The more interesting subject is the FMU export and import of the Functional and Logical Workbench. As it stands, there is no method for exporting models as FMUs. The option that is available is only for exporting already imported FMUs again and therefore is not suitable for using RFLP content externally. The import however is implemented as an extra method of defining DBM models for logical references. This is a convenient way of linking FMUs to logical references, which can be connected to the rest of the logical hierarchy for data exchange.

The second possibility of importing a FMU is to import it into a DBM model. This method is more robust, even though some issues regarding updating an imported FMU do exist. Both methods grant access to co-simulation in the RFLP environment.

For demonstration purposes, a simple setting was chosen. In the RFLP environment, two logical references have been defined (Figure 1). One holds a normal DBM model, containing solely the imported FMU. In the second reference a model of a damped rotation of a cylinder is implemented, see Figure 5. A torque drives the rotation and receives its amplitude from the FMU. Using the appropriate block in the DBM model and linking it with a 3D representation realizes the mechanical arm. Figure 1 shows the logical hierarchy and the cylinder in motion. Figure 6 is an example of a plotter window showing the angle of the mechanical arm plotted over time.

With the presented methods, setting up a co-simulation using FMUs in CATIA is possible. However, the potential is constricted by the existence of a way to export a model via FMI. It is to hope, that with further development of the FMI standard more simulators will add support for exporting FMUs, thus allowing a flexible choice of software for co-simulation.

3 Outlook

As the situation currently presents itself, the unique opportunities of combining CATIA V6 with a system simulator cannot be efficiently utilized for co-simulation yet. Once the software is further developed, the usability will be enhanced and integration of the physical data will enable efficient knowledge management across various domains. Furthermore, whole new options arise from the possibility of not only being able to drain information of CAD data but to alter the geometry data itself.

It is conceivable to build a model of a mechanical systems simulation inside an optimization loop. The optimization, for example a genetic algorithm, can then adapt the geometry to optimize e.g. the distribution of cross sections of a shaft.

One can imagine the possibility to automatically extrapolate a mechanical model from CATIA's Assembly Design, since the program already has the information of the relationship between different components. Therefore realizing dynamic analysis of complex components in a short timeframe would become realistic. Testing a product virtually before prototypes are built has the benefit, that complex interactions of parts are incorporated in the development. This results in reduction of development time and cost of mechanical parts.

Acknowledgement

This contribution is a post-conference publication from ERK 2013 Conference (22nd International Electrotechnical and Computer Science Conference, September 16 - 18, 2013, Portorož, Slovenia), with sessions organized by SLOSIM, the Slovenian simulation society (EUROSIM member). The contribution is a modified publication from the paper published in the Proceedings of ERK 2013 (Zajc B, Trost A, editors. ERK 2013. *Proceedings of the twenty-second International Electrotechnical and Computer Science Conference*; 2013 Sept; Portorož; Ljubljana: Slovenia Section IEEE).

References

- [1] Dassault Systemes: CATIA V6R2013X Portfolio, 2013, <http://www.3ds.com/products-services/catia/portfolio/catia-version-6/v6-portfolio/>
- [2] Modelica Association: Modelica Specification, version 3.3, 2012, modelica.org
- [3] Wetter M. Building Controls Virtual Test Bed User Manual Version 1.3.0, 2013, simulationresearch.lbl.gov/bcvtb
- [4] Wetter M, et al. Modelica Buildings Library, 2012, *Journal of Building Performance Simulation*, (pre-print) 2013
- [5] Wetter M. Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed, 2011, *Journal of Building Performance Simulation*, 4(3):185-203, 2011.
- [6] Modelica Association: Functional Mockup Interface for Model Exchange and Co-Simulation, 2.0 Beta 4, 2012, www.fmi-standard.org

Dynamic Properties of the Non-linear Cholesterol-level Control in the Cell

Aleš Belič

University of Ljubljana, Faculty of Electrical Engineering, Trzaska 25, Ljubljana, Slovenija; ales.belic@fe.uni-lj.si

Simulation Notes Europe SNE 23(3-4), 2013, 133 - 136
DOI: 10.11128/sne.23.tn.10206
Received: Oct. 10, 2013 (Selected ERK 2013 Postconf. Publ.);
Revised Accepted: November 30, 2013;

Abstract. Current research in biomedicine is still mostly based on purely experimental discoveries while modelling and simulation are almost unused. Systems biology and systems medicine try to improve the situation, however, mathematical models are being treated in similar manner as biological models, therefore, their contribution is still limited. In this article we show a simplified model of cholesterol metabolism in the body and analyse its properties with respect to system dynamics. Cholesterol metabolism system shows high robustness on metabolic level, while gene expression regulation is susceptible to external disturbances. The model suggests that it is possible to push similar metabolic systems to alternative stable equilibrium point and cause severe problems (disease) without actually damaging the structure of any of the involved molecular species. This new concept of disease understanding may help solve many disease treatment problems.

Introduction

Dynamical properties of biological systems are still largely neglected. Non-linear high-order systems with a complex network of positive and negative feedback loops can exhibit a wide variety of dynamical behaviours, from critically damped and oscillatory to chaotic. One of the most interesting properties of such systems are multiple equilibrium points. Current opinion on disease onset and progression is mostly associated with some kind of a defect in the metabolic machinery of life, such as genetic mutation, ageing effects, and biological or chemical agents.

However, complex dynamical systems can spontaneously shift to an alternative equilibrium point. A disease is generally defined as persistence of one or more system states out of their normal value ranges accompanied with harmful effects to biological processes. Body temperature is one of the most commonly monitored body states that can indicate on a variety of harmful processes. In linear systems it is not possible to shift from one equilibrium point to another without changing the system parameters. This concept is also widely accepted in biology and medicine, where each disease would want to be described by system parameters such as gene mutations, chemical or isomeric alteration of a molecule etc. On the other hand, nonlinear dynamic systems can spontaneously shift from one equilibrium to another or the shift can be a consequence of environmental effects.

As non-linear systems can persist in any stable equilibrium point, some of the equilibrium points may be harmful to the organism. Many widespread diseases that also cause high costs to the healthcare system [1] such as metabolic syndrome, some cancers, etc., effectively elude all attempts to be classified as genetic disorders of wide range of genes while they may be caused by a complex shift of equilibrium point of several interconnected processes. The idea is further supported by a few cases of spontaneous remissions. The biggest problem is an adequate monitoring of the states in metabolic processes. Each time point requires either blood or tissue sample and can, therefore, not be performed routinely. Brain activity is much easier to monitor, since electroencephalography (EEG) recorders are common equipment and can measure electric activity of the brain with several non-invasive scalp-surface mounted electrodes with sampling frequencies in the range of kHz.

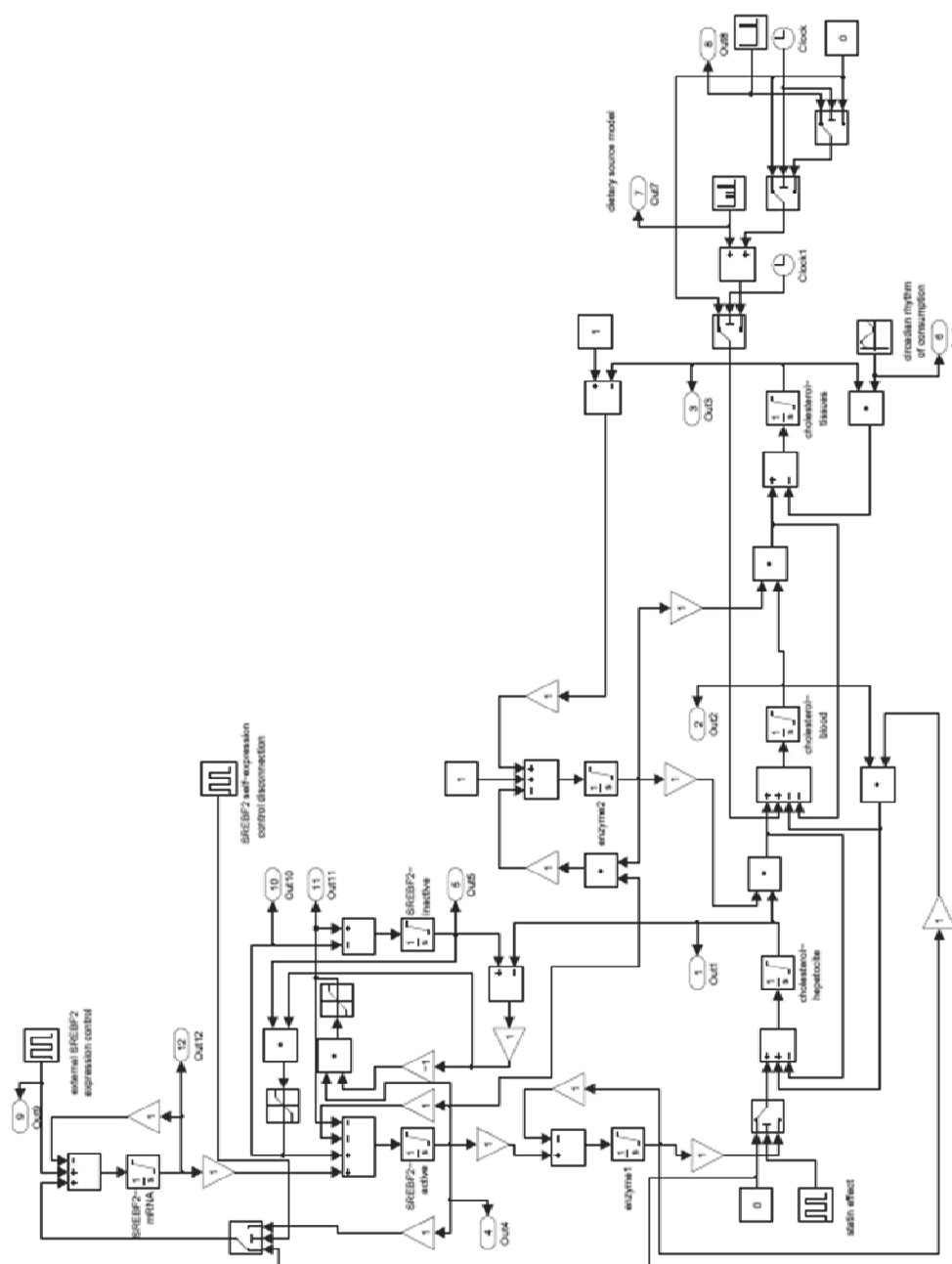


Figure 1: Simulink implementation of the model.

Therefore, it is not surprising that the field of brain activity research adopted system dynamics ideas quite early and currently, epileptic seizures are being interrupted by applying simple vocal disturbance applied at the most convenient time, regarding oscillations around equilibrium point, to push the system toward equilibrium that does not cause seizures [2].

In the absence of suitable monitoring systems, we can only rely on mathematical models when exploring dynamic properties of metabolic systems [3, 4]. In this study we present a cholesterol level regulation mechanism that has some required properties that could result in multiple stable equilibrium points.

1 Simple Model of Cholesterol Regulation in the Body

Cholesterol is one of the most important metabolites in the body, since it is the building block for cell membranes. Any disorder in cholesterol supply can cause severe body malformation of foetus and/or mental retardation since the cells cannot divide properly. Cholesterol is obtained from two sources, one is de-novo biosynthesis and other are dietary sources (Φ_D). Hepatocytes are cells that mainly produce cholesterol (C_{CH}) and supply other organs with the necessary amounts. Hepatocytes also remove excess cholesterol from the blood stream and convert it to bile. Peripheral tissues mostly consume cholesterol (C_{CP}) and are able to signal their needs for cholesterol.

The cell-level regulation of cholesterol is achieved by equilibrium of transcription factor SREBF2 and cholesterol. At high levels of cholesterol SREBF2 is kept inactive (C_{SI}), bound to cell membrane. SREBF2 controls the expression of all cholesterologenic genes (E_1) which act as a valve on cholesterol biosynthesis pathway. As the levels of cholesterol drop, SREBF2 is released and activated (C_{SA}), transported to nucleus and all the cholesterologenic genes are up-regulated. Since positive loop of SREBF2 on its gene expression (C_{SmRNA}) is a part of cascade regulatory system it does not cause unstable system but rather results in PI-controller like system that is capable of very precise control of cholesterol levels in the cell.

Additional expression control (Φ_{EX}) was added to the model to describe possible independent effects of other processes. Cholesterol is transported through blood (C_{CB}) mostly encapsulated in proteins and actively exported in or out of the cell with a series of transporters (E_2) which are controlled internally as well as externally (Φ_2). The following model was used to describe the presented situation.

$$\frac{dC_{CH}}{dt} = E_1 k_1 + C_{CB} E_1 k_2 - C_{CH} E_2 k_3 \quad (1)$$

$$\frac{dC_{CB}}{dt} = C_{CH} E_2 k_3 + \Phi_D - C_{CB} E_1 k_2 - C_{CB} E_2 k_4$$

$$\frac{dC_{CP}}{dt} = C_{CB} E_2 k_4 - C_{CP} k_5 \sin(\omega t)$$

$$\frac{dE_1}{dt} = C_{SA} k_6 - E_1 k_7$$

$$\frac{dE_2}{dt} = \Phi_D + (R_{CP} - C_{CP}) k_8 - C_{SA} E_2 k_9$$

$$\begin{aligned} \frac{dC_{SA}}{dt} = & C_{SmRNA} k_{10} + f((C_{CH} - C_{SI}) C_{SI}) k_{11} \\ & - C_{SA} k_{12} - f((C_{SI} - C_{CH}) C_{SA}) k_{13} \end{aligned}$$

$$\begin{aligned} \frac{dC_{SI}}{dt} = & f((C_{CH} - C_{SI}) C_{SA}) k_{13} \\ & - f((C_{SI} - C_{CH}) C_{SI}) k_{11} \end{aligned}$$

$$\frac{dC_{SmRNA}}{dt} = C_{SA} k_{14} - C_{SmRNA} k_{15} + \Phi_{EX}$$

$$f(x) = \begin{cases} 0; & x < 0 \\ x; & x \geq 0 \end{cases}$$

In the Equation 1 the k_i represent the model parameters. Since we have no realistic information on the values of the model parameters, because the models is so drastically simplified that they cannot be related to any specific reaction and because systems dynamics of the process is so poorly characterised, the values were chosen such that the system has at least one stable non-oscillatory nonzero equilibrium point and reasonably quick transient phenomena. Model simulation and analysis was performed in Matlab/Simulink 2009b (Mathworks Inc., Natick, Mass. USA). The simulink implementation of the model is presented in Figure 1.

2 Simulation Experiments

With simulation experiments the effects of several conditions on cholesterol level control in the hepatocytes were tested. Since the control algorithm is a type of ratio control, it is always possible that concentration level set-point could shift, as long as the ration of the involved species stays within range defined by a controller design. Therefore, we tested several conditions that pushed the controlling sub-system to saturation. It was shown in [5] that PI-like control controls the cholesterol levels in the cell, however, the character changes when the system is overloaded. The most important goal was to test if such high disturbances could push the system into a new equilibrium point.

The following conditions were tested: extreme increase in cholesterol uptake from dietary sources, no cholesterol diet, complete blocking of cholesterol biosynthesis (statin activity), elevated demand on cholesterol in peripheral tissues, external influence on SREBF2 expression, and breaking of SREBF2 feedback loop on its expression. The simulation results are presented in Figures 2 to 7. The first simulation experiment was intended to test the capacity of the regulatory mechanisms. The dietary uptake of cholesterol was increased 30 times for a short time, which simulated a large feast. After the large meal, levels decreased to normal values, although the control system was operating in saturation for a while (see Figure 2).

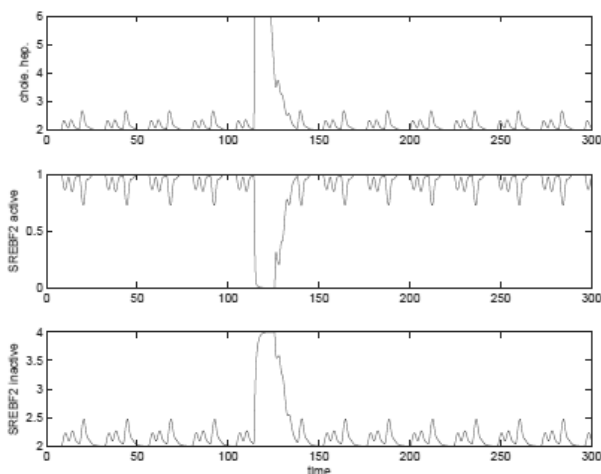


Figure 2: Simulation of large dietary uptake (30 times larger than normal uptake).

The second simulation experiment was designed to test low cholesterol diet circumstances. As this is normal situation for most animals, the system replaces all the dietary cholesterol with internally synthesised and normally functions (see Figure 3).

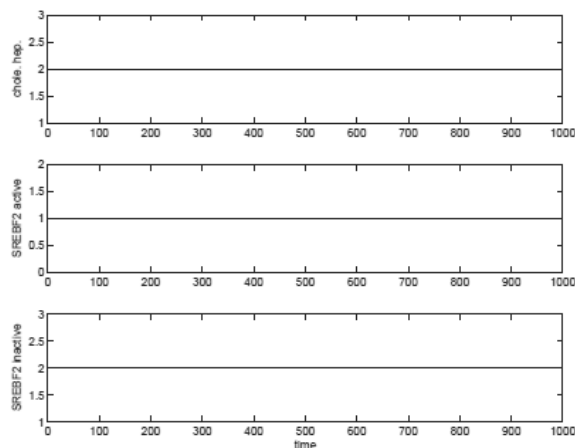


Figure 3: Simulation of no cholesterol diet.

The third simulation experiment explored the influence of statins, cholesterol lowering drugs on cholesterol biosynthesis. Statins are designed to block cholesterol biosynthesis by binding to one of the important synthesis enzymes. Complete blocking of cholesterol biosynthesis resulted in complete depletion of cholesterol in hepatocytes, however, as blockade was lifted, the levels were completely restored (see Figure 4).

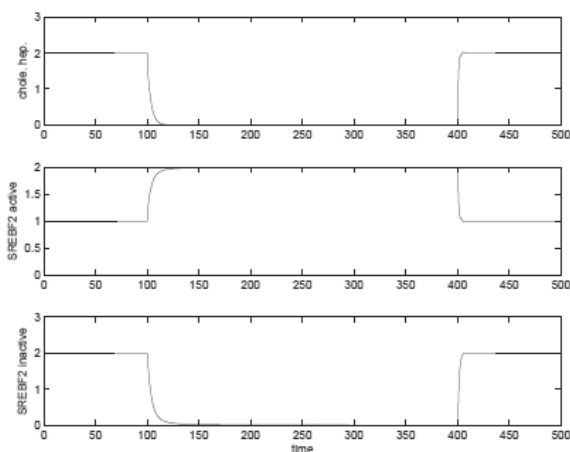


Figure 4: Simulation of complete biosynthesis blocking.

The fourth experiment targeted the influence of increased demand on cholesterol levels in hepatocytes. The demand was increased to double value of the usual demand. Only minor reduction in cholesterol levels in hepatocytes was observed (see Figure 5).

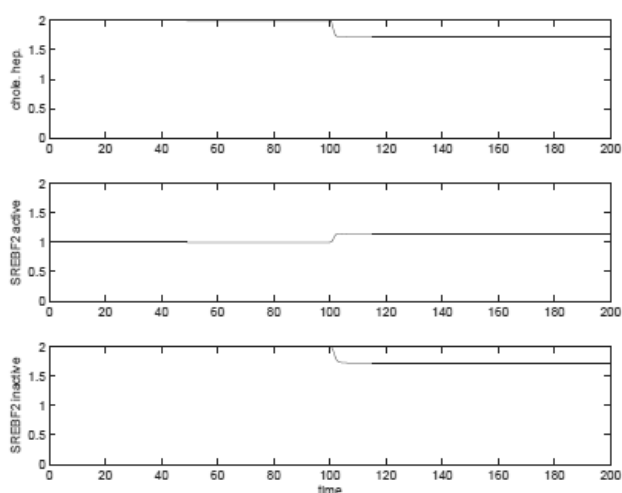


Figure 5: Simulation of increased demand (2x) in peripheral tissues.

The fifth experiment was designed to test the gene expression of SREBF2 influence on the cholesterol regulation. A short external intervention causes long term disruption of cholesterol level control by changing its set-point (see Figure 6).

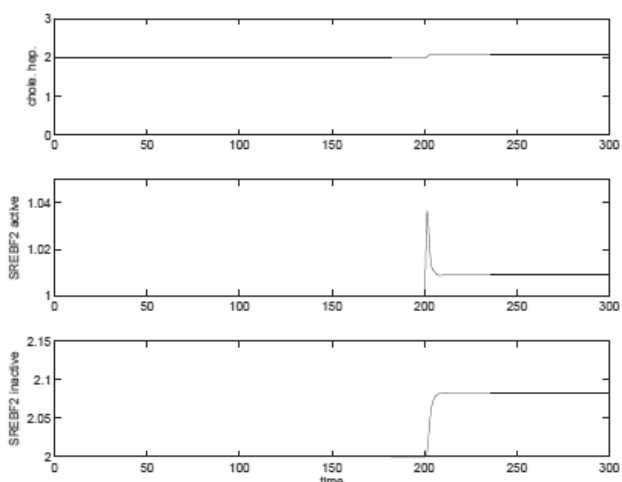


Figure 6: Simulation of external influence on SREBF2 expression (1% of normal expression flux, short pulse).

The sixth simulation experiment also targeted SREBF2 expression regulation, however, this time feedback from SREBF2 protein was disconnected. Again, this resulted in set-point disturbance. As the disconnection was long enough, active and inactive SREBF2 were depleted and the system could no longer regain its functionality (see Figure 7).

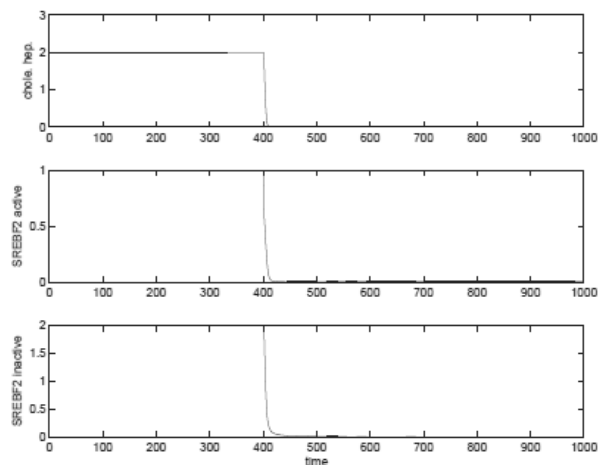


Figure 7: Simulation of a short break of the internal control loop for SREBF2 expression.

3 Discussion

Simulation experiments show interesting property of the metabolic systems. They are extremely robust to any disturbance on metabolic level and their regulation can compensate for more than 10-fold changes in uptake or consumption. Sigmoid type non-linearities arising from products of variable in enzyme reaction models reduce control problems at high fold changes of levels since they have very low attenuation at high values and thus prohibit the propagation of such spikes through the system. Although we expected problems when system would be pushed to saturation it completely recovered to original equilibrium point.

Saturation of the metabolic pathway means that further elevation of concentration cannot significantly elevate the flux through the network which can be beneficial in cases of uptake and quite problematic in cases of elimination since we can expect accumulation of metabolites in tissue, however, regulation system remains intact. The protein and gene expression levels seem to be more sensitive to perturbations. Any interference with gene expression regulation results in equilibrium point change that is not reversible.

However, such external gene expression controllers exist, since regulatory network is also full of feedback-mechanisms that take care of flux distribution and rational energy consumption of the body. They may not be entirely independent of cholesterol metabolic pathway, however, we can expect that in some specific conditions controller set-point might actually change which would cause system wide problems.

4 Conclusion

Although the presented simulation results do not prove the concept of system dynamics as one of the reasons for entering disease state, they provide some important insights into metabolic systems functioning. Almost all regulatory mechanism operate in ration control mode, which makes them potentially susceptible to spontaneous setpoint changes. As shown in this paper, it takes only short term disturbances at the right place of the system and long-term consequences can occur. This new concept of disease analysis and treatment may add a new piece to our understanding of the diversity of biological processes.

Acknowledgement

This contribution is a post-conference publication from ERK 2013 Conference (22nd International Electrotechnical and Computer Science Conference, September 16 - 18, 2013, Portorož, Slovenia), with sessions organized by SLOSIM, the Slovenian simulation society (EUROSIM member). The contribution is a modified publication from the paper published in the Proceedings of ERK 2013 (Zajc B, Trost A, editors. ERK 2013. *Proceedings of the twenty-second International Electrotechnical and Computer Science Conference*; 2013 Sept; Portorož; Ljubljana: Slovenia Section IEEE).

References

- [1] Glock B, Zauner G, Einziger P. Compartment modelling of obesity in inhomogeneous populations: Problems and alternative approaches. In: Snorek M, Bulk Z, Cepek M, Drchal J, editors. *EUROSIM 2010*. Proceedings of 7th EUROSIM Congress on Modelling and Simulation; 2010; Prague, Czech. 2: P. 777–783
- [2] Rajna P, Lona C. Sensory stimulus for inhibition of epileptic seizures. *Epilepsia*. 1989; 30(2): 168–174
- [3] Logan JA, Kelly ME, Ayers D, Shipillis N, Baier G, Ray PJR. Systems biology and modeling in neuroblastoma: practicalities and perspectives. *Expert Rev. Mol. Diagn.* 2010; 10(2): 131–145
- [4] Al-Nuaimi Y, Goodfellow M, Paus R, Baier G. A prototypic mathematical model of the human hair cycle. *Journal of Theoretical Biology*. 2012; 310: 143–159
- [5] Belič A, Ačimovič J, Naik A, Goličnik M. Analysis of the steady-state relations and control algorithm characterisation in a mathematical model of cholesterol biosynthesis. *Simulation modelling practice and theory*. 2013; 33: 18–27

Comparison Open Modelica – Dymola for Power Plant Simulation

Elena T. Aparicio¹, Erik Dahlquist^{1*}, Hailong Li¹, Peter Fritzson², Per Östlund²

¹School of Sustainable Development of Society and Technology, Mälardalen University, Högskoleplan 1, 721 23 Västerås, Sweden; *erik.dahlquist@mdh.se

²Programming Environment Laboratory (PELAB), Linköping University, SE-581 83 Linköping, Sweden

Simulation Notes Europe SNE 23(3-4), 2013, 139 - 146

DOI: 10.11128/sne.23.tn.10207

Received: March 10, 2013 (Selected SIM 2012 Postconf. Publ.);

Revised Accepted: November 10, 2013;

Abstract. In this paper a comparison is made between OpenModelica and Dymola for a simulation model of a power boiler. The similarities and differences are presented. Dymola has the advantage of having a more elaborate user interface and solver, but the OpenModelica user interface and solver has improved very much during last few years. The advantage of OpenModelica is that anyone can use the models without having to pay high license fees, something that is of significant interest when installations are made in industries. In many ways a combination of the two is advisable, where Dymola can be used for application developments and later OpenModelica can be used in the actual installations. It has been seen in this application for a CFB boiler that it is easy to use the same model in both environments without any modifications. Still, the solver for OpenModelica is not as powerful as for Dymola, which may be a problem for on-line applications for larger models, while no problem for small models.

Introduction and Literature Review

Energy conversion system models are usually complex models which require a high structured programming language. Generally dynamic models are preferred in order to reach a deeper understanding of the process. Many studies have pointed out Modelica as a straightforward object oriented language developed for modeling of large physical systems [12], [13]. Dymola is an engineering simulation tool using Modelica language [20].

OpenModelica is an alternative tool based on open source code. Both Dymola and OpenModelica include several Modelica libraries (MSL – the Modelica Standard Library) from Modelica Association in their distributions [22]. There are also Modelica libraries for e.g. power plants [3] and other energy conversion systems [25]. In [4] a dynamic model of a biomass-fired-power-plant is presented. Jansson et al (2008), Järvensivu (2001), Hauge et al (2005), Mercangöz and Doyle (2006) and Dhak et al (2004) have shown how model based control can be implemented and Karlsson et al (2009) how models can be adapted to compensate for fouling. Sandberg et al. have modeled the actual fouling. Mälardalen University has developed their own Modelica models for pulp and paper industry and power plant applications, as well as for gasification.

There is also commercial equation-based modeling and simulation software similar to Modelica used for process industry. This is named gPROMS and was developed by Imperial College in London under leadership of professor Costas Pantelides. Model structure is similar, but at least some years ago there were only two solvers available for gPROMS. These have a varying time step depending on how fast the dynamics of the process is. OpenModelica and Dymola have solvers for both varying time step and fixed time step. Dymola have additional functions compared to OpenModelica, but OpenModelica is adding new functions continually as well.

1 Methodology

A semi-dynamic on-line application of the proposed simulation approach has been used in the simulation of a CFB boiler at Mälarenergi in Västerås as well as at Korsnäs pulp mill in Gävle in tests during 2010 - 2011.

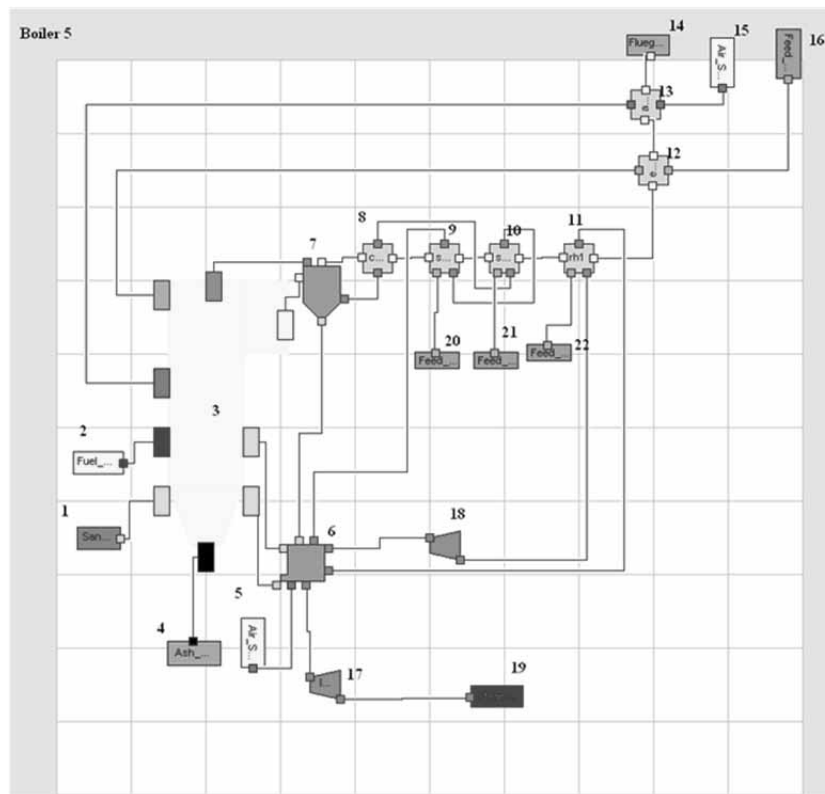


Figure 1: View of the CFB boiler in Dymola interface. Showing boiler 5 at Mälarenergi CHP plant.

Here the connection between the DCS and the simulation model was established with Simulink. Modelica models were used after compilation allowing better control of the signal processing between simulation and the process database. The project at Mälarenergi was financed by Värmeforsk (and power companies) and the project at Korsnäs by KKS (and Korsnäs and ABB).

The CFB boiler model developed by Mälardalen University (MDH) includes the combustion section, the water/steam system and the exhaust gas train. The model is validated towards real plant data and is capable to successfully predict operation performance. A more detailed description of this model can be found in Sandberg *et al.* (2011).

During 2010 and 2011 the model was used on-line at Mälarenergi AB for diagnostic purposes. The plant is a combined heat and power plant (CHP). Boiler 5 has been modeled. This is a 180 MW_{th} biomass fired CFB boiler.

The components of the Modelica/Dymola model can be seen in Figure 1. The components are as follows:

1. Air flow to boiler,
2. Fuel flow and composition,
3. Boiler/reactor,
4. Ash flow,
5. Air flow to Intrex,
6. Intrex – fluid bed/G-valve where solids come down from cyclone and is heat exchanged towards steam,
7. Cyclone separator where larger particles are separated but also gas cooled,
8. The heat exchanger in the walls of the cyclone where gas is cooled towards steam,
- 9- 11. Steam heating, gas cooling heat exchangers,
12. Economizers where feed water is heated (and evaporated),
13. Air pre-heating,
14. Exhaust gas flow and composition,
15. Air flow to preheater,
16. Feed water flow,
- 17-18. Steam flow and temperature/pressure to turbine,
19. Electric power produced,
- 20-22. Feed water injection to heat exchangers.

There are three mass inventories in the model: (i) gas in the boiler including the bed material, fuel etc.; (ii) bed material and gas in the so called Intrex, a bubbling bed below the cyclones where separated sand is fluidized and cooled in two super heaters before the sand is reentering the CFB boiler; (iii) the steam system with water and steam.

Temperature, gas composition and flow rates are measured all the way through the boiler and exhaust gas train and in the steam system. These measurements are then compared to the values predicted from the simulation using the same input data. This includes fuel flow, fuel composition, air flow and feed water flow to the steam drum. Unfortunately the fuel composition has not been able to measure; if moisture content varies the impact will be significant. Measurement of moisture and higher heating value (HHV) of the fuel we want to include in the future research. Use of NIR (near infrared) measurement on-line is a tool we develop together with Bestwood. On-line measurement of moisture is already installed at Eskilstuna CHP, but the development of models for HHV is still ongoing at our lab in Västerås.

The equations used are primarily stoichiometric calculations of how the biomass is converted through combustion, giving adiabatic temperature and cooling through heat transfer and through transport of material from the boiler combustion zone. The mass in the bed inventory as a function of time is given from:

$$\frac{\partial m_{inventory}}{\partial t} = \sum m_{i,in} - \sum m_{i,out} \quad (1)$$

where $m_{i,in}$ is the mass input flow of each single component of the composition vector $i=(C,H,O,N,CO_2,H_2O,NO_2,ash)$ and $m_{i,out}$, is the corresponding output flow. The change in concentration of each component is given by c_i in the bed inventory:

$$\frac{\partial c_i}{\partial t} = \frac{\sum (c_i - m_j)_{in} - \sum (c_i \cdot m_k)_{out}}{m_{inv.}} \quad (2)$$

where j runs through all incoming flows and k all outgoing flows of the inventories. Except the bed inventory we also have one inventory for the Intrex and one for the steam system. The steam system has only water and steam components, while the Intrex has the same components as the bed. The temperature $T_{inventory}$ in the inventory is calculated from the energy balance:

$$\frac{\partial T_{inventory}}{\partial t} = \frac{(\sum T_j \cdot c_{pi} \cdot c_i \cdot m_{j,in}) - (\sum T_j \cdot k \cdot c_{pi} \cdot c_i \cdot m_{k,out})}{m_{inventory} \cdot (\sum c_i \cdot c_{pi})} + \frac{\Delta H - U \cdot A \cdot (T_{inventory} - T_{outside})}{m_{inventory} \cdot (\sum c_i \cdot c_{pi})} \quad (3)$$

Here ΔH (enthalpy) is the energy released during combustion and U is the overall heat transfer coefficient, A is the heat exchanger area and $T_{outside}$ the temperature at the other side of the heat exchanger surface (steam temperature vs. exhaust gas temperature), c_{pi} is the heat capacity for each component i . The correlations describing the change in each single component is also included in the model. Carbon, C, in the biomass is combusted to CO_2 , and the hydrogen is forming H_2O . Oxygen, O, in the fuel is used for the combustion aside of the oxygen in the air. N, in the fuel is assumed oxidized to NO_2 partly, as a function of oxygen surplus and temperature. Separation of sand is performed in cyclones and cooling in heat exchangers with gas to gas, gas to steam or gas to water transfer. We have not included inventories in the heat exchangers as the residence time is very short.

The combustion/gasification processes are modeled as an extension of equation (2):

$$\frac{\partial c_i}{\partial t} = \frac{(\sum (c_i - m_j)_{in} - k \cdot [c_i]^a) - \sum (c_i \cdot m_k)_{out}}{m_{inv.}} \quad (4)$$

where k is a reaction constant and a an exponent giving the non-linearity of the conversion. For components being removed c_i is decreasing while for those being created c_i is increasing.

So far we have primarily been running the models as semi-steady state, but we want to include the full dynamics as this has a significant importance for both the diagnostics and the model based control, especially to meet varying moisture and HHV of the fuel.

2 Model Validation and Results

The boiler simulation model has been verified towards process data. In Table 1 a comparison between measured and predicted data from the simulation for full load and partial load is presented. We have just included these situations for average load conditions to give a picture of how the model has been tuned towards process data. It should be noticed that the measurements in the boiler are not ‘the truth’.

In reality the temperature at different positions in the boiler varies a lot. We thus have tuned the model towards reasonable averages measured in the on-line positions at different positions in the boiler.

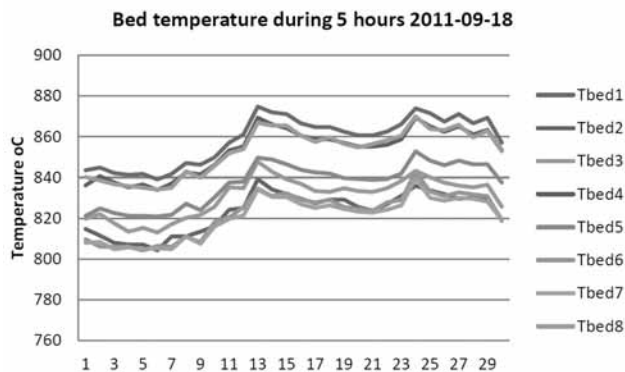


Figure 2: Bed temperature at the same elevation for eight meters.

The absolute value though may vary many hundred degrees between the wall and the center of the boiler at the same elevation, according to measurements we have made but not published yet.

Part load (July 5, 2011) at 6.2 kg/s fuel, 30.1 kg/s			
Variables	DCS	RH	Error%
Steam temperature after HPSH2 (°C)	434	439	1.0
Fluegas temperature after cyclone (°C)	550	566	2.9
Fluegas temperature before cyclone (°C)	551	576	4.6
Steam temperature after cyclone (°C)	366	353	-3.6
Full load (September 2011) 16.5 kg/s fuel, 79.8 kg/s air, 48.8 kg/s feed water			
Steam temperature after High Pressure Super Heater 2 (°C)	494	488	-1.2
Bed temperature (°C)	833	879	5.2
Fluegas temperature before cyclone (°C)	758	757	0
Steam temperature after cyclone (°C)	385	379	-1.5

Table 1: Measured process data (DCS) compared to predictions made with the simulation data.

As can be seen the absolute error varies between 0 and 5%. With the proposed approach with a new methodology for adaptation of models to process data we will improve the accuracy significantly. This includes correlating the on-line measurements to manual measurements of the temperature profile at specified positions.

In Figure 2 we can see how the difference between different sensors in the boiler at the same elevation varies in time over five hours. As can be seen they change level simultaneously, but the absolute value varies with approximately 40 °C in this case.

For our purpose of using the model for diagnostics, decision support, maintenance on demand and model based control still it is the variations we want to measure and not primarily the absolute values of temperature etc. Then it is OK just to have reasonable data to fit the mass and energy balances.

3 Comparison of Dymola and OpenModelica from a User Perspective

OpenModelica and Dymola, as well as gPROMS all are equation-based and object oriented and all have simultaneous solver approach. OpenModelica and Dymola as well as some other vendors support the Modelica model standard, whereas gPROMS has its own model format.

You configure the complete simulation problem as a big equation system, which is solved simultaneously. In reality the equation system is split into smaller systems automatically, to get faster convergence, but it is a big difference compared to the earlier simulation systems with sequential solvers. With the simultaneous solver we can really correlate different sensor along the process to each other, which gives us the possibility to get better diagnostics.

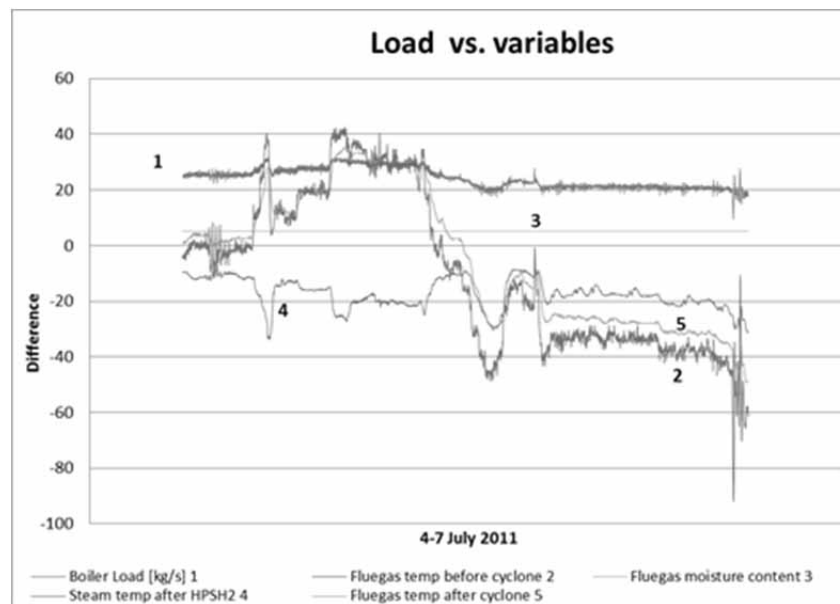


Figure 3: Difference between simulated and measured data for five variables in boiler 5 July 4-7 2011.

A key task is then to formulate equations covering all modeling tasks, where the number of equations and variables must be the same for each sub-model. This can seem like a simple task- just do it. In reality still it is not self- evident what is a variable or a parameter and thus it may be complicated to know ‘where you are’. We have noticed that it is even stricter with this in OpenModelica than in Dymola. A major problem thus for the model developer is to both get the right number of variables and equations, as well as to know that it is actually the RIGHT equations we have formulated. Sometimes you can note that you really would need one extra equation, but it is definitely not clear which you can take away instead! A preliminary (not released) version of OpenModelica contains a debugging tool that is addressing many of these problems, explaining which equations give rise to selected computed variables, and providing on-line interactive stepping, breakpoints, browsing of variable values, etc. [23]. See also OpenModelica on-line simulation [1].

Because of the structure with simultaneous equation solver it is very difficult to do debugging. In principle you can work in a structured way so that you make a smaller system and then connect several smaller systems to one big one. This is in principle simple, but in practice not that easy. This is especially tricky when you want to use simulations on line, and start developing a steady state model with fixed values as input. When you then connect dynamic input signals it may be quite tricky to really get the simulator to work, as the number of variables and equations suddenly are incorrect. At MDH we have been using a link to process data bases making Modelica models as compiled objects into Simulink/Matlab. When you then debug the actual Modelica model and all is working, it may still be tough to get the compiled model to work when you make the linking to the on-line signals. It thus would be interesting in the future to have some kind of automatic function between on-line use and off-line use, where you just run with constants or simple general functions to generate input to the simulations.

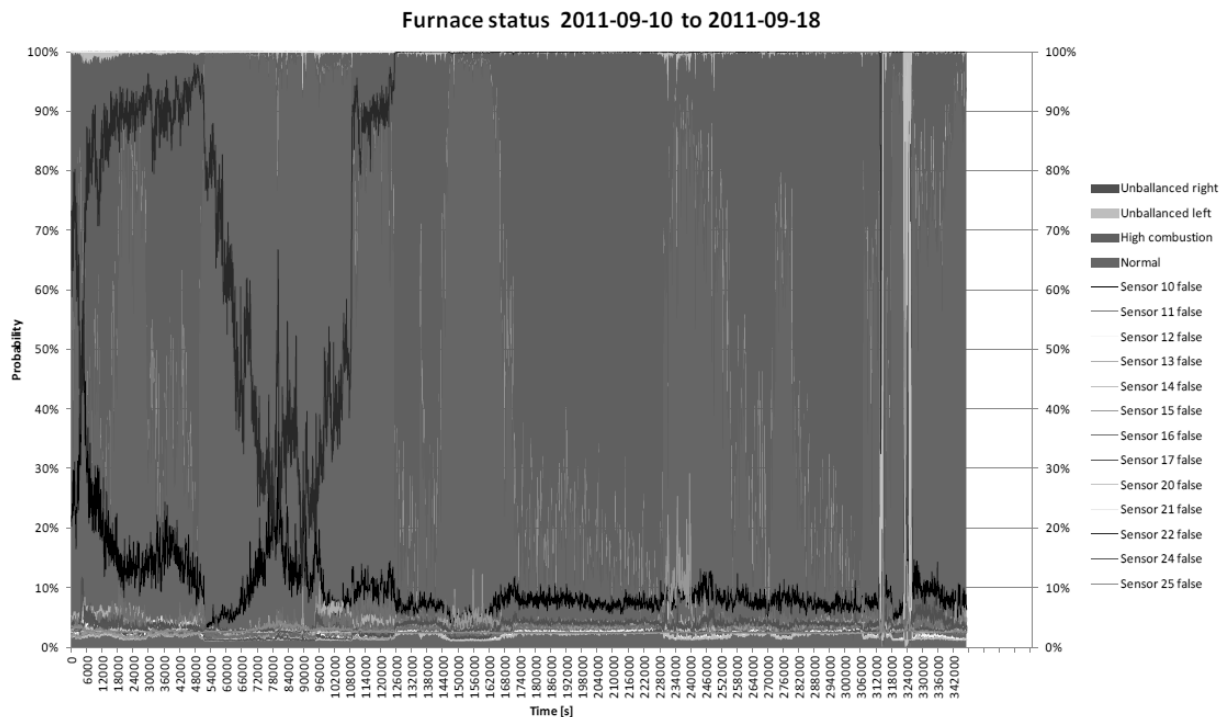


Figure 4: Bayesian Net of differences between simulated and measured data from Boiler 5 during the period September 10 to September 18, 2011.

Concerning the transfer between Dymola and OpenModelica we have noticed that the new version of OpenModelica actually can directly convert a Modelica Dymola model into one that can be simulated in OpenModelica and reverse. Still, where we see a warning in Dymola it may be a real fault in Open Modelica, as this has stricter definitions of what is accepted, more in accordance with the Modelica language standard. This is both good and bad, but is definitely a problem as you have even more difficulty to debug them in Dymola. A recommendation thus would be to have less strict rules for this, so that you have a chance to compile and test-run your code, before you do changes in the code. In future versions of Open Modelica there will be a warning instead, to make it easier to debug.

Concerning accuracy we have not seen any difference between OM and Dymola, but sometimes the solutions have taken longer time with OM. As the solvers for OM are developed continuously we hope this will be less in the future, but it is of course a “moving target”.

An example of results from the on-line application of simulation towards measured process data is seen in Figure 3.

Here we can see that the load (curve 1) was going down after approximately July 5. We can see that the temperature predicted by the model first goes up in the flue gas compared to the measured temperature before the cyclone as well as after the cyclone, but as the load is lowered this swings to the opposite. The flue gas moisture signal (3) is constant all the time, and the variations are so small that it can be questioned if the sensor is giving correct values. The steam temperature in the corresponding heat exchanger High Pressure Super Heater 2 (HPSH2, curve 4) is moving in the opposite direction during the first two days, but is following the same direction during the last two days. It is not obvious what the reason is, but may be related to fouling of the heat exchangers during the first two days with a reduced heat transfer rate as a result, while the heat transfer is better during the last two days. Still, this is not verified. The difference in fluegas temperature after cyclone (5) is first increasing, but later decreasing again. We believe it is due to temporarily combustion in the cyclone, which should not take place under normal conditions.

We also have sent the differences to a Bayesian Net, where different faults can be seen visually. An example of this is seen in Figure 4.

Figure 4 indicates problems with high combustion, feeding problems of fuel on right (violet) and left (green) side, and also deviation between predicted and measured values for temperature meters in the bed above the feed at the left respectively right side of the bed. Variables to the right: Unbalance right, unbalanced left, high combustion, 6 temperature sensors in the bed, 7 temperature sensors above the bed.

4 Discussion and Conclusions

From the tests we have performed at Mälarenergi AB with on-line simulation combined with process data measurements we can conclude that a number of different type of faults have been possible to determine. The combination with BN, Bayesian Nets, is a feasible approach as experiences of different kind can be combined to generate a good decision support for process operators. Still, it will take time to make the system automatically adaptive with new experiences, although the goal will be to reach this. It is principally possible to develop a model in e.g. Dymola and then convert it into an Open Modelica model just by opening it in this environment, or the reverse. This is advantageous as it then is possible to make use of the advantages in the two platforms. This is very positive for the users of Modelica language. It is thus possible to use new functions developed in one of the platforms also for models developed in the other.

Acknowledgments. We would like to thank Mälarenergi for support in the work and Swedish Energy Agency, Värmeforsk and the Knowledge Foundation for financial support.

This contribution is a post-conference publication from SIMS 2012 Conference (53rd SIMS Conference, Reykjavík, Iceland, October 4 - 6, 2012). The contribution is a (partly) modified publication from the paper published in the Proceedings of SIMS 2012, published by Orkustofnun, National Energy Authority Iceland, ISBN: 978-9979-68-318-6, electronically available at <http://www.scansims.org/sims2012/SIMS2012.pdf>.

References

- [1] Asghar A, Tariq S, Torabzadeh-Tari M, Fritzson P, Pop A, Sjölund M, Vasaiely P, and Schamai W. An Open-Source Modelica Graphic Editor Integrated with Electronic Notebooks and Interactive Simulation. In: *Proceedings of the 8th International Modelica Conference (Modelica'2011)*; 2011 March 20-22 Dresden, Germany.
- [2] Avelin A, Jansson J, Dotzauer E, Dahlquist E. Use of combined Physical and Statistical Models for Online Applications in the Pulp and Paper Industry. *Mathematical and Computer Modeling of Dynamical Systems: Methods, Tools and Applications in Engineering and Related Sciences*, 2009; 15(5): 425-434. Doi: 10.1080/13873950903375403
- [3] Casella F, Leva A. Object-Oriented Modelling & Simulation of Power Plants with Modelica. In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*; 2005 Dec 12015; Seville, Spain. P. 7597 – 7602. Doi: 10.1109/CDC.2005.1583388
- [4] Casella F, van Putten H, Colonna P. Dynamic simulation of a biomass-fired steam power plant: a comparison between causal and a-causal modular modelling. In: *Proceedings of the IMECE '07, International Mechanical Engineering Congress and Exhibition*, No. IMECE2007-41091 ASME; 2007. Seattle, WA. Doi: 10.1115/IMECE2007-41091
- [5] Casella F, Colonna P. Dynamic modeling of IGCC. *Applied Thermal Engineering*, 2012; 35(1): 91-111
- [6] Chen SC, Berggren J, Zehnpfund A. Multivariable CD control applications, *ipw*, 2008; 10: 6p.
- [7] Chen SC. *Two dimensional web property variation modeling and control*. US Patent 5,893,055. 1999 Apr 6.
- [8] Ciarapica FE, Giacchetta G. Managing the condition-based maintenance of a combined-cycle power plant: An approach using soft computing techniques. *Journal of Loss Prevention in the Process Industries*, 2006; 19(4): 316–325. Doi: 10.1016/j.jlp.2005.07.018
- [9] Dahlquist E, Widarsson B, Tomás-Aparicio E. *Demand-based maintenance and operators support based on process models*. Project report P08-815. 2011 Dec.
- [10] Dhak J, Dahlquist E, Holmström K, Ruiz J, Belle J, Goedsche F. Developing a generic method for paper mill optimization. In: *Proceedings of the PAPTAC Control Systems 2004 Conference*; 2004 June 14017; Quebec City, Canada.

- [11] Donoso-Bravo A, Mailier J, Martin C, Rodriguez J, Aceves-Lara CA, Vande Wouwer A. Model selection, identification and validation in anaerobic digestion: A review. *Water research*, 2011; 45(17): 5347-5364. Doi: 10.1016/j.watres.2011.08.059
- [12] Fritzson, P. *Principles of Object Oriented Modeling and simulation with Modelica 2.1*. Wiley-IEEE Press; 2004. 940 p.
- [13] Modelica Association. Modelica – A unified Object-Oriented Language for Physical Systems Modeling, Language Specification Version 3.0 [Internet]. [cited 2007]. Available from: www.modelica.org
- [14] Hauge TA, Slora R, Lie B. Application and roll-out of infinite horizon MPC employing a nonlinear mechanistic model to paper machines. *Journal of Process Control*, 2005; 15(2): 201-213. Doi: 10.1016/j.jprocont.2004.05.003
- [15] Jansson J, Grobler F, Dahlquist E. Modelbased control and optimization of continuous digester. *TAPPSA Journal*, 2008; p. 26-33.
- [16] Javed F, Arshad N, Wallin, F, Vassileva I, Dahlquist E. Engineering Optimization Models at Runtime for Dynamically Adaptive Systems. In: *15th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*; 2010 March 22-26; Oxford, UK. P. 253 – 254. Doi: 10.1109/ICECCS.2010.23
- [17] Järvensivu M, Saari K, Jämsä-Jounela SL. Intelligent control system of an industrial lime kiln process. *Control Engineering Practice*, 2001; 9(6): 589-606. Doi: 10.1016/S0967-0661(01)00017-X
- [18] Karlsson CP, Avelin A, Dahlquist E. New Methods for Adaptation to Degeneration in Process Models for Process Industries. *Chemical Product and Process Modeling*, 2009; 4(1). Doi: 10.2202/1934-2659.1127
- [19] Lübken M, Wichern M, Schlattmann M, Gronauer A, Horn H. Modelling the energy balance of an anaerobic digester fed with cattle manure and renewable energy crops. *Water Research*, 2007; 41(18): 4085–4096
- [20] Mattsson SE, Elmqvist H, Otter M. Physical system modeling with Modelica. *Control Engineering Practice*, 1998; 6(4): 501–510. Doi: 10.1016/S0967-0661(98)00047-1
- [21] Mercangöz M, Doyle FJ. Plantwide optimization of a pulp mill process. In: *Proceedings Control Systems*; 2006 June 06-08; Tampere, Finland.
- [22] Otter M, Elmqvist H. Modelica – Language, Library, Tools, Workshop and EU-Project RealSim. 2001.
- [23] Pop A, Sjölund M, Asghar A, Fritzson P, Casella F. Static and Dynamic Debugging of Modelica Models. In: *Proceedings of the 8th International Modelica Conference (Modelica'2012)*; 2012 Sep 03-05; Munich, Germany.
- [24] Sandberg J, Bel Fdhila R, Dahlquist E, Avelin A. Dynamic simulation of fouling in a circulating fluidized biomass fired boiler. *Applied Energy*, 2011; 88(5): 1813–1824. Doi: 10.1016/j.apenergy.2010.12.006
- [25] Salogni A, Colonna P. Modeling of solid oxide fuel cells for dynamic simulations of integrated systems. *Applied Thermal Engineering*, 2010; 30(5): 464-477. Doi: 10.1016/j.applthermaleng.2009.10.007
- [26] Thorin E, Lindmark J, Dahlquist E, Nordlander E, Kastensson J, Leksell N. Performance Optimization of the Växtkraft Biogas Production Plant. *Applied Energy*. Accepted paper 6 March, 2012.
- [27] Vassileva I, Wallin F, Dahlquist E. Analytical comparison between electricity consumption and behavioural characteristics of Swedish households in rented apartments. *Applied Energy*, 2012; 90(1): 182-188. Doi: 10.1016/j.apenergy.2011.05.031
- [28] Vassileva I, Dahlquist E. Economical savings through lowering energy consumption in rented apartments. A case study in Västerås area. *Journal of numerical and applied mathematics*, 2008; 1(90): 234-245
- [29] Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 2003a; 27(3): 293-311. Doi: 10.1016/S0098-1354(02)00160-6
- [30] Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, 2003b; 27(3): 313-326. Doi: 10.1016/S0098-1354(02)00161-8
- [31] Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers and Chemical Engineering*, 2003c; 27(3): 327-346. Doi: 10.1016/S0098-1354(02)00162-X
- [32] Widarsson B, Dotzauer E. Bayesian Networks-based early-warning for leakage in recovery boilers. *Applied Thermal Engineering*, 2008; 28(7): 754-760. Doi: 10.1016/j.applthermaleng.2007.06.016
- [33] Widén J, Lundh M, Vassileva I, Dahlquist E, Ellegård K, Wäckelgård E. Constructing load profiles for household electricity and hot water from time-use data – modeling approach and validation. *Energy and Buildings*, 2009; 41(7): 753–768. Doi: 10.1016/j.enbuild.2009.02.013

A Case Study on Object-oriented Modelling and Simulation of Machine Tools

Bernhard Heinzl^{1,2*}, Michael Landsiedl², Fabian Dür³,
Alexandros-Athanassios Dimitriou³, Wolfgang Kastner¹, Felix Breiteneker⁴

¹Institute of Computer Aided Automation, Vienna University of Technology, Treitlstraße 1-3,
1040 Vienna, Austria; * bernhard.heinzl@tuwien.ac.at

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

³Institute for Production Engineering and Laser Technology, Vienna University of Technology,
Karlsplatz 13, 1040 Vienna, Austria;

⁴Institute for Analysis and Scientific Computing, Vienna University of Technology,
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria;

Simulation Notes Europe SNE 23(3-4), 2013, 147 - 154

DOI: 10.11128/sne.23.tn.10209

Received: March 10, 2013 (Selected I3M 2012 Postconf. Publ.);

Extended Revised: August 10, 2013; Accepted: Nov. 20, 2013;

Abstract. A number of current research projects aim at increasing energy efficiency in production by using comprehensive simulation models of manufacturing plants. In order to gain knowledge about the energetic optimization potential of machine tools, a simulation model of a turning lathe is developed. Using an object-oriented modelling approach allows combining mechanical, electrical as well as thermal aspects in a structural manner into one comprehensive multi-domain model. This bottom-up approach is combined with stepwise top-down model refinement in three stages in order to identify numerical boundaries of the simulation. Simulation results are validated against measurement data. Though object-oriented modelling leads to flexible and modular models, the translated equations are less efficient during simulation, therefore making it necessary to perform manual adjustments in the model. To increase simulation speed, multirate simulation is performed in Simscape using local implicit fixed-step solvers.

Introduction

Rising energy costs and efforts to increase productivity in manufacturing facilities lead to an increased focus on energy efficiency in production. Especially machine tools in metal-cutting manufacturing are among the largest consumers of energy, which have great potential for optimization compared to other energy-intensive manufacturing processes [3].

For this reason, several current research projects aim at increasing energy efficiency by developing comprehensive simulation models of production facilities for energy analysis in order to be able to make qualified prediction about the efficiency of different energy saving measures and identify optimization potential [3].

One part of the work presented here investigates the microstructures of production plants (individual processes and machines) by making extensive energy analysis based on simulation models. Some of these aspects are studied in more detail by developing a multi-domain model of a turning lathe as an example of a machine tool.

A comprehensive approach combines electrical, mechanical as well as thermal aspects of the lathe in one overall model, which afterwards allows for extensive analysis and evaluation regarding energy distribution, comparison of feed and cutting forces as well as dissipated heat.

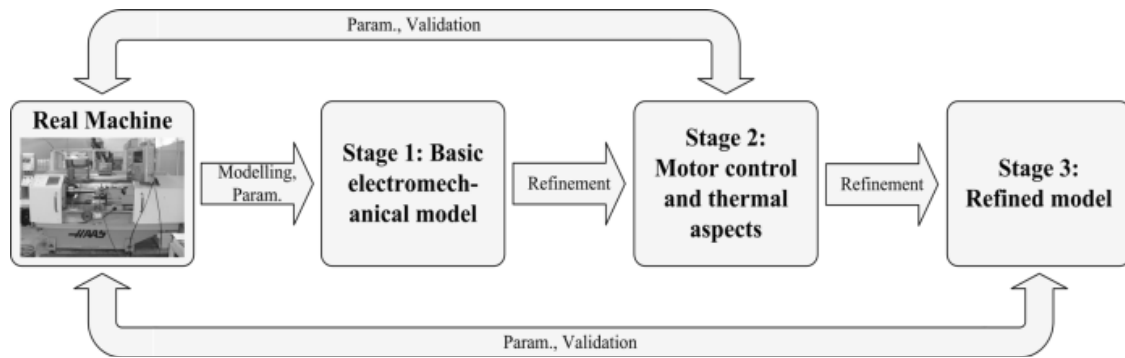


Figure 1: Three stages in the modelling process with increasing level of detail.

For modelling, we consider a high-level object-oriented approach for physical systems, which provides flexibility and modularity for combining bottom-up modelling with stepwise top-down model refinement in three stages with increasing level of detail, see Figure 1.

This procedure enables identifying numerical boundaries of the simulation and shows which model complexity can be handled with sufficient performance and which physical components can therefore be taken into account.

Simulation results are validated against real measurement data obtained from an actual turning lathe.

Implementation is done in MATLAB/SimScope as a common simulator for object-oriented modelling of physical systems [5], [6].

1 Stage 1: Basic Electro-mechanical Model

A first step for model development requires investigation of the turning lathe to be modelled and identification of the main electrical and mechanical components. 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, linear bearings and slide holding the cross feed.
- Cross feed (x-axis): Servomotor, leadscrew drive, linear bearings and cross-slide with 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 2. 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 in the first step as ideal voltage sources.

For implementing the simulation models, SimScope as an extension of MATLAB/Simulink provides an environment for object-oriented multi-domain modelling and simulation of physical systems [5], [6].



Figure 2: 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 first overall model is comparatively simple and contains the main mechanical and electrical components of the main drive and the slides for automatic feed and cross feed. As part of this first model, Figure 3 shows the Simscape model of the main drive with asynchronous engine, voltage supply, gear belt drive, friction components and mechanical loads such as inertias from spindle, chuck and workpiece. The basic structure of the drive is easy to see which is helpful for further model adjustments and refinements, therefore pointing out one of the big advantages of this object-oriented modelling approach.

The asynchronous motor as well as the servo motors for the remaining drives of the lathe and certain basic mechanical components like gear belt drive, lead screw and linear bearings are implemented as custom Simscape components using Simscape Language (see [6]) with parameters extracted from available data sheets. Listing 1 depicts a code fragment of this implementation for the asynchronous machine. It shows common equations in normalized space vector description that can be found in relevant literature (e.g. [7]).

```
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 ASM
    us == is*rs + psis.der/Omegaref_el;
    ur == ir*rr + psir.der/Omegaref_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 the asynchronous machine model in Simscape Language.

Existing Simscape blocks from the Simscape Foundation library (see [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. This load is modelled as a torque source, where the value of the torque is calculated externally using common formulas and parameters (like shown in [2]).

In order to keep the first model simple and focus on modelling of electrical and mechanical parts, feedback control for the drive motors is not included. This however limits possible simulation scenarios, for example only cases with constant motor speed can be considered. Also, thermal aspects are not yet provided in this model.

2 Stage 2: Motor Control and Thermal Aspects

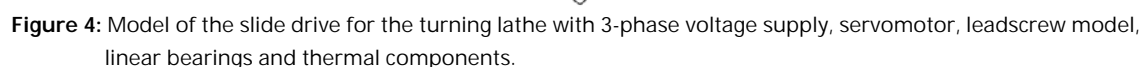
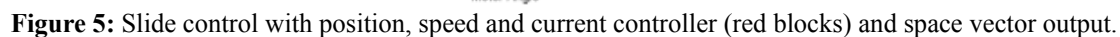
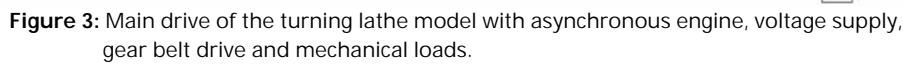
The first modelling stage showed that the object-oriented modelling approach is indeed suitable for basic modelling tasks regarding machine tools. In this next stage we further develop the model and therefore obtain further possible simulation scenarios for observation.

The basic electromechanical model is extended by a number of components:

- Feedback control for all three drive motors
- Calculation of generated heat in lossy components, especially the drive motors
- Heat transition to the environment
- Modelling of energy division in the cutting process

2.1 Modelling

The necessity for appropriate motor control for the overall dynamics is also established in [4]. Figure 4 shows the subsystem for the feedback control implemented in Simulink. Cascaded controllers allow control of position, speed and stator current. Since the stator current is typically controlled in the rotor coordinate system, coordinate transformation using the rotor angle has to be performed. Nominal values are created including speed and acceleration limits according to target positions which are defined in advance. For controller design, we made use of available data from data sheets as well as manual calibration in order for the system to work properly.



The output of the controller subsystem is a vector for the stator voltage, which is then split into phase voltages for an idealized 3-phase converter, which directly supplies the drive motor. Figure 5 shows the graphical representation of slide drive model including the inverter.

Further model extensions take into account the diffuse waste heat of various components, especially the drive motors, gear belt drive and friction elements. For that, all necessary components from the model in Section 1 are modified with a thermal output port. The waste heat is stored and dissipated into the surrounding environment via convective and radiative heat transfer, see Figure 5, with necessary heat transfer coefficients taken from available literature [1], [8].

In order to increase simulation speed, some adjustments had to be made in the Simscape implementation. The three drive train models (main drive, slide and cross-slide) were split into separate Simscape networks (object diagrams), only connected via directed (causal) Simulink signal connections (an overview can be seen in Figure 9). This partial decoupling enables more efficient equation handling by the simulator.

2.2 Simulation results

The modifications now allow more complex simulation scenarios. As an example, we investigate a typical turning process sketched in Figure 6. The following cutting parameters were used in the scenario: Cutting speed $v_c = 200$ m/min, feed $f = 0.2$ mm/U, cutting depth $a_p = 2$ mm, material C45E.

Figure 7 depicts the resulting trajectory of the tool tip and respective time values. All position values are measured with respect to the coordinate system illustrated in Figure 6 (green arrows). The simulation starts at an out-side position. First, slide and cross-slide are accelerated to maximum velocity in order to get to the starting position for the turning process. After that, the turning process is started with smaller feed velocity. The impact point between tool and workpiece does not leave any noticeable disturbances. The process is finished with negative infeed to the final position.

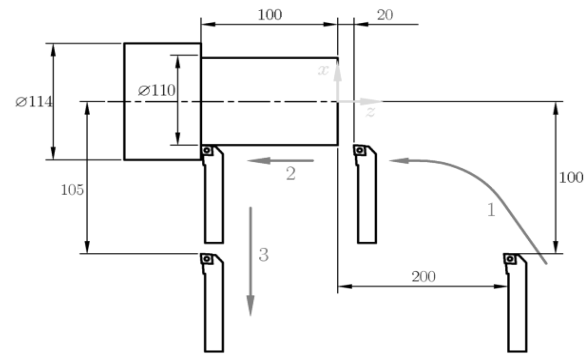


Figure 6: Simulation scenario of a turning process.

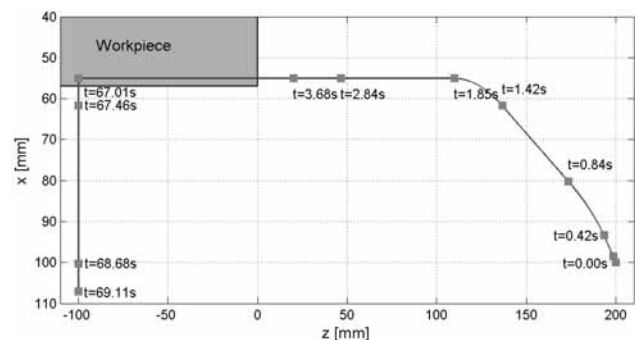


Figure 7: Resulting trajectory of the tool tip and time values for the simulation scenario shown in Figure 6.

For validation, Figure 8 compares the calculated total power consumption against measurement data obtained from the turning lathe. Although the cutting parameters were the same for both cases, there is still significant difference in the results, which shows that further model refinement is necessary.

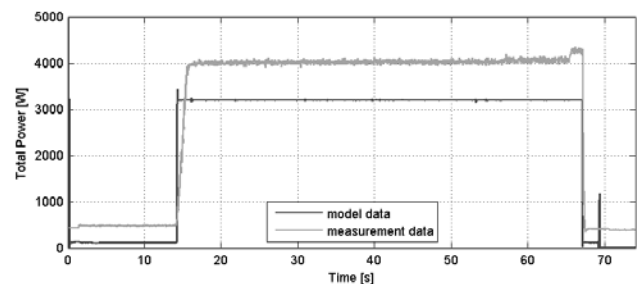


Figure 8: Comparison of total power consumption between simulation and measurement data.

3 Stage 3: Refined Model

The model of the third and final stage is refined by more detailed investigations of the energy supply including power electronic components for rectifier and inverter.

3.1 Modelling

Since the high frequency switching operations of an inverter are difficult to realize with sufficient simulation speed in a mainly continuous model, the switching inverter is replaced by an idealized version which only implements direct energy conservation. Additional electrical loads are also considered because of their influence on the total power consumption.

For better simulation performance, the division of different drivelines into isolated Simscape networks (which are only connected to each other via directed (causal) Simulink signal connections), which was also mentioned in Section 2, is continued and expanded on the new model of rectifier and inverter. Figure 9 gives an overview of these Simscape networks. In the top part, the three blocks in the middle represent the subsystems for the drivelines shown in Figure 3 and Figure 5, resp., which each belong to a separate Simscape network.

In addition, Simscape allows combining the global solver algorithm with local implicit fixed-step solvers, which can handle isolated Simscape networks and therefore allow performing multirate simulations for better performance [6]. In the given model, this method is employed for the rectifier subsystem, therefore making it necessary to isolate the respective part from all other Simscape networks based on assumptions for signal causality between these networks (see Figure 9 bottom).

The local system acts like a discrete subsystem to the global solver, which triggers an event at each local step. A comparison of solver steps between global and local solver is given in Figure 10.

On the one hand, this method results in loss of accuracy for the specified local part, but on the other hand the global solver does not have to resolve high frequency oscillations in the inverter, since this part only appears as a discrete subsystem in the global model.

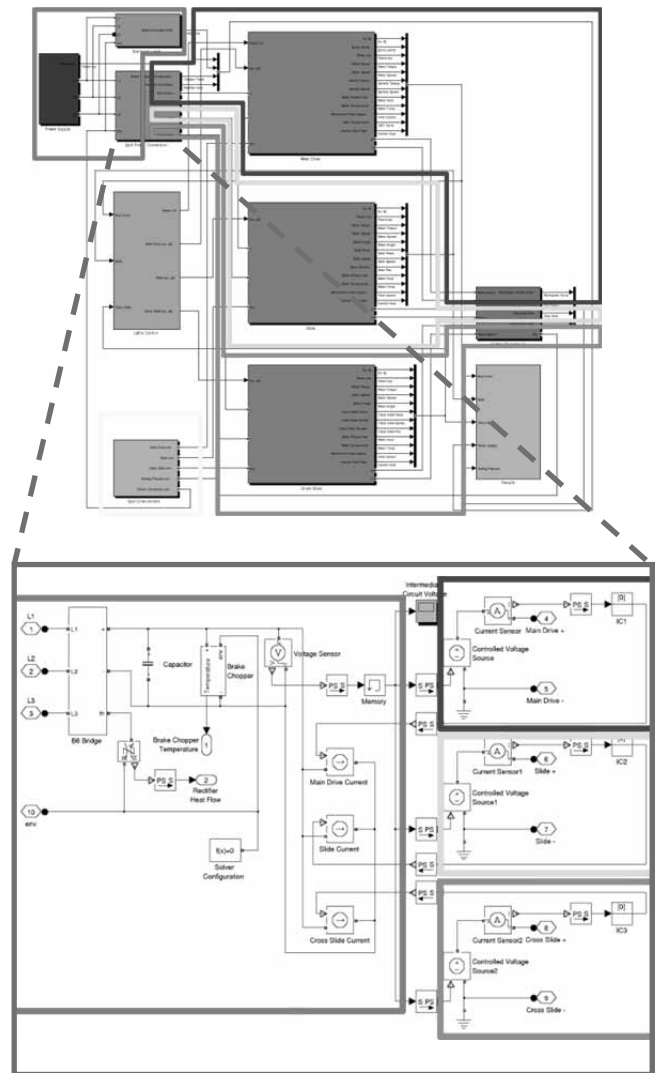


Figure 9: Top: Overview of the total simulation model in Simulink/Simscape. Marked are the separate Simscape networks. Bottom: Detail of the subsystem for the electric 3-phase converter with the three inverters and the rectifier in separate Simscape networks.

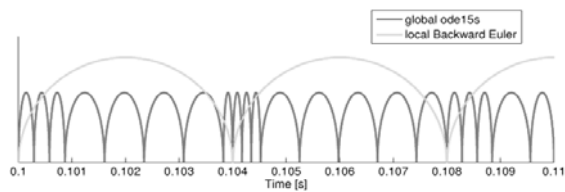


Figure 10: Comparison of steps between local solver (Backward Euler) and global solver (Simulink ode15s).

3.2 Simulation results

For comparison of simulation results, we again consider the scenario shown in Figure 6. The results of a simulation run with the refined model are shown in Figure 11.

Figure 12 visualizes the energy distribution in the system for the given scenario. The input energy is converted into heat mainly in power electronics components, mechanical friction components and during the cutting process. A small part remains as latent energy in the system.

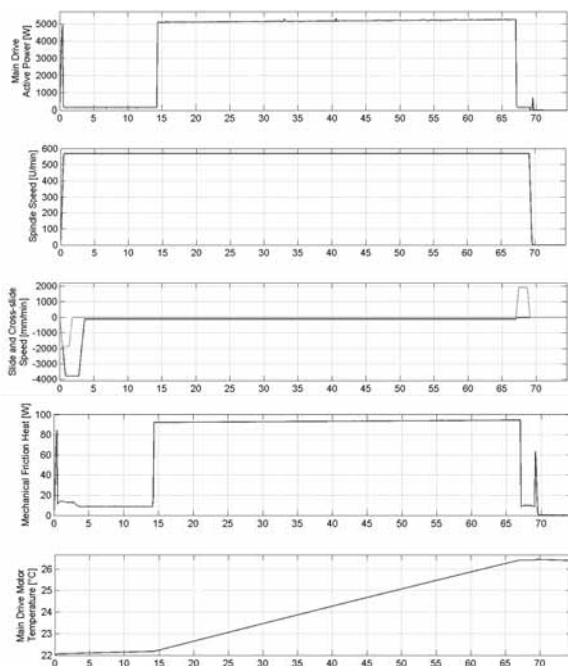


Figure 11: Results of a simulation run with the refined model. The plots show (from top to bottom) main drive active power, spindle speed, slide and cross-slide speed, mechanical friction heat and main drive motor temperature.

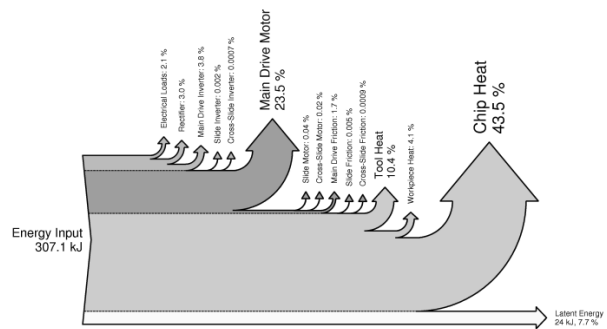


Figure 12: Energy distribution in the system, grouped by parts where input energy is converted into heat (power electronics, mechanical friction, cutting process, latent energy).

4 Conclusion

This case study confirms the advantages of using an object-oriented approach for model development of physical systems. Modular models preserve the basic structure of the original system and can easily be adapted and refined. However, this approach also results in more complex models with larger and less efficient equation systems, so that difficulties with insufficient performance are likely to arise even for models of moderate size.

Manual model adjustments can help increase simulation speed significantly, for example by decoupling the model into isolated networks. This also allows performing multirate or co-simulation by using different and customized solvers. However, splitting the model requires assumptions regarding signal causality, which basically is contradictory to the acausality principle of object-oriented modelling.

Comparison of the simulation results with measurement data showed sufficient agreement in principle; however some improvements are still possible, for example in the models for the drive motors or the cutting process.

References

- [1] Boglietti A, Cavagnino A, Staton DA. Thermal analysis of TEFC induction motors. *Industry Applications Conference*; 2003. P.849-856
- [2] Degner W, Lutze H, Smejkal E. *Spanende Formung. Theorie – Berechnung – Richtwerte*. München: Hanser Verlag; 2009.
- [3] Dorn C, Bleicher F. Interdisziplinäre Forschung zur Energieoptimierung in Fertigungsbetrieben. *Junior Science Conference*; 2010.
- [4] Heinzl B, Dorn C, Dimitriou A. Object-oriented modelling of machine tools for energy efficiency analysis in production. In *Proceedings of MATHMOD Vienna 2012. 7th Vienna Conference on Mathematical Modelling*; 2012; Vienna.
- [5] MathWorks. *Simscape Language Guide*. 2011. Available from: http://www.mathworks.com/help/pdf_doc/physmod/simscape/Simscape_lang.pdf.
- [6] MathWorks. *Simscape User's Guide*. 2011. Available from: http://www.mathworks.com/help/pdf_doc/physmod/simscape/Simscape_ug.pdf.
- [7] Schröder, D. *Elektrische Antriebe – Grundlagen*. Berlin Heidelberg: Springer; 2009. ISBN 978-3-642-02989-9.
- [8] Staton DA, Cavagnino A. Convection heat transfer and flow calculations suitable for electric machines thermal models. *IEEE Transactions on Industrial Electronics*; 2008. P.3509-3516.

A Case Study On Coupling Thermo-dynamical and Electro-mechanical Models in Modelica-based Simulators

Matthias Rößler^{1,2}

¹dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria; matthias.roessler@dwh.at

²Institute for Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

Simulation Notes Europe SNE 23(3-4), 2013, 155 - 160
DOI: 10.11128/sne.23.tn.10211
Received: March 10, 2013 (Selected I3M 2012 Postconf. Publ.);
Extended Revised: Sept. 15, 2013; Accepted: Nov. 20, 2013;

Abstract. Due to higher ecological awareness and more stringent environmental laws the necessity of lowering the energy consumption of manufacturing plants rises. Therefore the possibility to optimize a planned production hall becomes more and more important. The research project INFO, a project of the Vienna University of Technology in cooperation with industry partners and supported by the FFG, the Austrian Research Promotion Agency, creates an overall simulation of a cutting factory. Overall models of such proportions deal with a lot of difficulties, one of them is that through the interdisciplinaryity of the problem there are many submodels that have to be coupled to get to the overall simulation. This work studies one possible solution for this problem, namely the coupling of different model parts in one simulator. First the two partial models are described and their simulation results validated against measurement data. Afterwards the models are coupled and possible problems are analysed and the approach is evaluated.

Introduction

As the necessity of lowering the energy consumption of manufacturing plants rises due to increased energy costs and environmental awareness, the possibility to optimize the energetic behaviour of a production hall becomes more and more important. Especially during the planning phase, saving potential is very high. Therefore the research project INFO, a project of the Vienna University of Technology in cooperation with industry partners and supported by the FFG, the Austrian Re-

search Promotion Agency, creates an overall simulation of a cutting factory. The holistic approach of the project makes it necessary, that models from different physical domains have to be built, taking into account microscopic and macroscopic processes, like machines, the building hull or influences of the weather ([1]). Therefore one important question the project had to solve, was how to couple models from different domains.

In this paper the coupling of different model parts in one simulator will be studied. Therefore two model parts, a room model and a model of a machine tool are developed and afterwards coupled. The models are intentionally kept simple to identify the problems that occur because of the coupling of the model parts rather than numerical problems because of the size of the resulting equation system. The model of the machine tool is a linear guiding device and for the room model a compartment model is chosen. For both models exist test rigs, so they can be validated against measurement data.

Additionally a comparison of two different simulators, Dymola and MapleSim, is made. Both simulators rely on the modelling approach of Modelica, so their numerical behaviour and efficiency can be studied and their suitability for coupling models from different domains can be evaluated.

In the first chapter those two simulators are presented. Then the two models, the room model and the machine model are described in detail. Afterwards the coupling of the two models is realised and the results are presented. Finally an outlook to possible future work is given.

1 Simulators

The Modelica Standard, which is maintained by the Modelica Association, uses an object-oriented approach

to model physical systems. Using this approach a system is modelled by building an acausal block-diagram, where every block represents a real component, for instance a linear translational spring or an electrical resistor. By using the defining equations for each component, which are stored in the representing blocks, and additional equations, that result from connecting the blocks with acausal connections, an equation system is built, which can be solved by a simulator. It is important to note, that neither in the components nor in the connections a causality between the variables is assumed.

This approach works in various physical domains such as mechanics, electrical engineering or thermodynamics, which makes it possible to build multi-domain models. The Modelica Standard only provides a language to describe models in such a manner. To simulate the models a simulator, which understands this language and provides the necessary algorithms to derive and solve the resulting equation system, has to be used. The current version of the Modelica Standard is version 3.2 ([2]).

As mentioned before two different simulators are used in this paper: Dymola and MapleSim. These simulators shall now be described briefly laying a focus on the functionalities used in this paper.

Dymola. The past of Dymola is closely related to Modelica as the Modelica Standard is developed from Dymola. Because of this Dymola can be seen as the simulator who understands the Modelica language best. Models can be built either graphically or textually. The version used in this paper is Dymola 2012, which supports the Modelica Standard 3.2. In this version of the Modelica Standard all dissipative components have an optional heatport, which allows the use of the lost energy in an additional thermodynamical system. This feature is very useful for the purposes of this paper.

MapleSim. MapleSim is developed by MapleSoft. Models can only be built graphically, but new components can be defined using either the Modelica language or a built in Maple function, where only the equations and ports have to be defined and a Modelica component is created automatically. MapleSim supports Modelica Standard 3.1, where the optional heatport is only used in the electrical resistor component; this means that for the other dissipative components the lost energy has to be measured to be used in a thermodynamical model.

2 Submodels

The system that we want to investigate in this paper is a machine tool that stands in a room. The waste heat of the machine heats up the room and is distributed equally in it over time.

2.1 Room model

The surrounding of the machine is subdivided into compartments. The individual compartments can be seen as boxes of a certain size containing air. In reality such a model would have to take into account many things, so several assumptions are made for the model to simplify it:

1. The temperature is constant within the compartment
2. Heat flow is only permitted over the contact surface of adjacent compartments
3. The only way of heat transport is conduction, no convection or radiation is taken into account.
4. The walls of the room are perfectly isolated. No heat flow over the boundaries is permitted, so there is no energy lost in the system.
5. The thermal parameters of the air are assumed to be independent from the temperature and therefore constant over time.

Compartments. Each compartment is implemented as a submodel. Figure 1 shows the basic structure of the compartment.

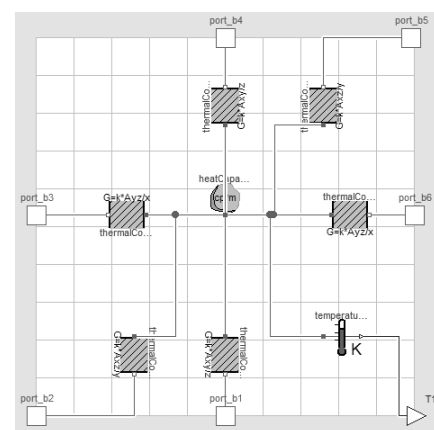


Figure 1: Basic structure of a single compartment in Dymola.

It consists of a heat capacitor that can be seen in the middle of the compartment. This satisfies the first assumption of the room model. In accordance to assumptions 2 and 3 the conductor blocks ensure the heat flow in the 6 possible directions. Additionally a temperature sensor to measure the temperature in the compartment is added.

In Table 1 the parameters for the compartment block are listed. The size of the compartment is given by the expansion along the x -, y - and z -axis, which have to be adjusted to the used compartments. Other parameters that are used by the components within the compartment are derived directly like the volume or the area of the faces of the compartment. Also the thermal parameters of the medium, that the compartment is filled with, are shown. Those parameters are the specific thermal capacity, the density and the conductivity of the medium. In the following simulations the parameters are set to values that correspond to the thermal properties of air at 20°C.

The components used in the compartment submodel use these parameters to calculate the heat flow between the compartments. The thermal mass of the heat capacitor and therefore of the compartment is derived from the volume and the specific thermal capacity that is given in the parameters. In the heat capacitor component the temperature of the compartment is calculated.

x	0.2675	c_p	1012
y	0.17	ρ	1.204
z	0.12	k	0.0257

Table 1: Parameter-set.

To derive the heat flow through the conductor components, the simple geometry of the compartment is used. The heat flow is then only dependent on the area of the contact surface between two compartments A , the distance between the centres of the two δ , the thermal conductance of the medium k and their temperature difference ΔT :

$$Q_{flow} = k \frac{A}{\delta} \cdot \Delta T \quad (1)$$

Validation. For the validation of the model a test rig was built to see if the assumptions made for this model are too restrictive. Therefore a Styrofoam box was used, where eight temperature sensors were attached inside. As a heat source a soldering rod was used. The parameters of the eight compartments used in this model are

depicted in Table 1, the test rig and the model can be seen in Figure 2.

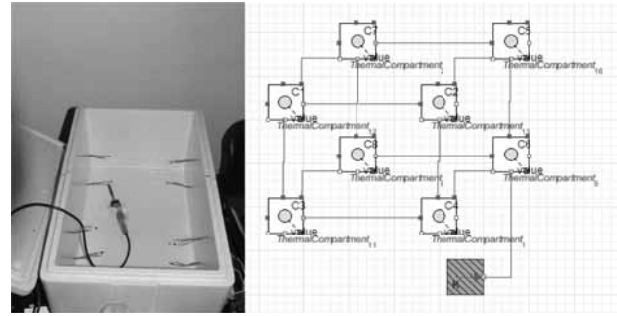


Figure 2: Test rig for the room model and model in MapleSim.

The measurement data of the test rig can be seen in Figure 3, the simulation results of Dymola and MapleSim in Figure 4.

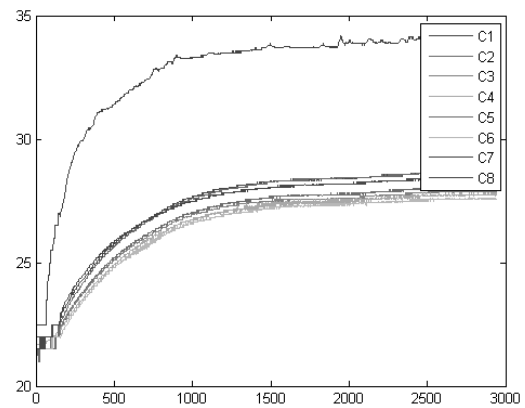


Figure 3: Measurement data of test rig for room model.

It can be seen that the qualitative behaviour of the temperature in the compartment, where the heat source is attached, matches the measurement data, but the magnitude of the temperature rise in the simulation is disproportional. An explanation for this effect can be that the test rig is not isolated well enough, so the assumption of the perfectly isolated walls cannot be met. Further the other compartments heat up much more uniformly in the experiment than in the simulation. This has two main reasons. First in the test rig the heat can flow in any direction, whereas in the simulation it is only permitted in six directions. Second the convection, which is neglected in the simulation, allows a much smoother heat distribution in the experiment.

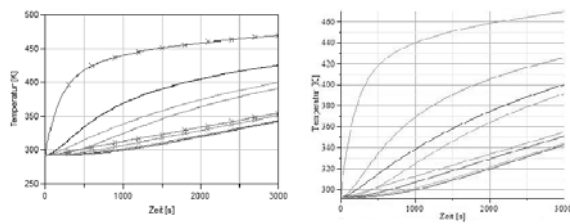


Figure 4: Simulation results from Dymola and MapleSim.

The comparison of the simulation results shows that the two simulators work equally well for this model.

2.2 Machine model

The linear guiding device is a simple part of a machine tool; it consists of a permanent magnet DC motor that drives a thread bar via a gear belt. The thread bar moves a cart, where the sliding mass is attached. The test setup for the validation of the model is provided by the Institute for Production Engineering and Laser Technology from the Vienna University of Technology ([3]) and is shown in Figure 5.

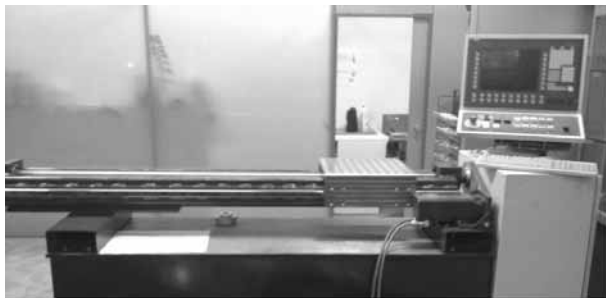


Figure 5: Test setup of linear guiding device.

For the model components of the Modelica Standard Library were used, the general structure of the model can be seen in Figure 6. It shows that the structure of the model is very similar to the structure of the real system, which is an important advantage of this modelling approach.

Due to the graphical representation of the model and the modular structure, the underlying equations are not easy to see, so the individual components of the model are described the following sections.

Permanent magnet DC motor. The electric motor is implemented as a permanent magnet DC motor model. The motor model was updated in version 3.2

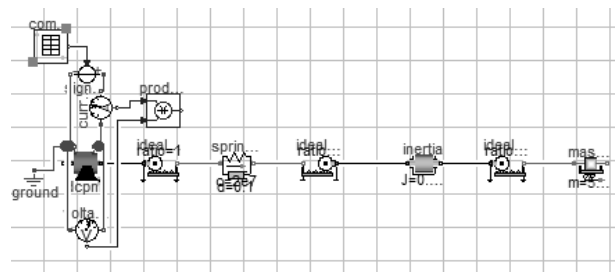


Figure 6: Model of the Machine Tool in Dymola.

of Modelica, so the models in Dymola and MapleSim differ in their complexity. The Dymola model takes different power losses into account, such as friction losses or losses in the stray load [4], which can not be directly modelled in MapleSim due to the lower Modelica version. To ensure comparability between the two models despite the different DCPM models, the parameters, that are used to calculate these losses, are set to zero and the only losses that are taken into account are the ones in the armature circuit, which is modelled as an inductor and a resistor connected in series. As already mentioned the resistor in MapleSim has an optional heatport as well and therefore the loss in this component can be calculated in both simulators in the same way.

Gear belt. The spring constant of a gear belt is always given in a translational manner in the data sheets. Therefore it is modelled as a linear translational spring-damper element, where the damping constant of the component is chosen in such a way that a swinging up of the spring is prevented. The transition between the rotational and translational mechanical domain is made through ideal gears, so no power is lost in this transition.

Thread bar. The defining parameters for a thread bar are its inertia and its lead. So the thread bar in this model consists of an inertia component and an ideal gear, where the transition ratio is directly proportional to the lead of the thread bar.

Cart and mass. As the cart and the mass that shall be moved are rigidly connected to each other, they can be modelled as one mass. The component chosen for the model is a mass with stop and friction, where also a hard stop can be implemented, but is not considered for this model.

The parameters of the friction were chosen in such a way, that the measurement data was approximated

properly by the simulation results. This can be justified by the fact that the other mechanical components in the model are implemented at least nearly ideal, so the friction component can be seen as a cumulative component, where all the mechanical frictions in the system are considered.

Voltage Source. The input for the voltage source, that supplies the voltage for the electric motor, is stored in a time table. The entries of the table are derived from measurement data. The data was taken over the course of one displacement process. The measuring equipment was placed in front of the power electronics, so the first step of adjusting the data was to subtract the mean value of the data points in the idle state, because the power electronics was not considered in the model. Additionally, because the measured signal was very noisy, the data was filtered by a very simple algorithm. Every data point v_i is replaced with the mean value of $v_{i-2}, v_{i-1}, v_i, v_{i+1}, v_{i+2}$. This procedure is repeated ten times. The result of the filtering is depicted in Figure 7.

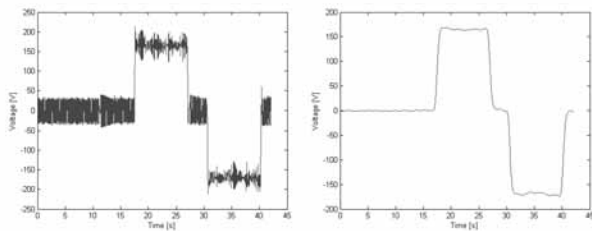


Figure 7: Input-Signal of Voltage Source before and after filtering.

2.3 Simulation results

To compare the simulation results with the measurement data the power consumption of the electric motor was used. Therefore the voltage and current in the armature circuit of the motor was measured in the model and then multiplied. Given the model does not account for the power electronics, the power consumption of the controller had to be added to simulation results, so they can be compared to the measurement data. Figure 8 and 9 show the results of the simulation runs in Dymola and MapleSim compared to the measurement data. Additionally the mean values of the power consumptions over time are shown.

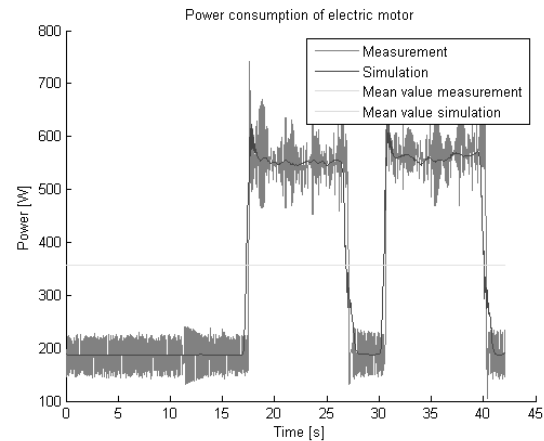


Figure 8: Simulation results in Dymola.

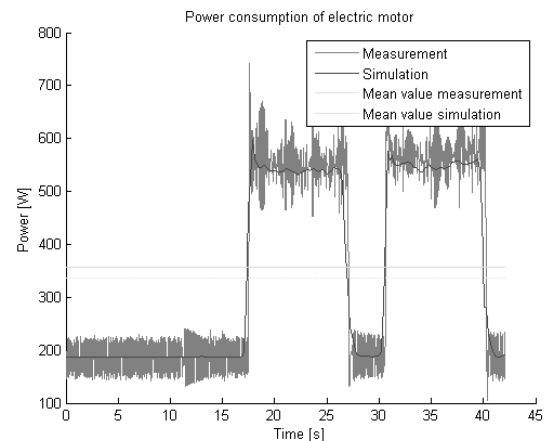


Figure 9: Simulation results in MapleSim.

A qualitative analysis shows that the results of both simulators match the measurement data very well. Comparing the two simulators it can be seen that the results are slightly different. Also the mean values over the displacement period are different. Dymola matches the measurement data better in this regard, whereas the machine in MapleSim consumes less power over the course of one period.

3 Coupling

In a final step the two models are coupled into one model. The energy lost in the dissipative components of the machine model is used as heat source for the room model. In the coupled model the machine stretches across two compartments. The heat loss of the electric motor is a heat source of one compartment and the

heat lost through friction in the mechanical parts heats up another compartment. Figure 10 shows the coupled model.

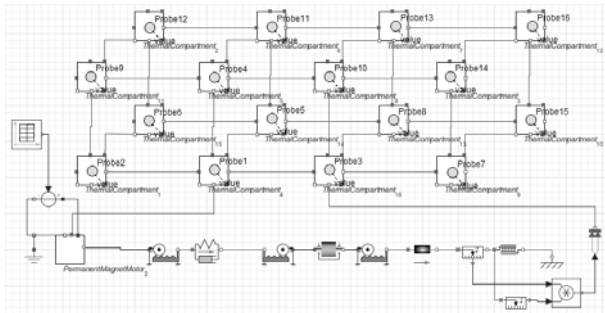


Figure 10: Coupled model in MapleSim.

To model a machine hall the compartments in this model are much larger than in the test scenario. The machine hall is assumed to have an expansion of 20m x 10m x 6m. Additionally changes in the machine model had to be made. In Dymola it was possible to activate the optional heatports to get the heat loss in the dissipative elements, whereas in MapleSim it was necessary to build a new permanent magnet DC motor component to retrieve the energy lost in the resistor of the armature circuit and to measure force and velocity at the friction component to calculate the heat from this component. Figure 11 depicts the simulation results in Dymola for a simulation time of six hours.

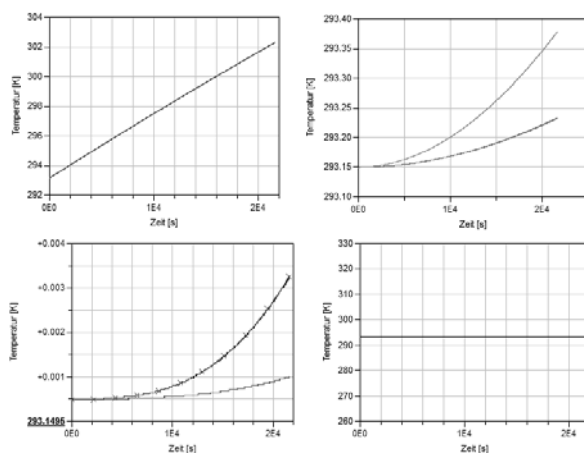


Figure 11: Simulation results of the coupled model in Dymola.

More interesting than the simulation results itself are the computation times it took to simulate the model. The two simulators both needed about half an hour to compute this relatively simple model. Seeing as the

used models are too simplified to represent a real machine or machine hall leads to the conclusion that for more complicated models a coupling in one simulator is nearly impossible even with bigger computational power. The main problem in coupling the two models arises because of the different time constants of the model parts. As the time constants for the electrical and the mechanical part of the machine model are similar, the thermal time constant is proportional to the thermal mass of the studied system. This has the effect that very small steps have to be made by the solver algorithm compared to the relatively large simulation time and in each of these time steps the whole thermal model has to be calculated, which would not be necessary because of the slowly changing behaviour of the system. This can also lead to numerical problems through loss of significance.

4 Conclusion

Due to the simplicity of these models it is possible to couple them in one simulator. But for more complex models the computational effort to simulate the models would rise definitely. To simulate a whole production hall over the course of a year is therefore not advisable with this approach. For the machine model it is to say, that it provides satisfactory results if the input signal is smooth enough. The room model is not sophisticated enough to represent a real room, for that the model assumptions are too restrictive. But the modular construction of the room model allows an easy refinement of the discretization of the model.

References

- [1] Leobner I, Ponweiser K, Neuschwandtner G, Kastner W, Energy Efficient Production – A Holistic Modelling Approach. In *WCST2011-Proceedings*, 62-67. 7th–10th November 2011, London.
- [2] Modelica Corporation, 2010. Modelica – A Unified Object-Oriented Language for Physical Systems Modeling – Language Specification Version 3.2. Available from: <http://www.modelica.org>
- [3] Salvatori, C., 2010, *Energieeffizienz von Werkzeugmaschinen - Planung und Aufbau eines Versuchsstandes zur Untersuchung eines Kugelgewindetriebes*, Bachelorthesis, Vienna UT
- [4] Kral C, Haumer A. Object Oriented Modeling of Rotating Electrical Machines, In: Schmidt, M., eds., *Advances in Computer Science and Engineering*, Croatia: InTech. 2011. 135-160.

Scripting Modelica Models using Python

Bernt Lie^{*}, Finn Haugen

Telemark University College, Porsgrunn, P.O. Box 203, N-3901 Porsgrunn, Norway; ^{*}*Bernt.Lie@hit.no*

Simulation Notes Europe SNE 23(3-4), 2013, 161 - 170
DOI: 10.11128/sne.23.tn.10212
Received: March 10, 2013 (Selected SIM 2012 Postconf. Publ.);
Revised Accepted: November 15, 2013;

Abstract. Modelica has evolved as a powerful language for encoding models of complex systems. In control engineering, it is of interest to be able to analyze dynamic models using scripting languages such as MATLAB and Python. This paper illustrates some analysis and design methods relevant in control engineering through scripting a Modelica model of an anaerobic digester model using Python, and discusses advantages and shortcomings of the Python+Modelica set-up.

Introduction

Modelica is a modern language for describing large scale, multidisciplinary dynamic systems (Fritzson, 2011), and models can be built from model libraries or the user can develop her own models or libraries using a text editor and connect the submodels either using a text editor or a visual tool. Several commercial tools exist, such as Dymola, MapleSim, Wolfram SystemModeler, etc. Free/research based tools also exist, e.g. OpenModelica and JModelica.org. More tools are described at www.modelica.org.

For most applications of models, further analysis and post processing is required, including e.g. model calibration, sensitivity studies, optimization of design and operation, model simplification, etc. Although Modelica is a rich language, the lack of tools for analysis has been a weakness of the language as compared e.g. to MATLAB, etc. Two commercial products are thus based on integrating Modelica with Computer Algebra Systems (MapleSim, Wolfram SystemModeler), while for other tools the analysis part has been more cumbersome (although Dymola includes possibilities for model calibration, an early but simple way of controlling models from MATLAB, etc.).

A recent development has been the FMI standard, which promises to greatly simplify the possibility to script e.g. Modelica models from MATLAB or Python (FMI Toolbox for MATLAB; PyFMI for Python). Several Modelica tools now offer the opportunity to export models as FMUs (Functional Mock-up Units), whereupon PyFMI can be used to import the FMU into Python. Or the FMU can be directly generated from PyFMI. PyFMI is integrated into the JModelica.org tool. More extensive integration with Python is under way for other (free) tools, too.

Python 2.7 with packages Matplotlib, NumPy, and SciPy offer many tools for analysis of models; a simple installation is EPD Free, but many other installations exist.

It is of interest to study whether the combination of (free software) releases of Modelica and Python can serve as useful tools for control analysis and design studies, and what limitations currently limit the spread of such a package. This paper gives an overview of basic possibilities for doing model based control studies by scripting Modelica models from Python. As a case study, a model of an anaerobic digester for converting cow manure to biogas is presented in Section 1. Section 2 presents various examples of systems and control analysis carried out by Python scripts using the model encoded in Modelica. Finally, the results are discussed and some conclusions are drawn in Section 3.

1 Case Study

1.1 Functional description

Figure 1 illustrates the animal waste conversion systems at Foss Biolab in Skien, Norway, which converts cow manure into biogas utilizing Anaerobic Digestion (AD). In this case study, we consider the reactor only (blue box), where the *Feed* is described by a volumetric feed rate $\dot{V}_f [L/d]$ (control input) with a given concentration $\rho_{s_{VS,f}}$ of volatile solids (disturbance).

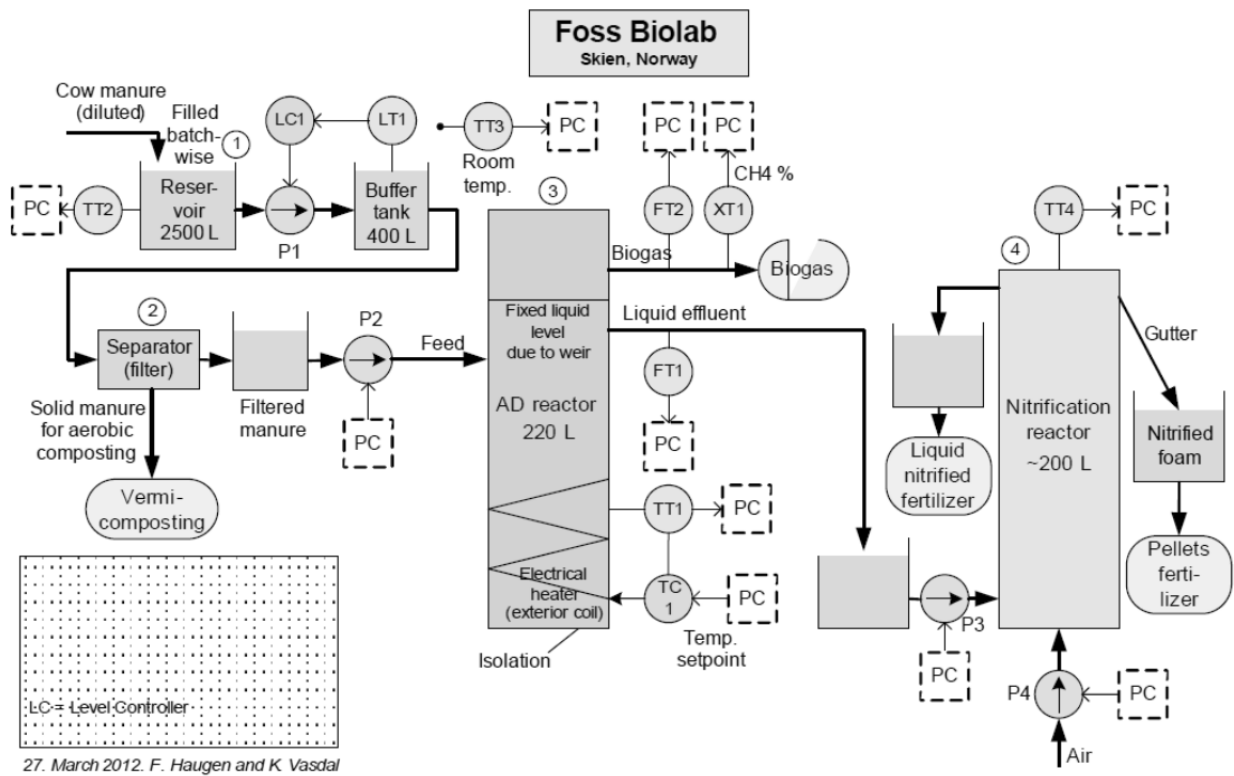


Figure 1: System for converting cow manure to biogas at Foss Biolab, Skien, Norway (Figure by F. Haugen and K. Vasdal).

The ‘liquid’ level of the reactor is made constant by the use of a weir system, and it is possible to control the reactor temperature T accurately using electric heating (potential control input). The main product considered here, is the mass flow rate of methane out of the reactor, $\dot{m}_{CH_4,X}$ (controlled variable).

1.2 Model summary

A model of the reactor is presented in Haugen et al. (2012); in this paper, the same model is used but with a modified notation. The operation of the bio reactor is described by four states $j \in \{\rho_{S_{bvs}}, \rho_{S_{vfa}}, \rho_{X_a}, \rho_{X_m}\}$:

$$\frac{d}{dt}\rho_j = \frac{1}{\theta_j} \frac{\dot{V}_f}{V} (\rho_{j,f} - \rho_j) + R_j$$

where V is constant due to perfect level control, the residence time correction $\theta_{S_j} = 1$ and θ_{X_j} may differ from 1, and furthermore:

$$\begin{aligned} R_{S_{bvs}} &= -Y_{S_{bvs}/X_a} R_a \\ R_{S_{vfa}} &= Y_{S_{vfa}/X_a} R_a - Y_{S_{vfa}/X_m} R_m \end{aligned}$$

with

$$R_{X_a} = R_a - k_a^d \rho_{X_a}$$

$$R_{X_m} = R_m - k_m^d \rho_{X_m}$$

$$R_a = \mu_a \rho_{X_a}$$

$$R_m = \mu_m \rho_m$$

$$\mu_a = \frac{\hat{\mu}_a}{1 + K_{S_{bvs}} \frac{1}{\rho_{S_{bvs}}}}$$

$$\mu_m = \frac{\hat{\mu}_m}{1 + K_{S_{vfa}} \frac{1}{\rho_{S_{vfa}}}}$$

$$\hat{\mu}_a = \hat{\mu}_m = \hat{\mu}_{35} + \alpha_{\hat{\mu}}(T - 35)$$

with units °C for T .

The production (exit) rate of methane is given by

$$\dot{m}_{CH_4,X} = R_{CH_4} V$$

$$R_{CH_4} = Y_{CH_4/X_m} R_m.$$

Feed concentrations of states are given as

$$\rho_{S_{bvs},f} = b_0 \rho_{S_{vs},f}$$

$$\rho_{S_{vfa},f} = a_f \rho_{S_{bvs},f}.$$

Nominal operating conditions for the system are given in Table 1.

Quantity	Value	Unit	Comment
$\rho_{S_{bvs}}(0)$	5.81	g/L	Initially dissolved substrate biodegradable volatile solids
$\rho_{S_{vfa}}(0)$	1.13	g/L	Initially dissolved substrate volatile fatty acids
$\rho_{X_a}(0)$	1.32	g/L	Initially concentration of acetogenic bacteria
$\rho_{X_m}(0)$	0.39	g/L	Initially concentration of methanogenic bacteria
\dot{V}_f	50	L/d	Volumetric feed flow of animal waste/manure
T	35	°C	Reactor temperature
$\rho_{S_{vs},f}$	32.4	g/L	Feed concentration of volatile solids

Table 1: Nominal operational data for biogas reactor at Foss Biolab.

Model parameters are given in Table 2.

1.3 System and control problems

A number of control problems are relevant for this system:

- simulation of the system for validation,
- study of model sensitivity wrt. uncertain parameters,
- tuning model parameters to fit the model to experimental data,
- state estimation for computing hidden model states,
- operation of control system,
- optimal control and model predictive control,
- etc.
-

Only a selected few of these problems are considered in the sequel.

Parameter	Value	Unit	Comment
V	250	L	Reactor volume
$\theta_{X_a} = \theta_{X_m}$	2.9	-	Correction of residence time for bacteria due to nonideal flow
Y_{S_{bvs}/X_a}	3.9	$\frac{gbvs}{gX_a}$	(Inverse) yield: consumption of bvs per growth of bacteria
Y_{S_{vfa}/X_a}	1.76	$\frac{gvfa}{gX_a}$	(Inverse) yield: production of vfa per growth of bacteria
Y_{S_{vfa}/X_m}	31.7	$\frac{gvfa}{gX_m}$	(Inverse) yield: consumption of vfa per growth of bacteria
Y_{CH_4/X_m}	26.3	$\frac{gCH_4}{gX_m}$	(Inverse) yield: production of methane per growth of bacteria
$K_{S_{bvs}}$	15.5	g/L	Half-velocity constant for bvs substrate
$K_{S_{vfa}}$	3.0	g/L	Half-velocity constant for vfa substrate
$\hat{\mu}_{35}$	0.326	d^{-1}	Maximal growth of rate at $T = 35^\circ C$
$a_{\hat{\mu}}$	0.013	$\frac{1}{^\circ C d}$	Temperature sensitivity of maximal growth rate, valid $T \in [20, 60]^\circ C$
$k_a^d = k_m^d$	0.02	d^{-1}	Death rate constants for acetogenic and methanogenic bacteria
b_0	0.25	$\frac{gbvs}{gvs}$	Fraction biodegradable volatile solids in volatile solids feed
a_f	0.69	$\frac{gvfa}{gbvs}$	Fraction volatile fatty acids in biodegradable volatile solids feed

Table 2: Nominal model parameters for biogas reactor at Foss Biolab.

2 Control Relevant Analysis

2.1 Basic Modelica description

The following Modelica encoding in file `adFoss.mo` describes the basic model:

```

Model adFossModel
// Simulation of Anaerobic Digestion Reactor at Foss
// Biolab
// Author: Bernt Lie
// Telemark University College, Porsgrunn, Norway
// August 31, 2012
// Parameter values with type and descriptive text
parameter Real V = 250 "reactor volume,
L";
parameter Real theta_X = 2.9 "residence
time correction for bacteria,
```

```

dimensionless";
parameter Real Y_Sbvs_Xa = 3.9 "Yield, g
  bvs/g acetogens";
parameter Real Y_Svfa_Xa = 1.76 "Yield, g
  vfa/g acetogens";
parameter Real Y_Svfa_Xm = 31.7 "Yield, g
  vfa/g methanogens";
parameter Real Y_CH4_Xm = 26.3 "Yield, g
  methane/g methanogens";
parameter Real K_Sbvs = 15.5 "Half-
  velocity constant for bvs, g/L";
parameter Real K_Svfa = 3.0 "Half-
  velocity constant for vfa, g/L";
parameter Real muhat_35 = 0.326 "Maximal
  growth rate at T=35 C, 1/d";
parameter Real alpha_muhat = 0.013 "Tem-
  perature sensitivity of
  max growth rate, 1/(C d)";
parameter Real k_d = 0.02 "Death rate
  constants for bacteria, 1/d";
parameter Real b0 = 0.25 "Fraction biode-
  gradable volatile solids in
  volatile solids feed, g bvs/g vs";
parameter Real af = 0.69 "Fraction vola-
  tile fatty acids in bvs feed,
  g vfa/g bvs";
// Initial state parameters:
parameter Real rhoSbvs0 = 5.81 "initial
  bvs substrate, g/L";
parameter Real rhoSvfa0 = 1.13 "initial
  vfa, g/L";
parameter Real rhoXa0 = 1.32 "initial
  acetogens, g/L";
parameter Real rhoXm0 = 0.39 "initial
  methanogens, g/L";
// Setting initial values for states:
Real rhoSbvs(start = rhoSbvs0, fixed =
  true);
Real rhoSvfa(start = rhoSvfa0, fixed =
  true);
Real rhoXa(start = rhoXa0, fixed = true);
Real rhoXm(start = rhoXm0, fixed = true);
// Miscellaneous variables
Real rhoSbvs_f "feed concentration of
  bvs, g/L";
Real rhoSvfa_f "feed concentration of
  vfa, g/L";
Real rhoXa_f "feed concentration of
  acetogens, g/L";
4
Real rhoXm_f "feed concentration of meth-
  anogens, g/L";

```

```

Real R_Sbvs "generation rate of Sbvs,
  g/(L*d)";
Real R_Svfa "generation rate of Svfa,
  g/(L*d)";
Real R_Xa "generation rate of Xa,
  g/(L*d)";
Real R_Xm "generation rate of Xm,
  g/(L*d)";
Real R_CH4 "generation rate of CH4,
  g/(L*d)";
Real R_a "reaction rate acetogenesis,
  g/(L*d)";
Real R_m "reaction rate methanogenesis,
  g/(L*d)";
Real mu_a "growth rate acetogenesis,
  1/d";
Real mu_m "growth rate methanogenesis,
  1/d";
Real muhat_a "maximal growth rate aceto-
  genesis, 1/d";
Real muhat_m "maximal growth rate methan-
  ogenesis, 1/d";
Real mdot_CH4x "mass flow methane produc-
  tion, g/d";
// Defining input variables:
input Real Vdot_f "volumetric feed flow -
  - control variable, L/d";
input Real T "reactor temperature -- pos-
  sible control input, C";
input Real rhoSvs_f "feed volatile solids
  concentration -- disturbance, g/L";
equation
// Differential equations
der(rhoSbvs) = Vdot_f/V*(rhoSbvs_f -
  rhoSbvs) + R_Sbvs;
der(rhoSvfa) = Vdot_f/V*(rhoSvfa_f -
  rhoSvfa) + R_Svfa;
der(rhoXa) = Vdot_f/V/theta_X*(rhoXa_f -
  rhoXa) + R_Xa;
der(rhoXm) = Vdot_f/V/theta_X*(rhoXm_f -
  rhoXm) + R_Xm;
// Feed
rhoSbvs_f = rhoSvs_f*b0;
rhoSvfa_f = rhoSbvs_f*af;
rhoXa_f = 0;
rhoXm_f = 0;
// Generation rates
R_Sbvs = -Y_Sbvs_Xa*R_a;
R_Svfa = Y_Svfa_Xa*R_a - Y_Svfa_Xm*R_m;
R_Xa = R_a - k_d*rhoXa;
R_Xm = R_m - k_d*rhoXm;
R_a = mu_a*rhoXa;

```

```

R_m = mu_m*rhoXm;
mu_a = muhat_a/(1 + K_Sbvs/rhoSbvs);
mu_m = muhat_m/(1 + K_Svfa/rhoSvfa);
muhat_a = muhat_35 + alpha_muhat*(T-35);
muhat_m = muhat_a;
// Methane production
mdot_CH4x = R_CH4*Xm*R_m;
R_CH4 = Y_CH4_Xm*R_m;
end adFossModel;

```

2.2 Basic Python script

The following Python script `adFossSim.py` provides basic simulation of the Anaerobic Digester reactor at Foss Biolab starting at the nominal operating point, and performing some step perturbations for the inputs:

```

# Python script for simulating Anaerobic Digester at
#Foss Biolab
# script: adFossSim.py
# author: Bernt Lie, Telemark University College,
#Porsgrunn, Norway
# location: Telemark University College, Porsgrunn
# date: August 31, 2012
# Importing modules
# matplotlib, numpy
import matplotlib.pyplot as plt
import numpy as np
# JModelica
from pymodelica import compile_fmu
from pyfmi import FMUModel
# Flattening, compiling and exporting model as fmu
adFoss_fmu = compile_fmu("adFossModel",
    "adFoss.mo")
# Importing fmu and linking it with solvers, etc.
adFoss = FMUModel(adFoss_fmu)
# Creating input data
t_fin = 100
adFoss_opdata =
    np.array([[0,50,35,32.4],[10,50,35,32.4],
        [10,45,35,32.4],
        [30,45,35,32.4],[30,45,38,32.4],[60,45,38,
        32.4],
        [60,45,38,40],[t_fin,45,38,40]])
adFoss_input = (["Vdot_f", "T",
    "rhoSvs_f"], adFoss_opdata)
# Carrying out simulation
adFoss_res = adFoss.simulate(final_time =
    t_fin, input = adFoss_input)
# Unpacking results
rhoSbvs = adFoss_res["rhoSbvs"]
rhoSvfa = adFoss_res["rhoSvfa"]
rhoXa = adFoss_res["rhoXa"]

```

```

rhoXm = adFoss_res["rhoXm"]
mdot_CH4x = adFoss_res["mdot_CH4x"]
Vdot_f = adFoss_res["Vdot_f"]
T = adFoss_res["T"]
rhoSvs_f = adFoss_res["rhoSvs_f"]
t = adFoss_res["time"]
# Setting up figure with plot of results
plt.figure(1)
plt.plot(t, rhoSbvs, "-r", t, rhoSvfa, "-
    g", t, rhoXa, "-k", t, rhoXm, "-
    b", linewidth=2)
plt.legend((r"$\rho_{S_{bvs}}$ [g/L]", r"$\rho_{S_{vfa}}$ [g/L]",
    r"$\rho_{X_a}$ [g/L]", r"$\rho_{X_m}$ [g/L]"), ncol=2, loc=0)
plt.title("Anaerobic Digestion at Foss
    Biolab")
plt.xlabel(r"time $t$ [d]")
plt.grid(True)
plt.figure(2)
plt.plot(t, mdot_CH4x, "-r", linewidth=2)
plt.title("Anaerobic Digestion at Foss
    Biolab")
plt.ylabel(r"$\dot{m}_{CH_4}$ [g/d]")
plt.xlabel(r"time $t$ [d]")
plt.grid(True)
plt.figure(3)
plt.plot(t, Vdot_f, "-r", t, T, "-
    g", t, rhoSvs_f, "-b", linewidth=2)
plt.axis(ymin=30, ymax=55)
plt.title("Anaerobic Digestion at Foss
    Biolab")

```

Running this Python script leads to the results in figures 2-4:

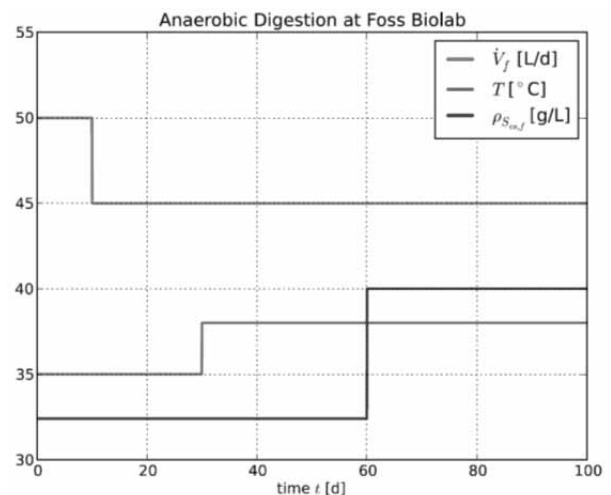


Figure 2: Nominal evolution of inputs at Foss Biolab, with perturbation.

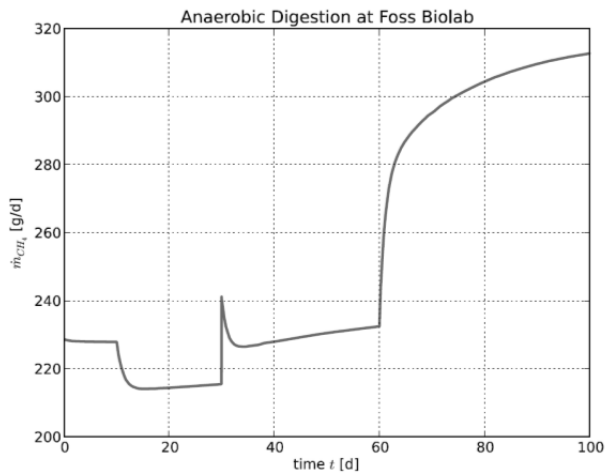


Figure 3: Nominal production of methane gas at Foss Biolab, with perturbation.

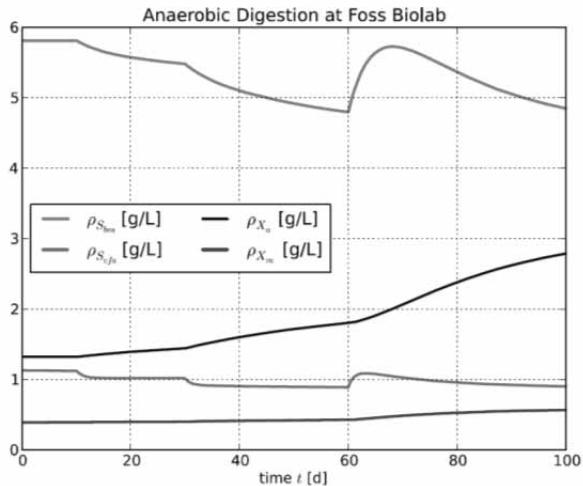


Figure 4: Nominal evolution of states at Foss Biolab, with perturbation.

2.3 Uncertainty analysis

Suppose the value of parameters b_0 and a_f are uncertain, but that we ‘know’ they lie in intervals $b_0 \in 0.25 \times [0.9, 1.1]$ and $a_f \in 0.69 \times [0.9, 1.1]$. We can study the uncertainty of the model by running a number N_{MC} of Monte Carlo simulations where we draw values at random from these two ranges – e.g. assuming uniform distribution. The following modifications of the Python code will handle this problem, excerpt of script `adFossSimMC.py`:

Python script for Monte Carlo study of Anaerobic Digester at Foss Biolab
script: adFossSimMC.py

author: Bernt Lie, Telemark University College, Porsgrunn, Norway

location: Telemark University College, Porsgrunn

date: August 31, 2012

Importing modules

matplotlib, numpy, random

`import matplotlib.pyplot as plt`

`import numpy as np`

`import numpy.random as nr`

`...`

Carrying out simulation

`adFoss_res = adFoss.simulate(final_time = t_fin, input = adFoss_input)`

`...`

Setting up figure with plot of results

`plt.figure(1)`

`plt.plot(t, rhoSbvs, "-r", t, rhoSvfa, "-g", t, rhoXa, "-k", t, rhoXm, "-b", linewidth=2)`

`plt.legend((r"$\rho_{S_{bvs}}$ [g/L]", r"$\rho_{S_{vfa}}$ [g/L]", r"ρ_{X_a} [g/L]", r"ρ_{X_m} [g/L]"), ncol=2, loc=0)`

`plt.title("Anaerobic Digestion at Foss Biolab")`

`plt.xlabel(r"time t [d]")`

`plt.grid(True)`

`...`

Monte Carlo simulations

`Nmc = 20`

`b0nom = adFoss.get("b0")`

`afnom = adFoss.get("af")`

`for i in range(Nmc):`

`b0 = b0nom*(1 + 0.1*(nr.rand()-0.5)*2)`

`af = afnom*(1 + 0.1*(nr.rand()-0.5)*2)`

`adFoss.set(["b0", "af"], [b0, af])`

`...`

Carrying out simulation

`adFoss_res = adFoss.simulate(final_time = t_fin, input = adFoss_input)`

`...`

Unpacking results

`rhoSbvs = adFoss_res["rhoSbvs"]`

`rhoSvfa = adFoss_res["rhoSvfa"]`

`rhoXa = adFoss_res["rhoXa"]`

`rhoXm = adFoss_res["rhoXm"]`

`mdot_CH4x = adFoss_res["mdot_CH4x"]`

`t = adFoss_res["time"]`

`...`

Setting up figure with plot of results

`plt.figure(1)`

`plt.plot(t, rhoSbvs, ":r", t, rhoSvfa, ":g", t, rhoXa, ":k", t, rhoXm, ":b",`

`linewidth=1.5)`

`plt.figure(2)`

`plt.plot(t, mdot_CH4x, ":m", linewidth=1.5)`

`plt.show()`

`...`

`plt.show()`

The results are as shown in figures 5 and 6.

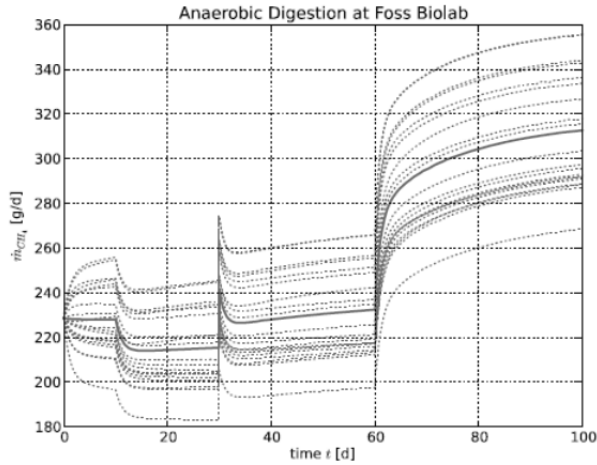


Figure 5: Monte Carlo study of methane production at Foss Biolab, with variation in b_0 and a_f .

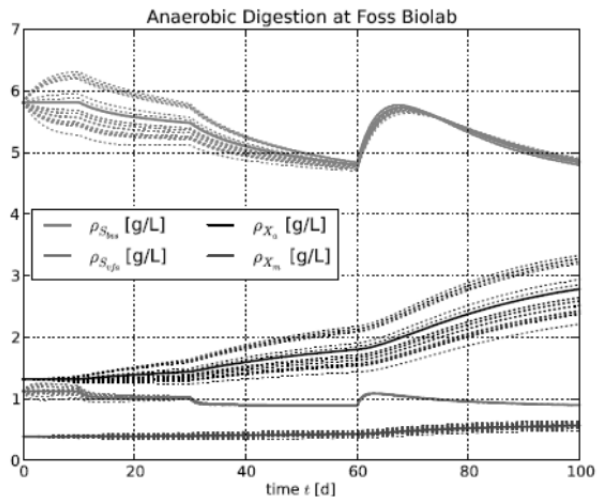


Figure 6: Monte Carlo study of evolution of states at Foss Biolab, with variation in b_0 and a_f .

2.4 Wash-out and recovery of reactor

Suppose that the reactor gets ‘washed out’ by accidentally applying too high a feed rate \dot{V}_f , e.g. $\dot{V}_f = 120 \text{ L/d}$, while T and $\rho_{S_{vs,f}}$ are as in Table 1. It is of interest to see whether the original production can be recovered. Figures 7–9 indicates the behavior over a period of more than 4 years (1500 d) of operation.

As seen, although increasing \dot{V}_f initially leads to a significant increase in the methane production, the bacteria are washed out of the reactor leading to a dramatic fall in the methane production. Furthermore, it takes an inordinate long time to recover after a wash-out if the input is simply set back to the original flow rate.

The steady state values at wash-out ($t = 400 \text{ d}$) can be found to be

$$\rho_{S_{bs}, \text{wash-out}} = 8.0999999985826001$$

$$\rho_{S_{vs,f}, \text{wash-out}} = 3.96169944436781$$

$$\rho_{X_a, \text{wash-out}} = 1.3193454767561001 \times 10^{-9}$$

$$\rho_{X_m, \text{wash-out}} = 0.13282069444970099$$

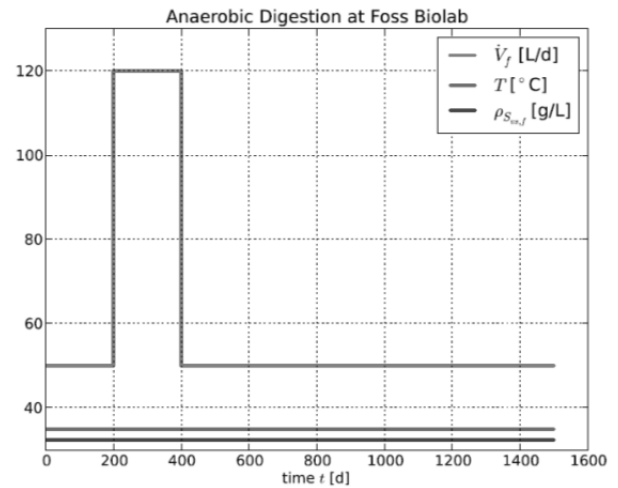


Figure 7: Evolution of inputs at Foss Biolab leading to wash-out/recovery.

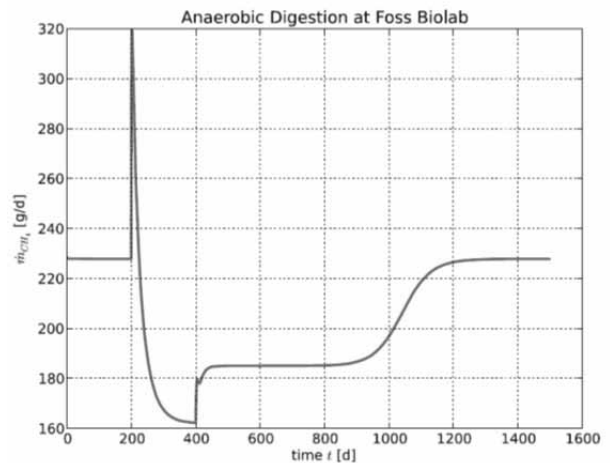


Figure 8: Production of methane gas at Foss Biolab during wash-out/recovery.

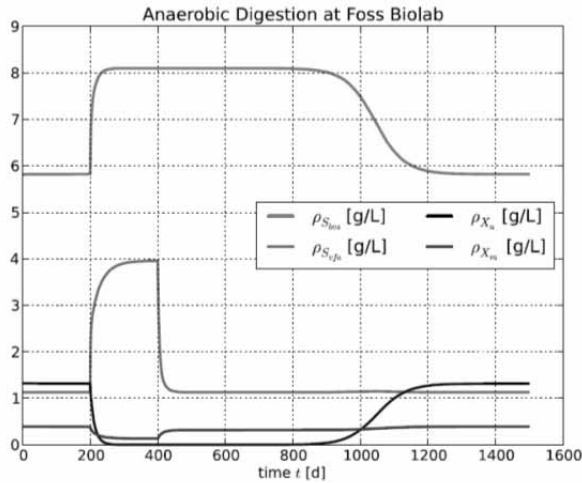


Figure 9: Evolution of states at Foss Biolab during wash-out/recovery.

2.5 Optimal recovery of methane production

The accidental wash-out of bacteria is a serious problem in the operation of Anaerobic Digesters. It is thus of interest to see whether it is possible to recover the operation in an optimal way. We consider the possibility of recovering the operation in the 1100 d horizon spent to wash-out the bacteria, ...g. 7–9. We thus seek to maximize the production of methane, but without using too much feed of animal waste. The following criterion is thus sought *maximized*:

$$J = \int_0^{T_h} (\dot{m}_{CH_4,x} - c_V \dot{V}_f) dt$$

where c_V is a cost parameter. We add the following constraints to make sure that the solution has physical meaning.

$$\rho_j \geq 0$$

$$\dot{m}_{CH_4,x} \geq 0$$

$$\dot{V}_f \in [0, 120] L/d.$$

We assume that the temperature T and the disturbance $\rho_{Svs,f}$ are as in Table 1.

To solve this problem, we use the Modelica extension class *optimization* in JModelica.org. In Modelica, the criterion function is *minimized*, so the criterion in Modelica needs to be $-J$ where J is as above. The essence of the Modelica code for this problem is as given below:

```
optimization adFossOpt(objective =
  J(finalTime), startTime=0,
  finalTime=T_h)

// Optimal recovery of Anaerobic Digestion Reactor at
// Foss Biolab
// Author: Bernt Lie
// Telemark University College, Porsgrunn, Norway
// September 2, 2012
// Instantiating model adf from class adFossModel
adFossModel adf;

// Additional parameters
parameter Real T_h = 1100 "time horizon
  in optimization criterion, d";
parameter Real cost_V = 1 "relative cost
  of animal waste";
parameter Real Vdot_max = 120 "maximal
  allowed feed rate, L/d";
parameter Real T_nom = 35 "nominal reac-
  tor temperature, C";
parameter Real rhoSvs_f_nom = 32.4
  "nominal feed concentration
  of volatile solids, g/L";

// Defining cost function
Real J(start=0, fixed=true);

// Defining input variable:
input Real Vdot_f(free=true,
  min=0,max=Vdot_max) "max feed flow,
  L/d";
equation

// Passing on inputs to model instance
adf.Vdot_f = Vdot_f;
adf.T = T_nom;
adf.rhoSvs_f = rhoSvs_f_nom;

// Computing cost function
der(J) = - adf.mdot_CH4x + cost_V*Vdot_f;
constraint

// Constraining states
adf.rhoSbvs >= 0;
adf.rhoSvfa >= 0;
adf.rhoXa >= 0;
adf.rhoXm >= 0;

// Constraining methane production
adf.mdot_CH4x >= 0;
end adFossOpt;
```

With $c_{\tilde{V}}$, the result is as in figures 10 – 12. With $c_{\tilde{V}}$, the result is highly oscillatoric time evolutions.

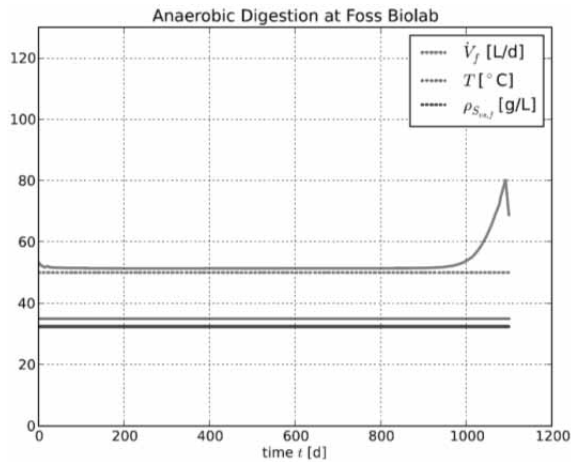


Figure 10: Evolution of optimal input \tilde{V}_f at Foss Biolab after wash-out (solid lines), with initial guess (dotted lines).

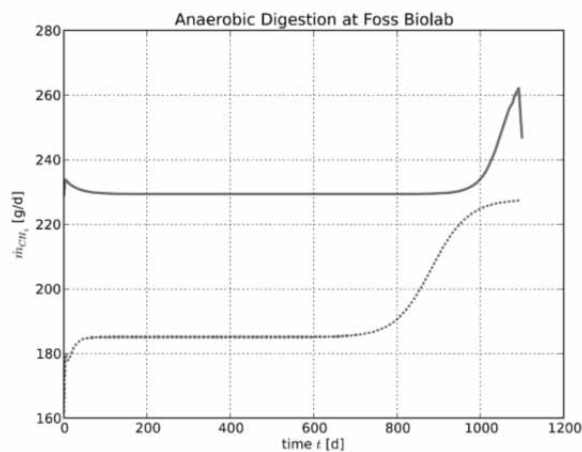


Figure 11: Evolution of optimally recovered methane production at Foss Biolab after wash-out (solidlines), with initial guess (dotted lines).

3 Discussion and Conclusions

Comparing Python to MATLAB for use in control studies reveals clear advantages and clear disadvantages for Python. Python is a free tool, and a rich programming language. However, there is (currently) no control toolbox for Python, the various packages and sub packages are not so well documented, and the quality of some tools are far from perfect. Yet, the combination of Python and Modelica/PyFMI offers ample opportunities for analysis of models and control studies.

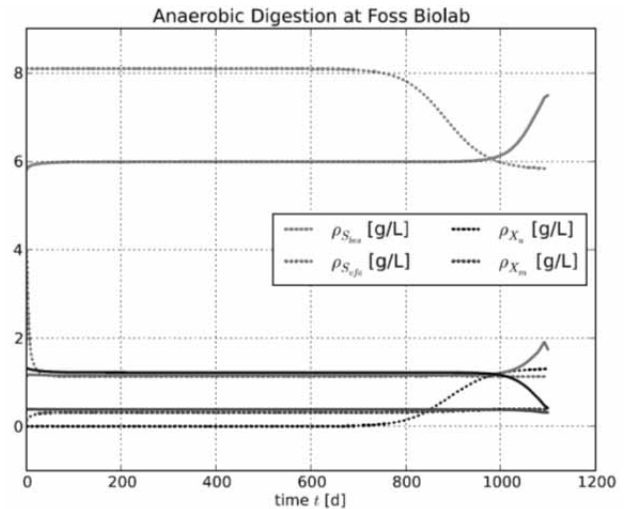


Figure 12: Evolution of optimally recovered states at Foss Biolab after wash-out (solid lines), with initial guess (dotted lines).

This paper illustrates this by showing how natural models can be encoded in Modelica, and how easy Modelica models can be accessed from Python using e.g. PyFMI. Furthermore, it is shown how natural and powerful Python is as a scripting language, e.g. for doing uncertainty/sensitivity analysis of dynamic models. Finally, a simple optimal control problem illustrates on-going research and development in extending the Modelica language using JModelica.org; similar extensions of the Modelica language are also studied in e.g. Bachmann et al.(2012). And yet, in this paper, only the most rudimentary use of Modelica and Python has been touched upon.

Currently, some key problems with the Python+Modelica combination are:

- There is no equivalent of MATLAB's Control Toolbox. This is such a shortcoming that many control engineers will not seriously consider the Python + Modelica combination. Some work at CalTech aims to resolve this problem by developing a Python control toolbox, but there does not appear to be a clear timeline for such a toolbox. Within the Modelica groups, some on-going work addresses this by developing a Linear Systems library within Modelica.

- Although there are a number of powerful (and free) optimization solvers, it is not trivial to integrate these into Python, and those which already have simple Python installers are often poorly documented and/or uses non-standard array packages. A minimal package should include LP, QP, NLP, and NLS solvers of high quality, and they should be equally simple to install in the main OS platforms.
- The FMI is a very positive initiative, and well suited to scripting using either Python or MATLAB. More work is needed in order to make FMI export from the various tools more standardized.
- The initiative of extending Modelica with optimization (and model calibration) possibilities is very interesting for the control society. It would be even more interesting if some standards evolve.

The evolution of alternatives to MATLAB + SIMULINK is very interesting, and Python + Modelica holds promise to be such a tool. There are advantages with commercial tools such as MATLAB+ SIMULINK and similar tools for Modelica such as MapleSim and Wolfram SystemModeler, but in academia with limited resources for buying software, free software is of interest.

Acknowledgement

This contribution is a post-conference publication from SIMS 2012 Conference (53rd SIMS Conference, Reykjavik, Iceland, October 4 - 6, 2012). The contribution is a (partly) modified publication from the paper published in the Proceedings of SIMS 2012, published by Orkustofnun, National Energy Authority Iceland, ISBN: 978-9979-68-318-6, electronically available at <http://www.scansims.org/sims2012/SIMS2012.pdf>.

References

- [1] Bachmann B, Ochel L, Ruge V, Gebremedhin M, Fritzson P, Nezhadali V, Eriksson L, Sivertsson M. (2012). Parallel Multiple-Shooting and Collocation Optimization with OpenModelica. Modelica 2012. *Proceedings of the 9th International Modelica Conference*; Munich.
- [2] Fritzson, P. Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica R. *IEEE*. 2011; Hoboken.
- [3] Haugen F, Bakke R, and Lie B. Mathematical Modelling for Planning Optimal Operation of a Biogas Reactor for Dairy Manure. Presented at the *IWA World Congress on Water, Climate and Energy (IWA-WCE)*; 2012; Dublin.

Acausal Modeling and Simulation of the Standalone Solar Power Systems as Hybrid DAEs

Arash M. Dizqah^{1,*}, Krishna Busawon¹, Peter Fritzson²

¹School of Computing, Engineering and Information Sciences, Northumbria University, NE1 8ST Newcastle Upon Tyne, United Kingdom; **arash.moradinegade@northumbria.ac.uk*

²PELAB Programming Environment Lab, Dep. Computer Science, Linköping University, SE-581 83 Linköping, Sweden

Simulation Notes Europe SNE 23(3-4), 2013, 171 - 178
DOI: 10.11128/sne.23.tn.10213
Received: April 10, 2013 (Selected SIM 2012 Postconf. Publ.);
Revised Accepted: November 15, 2013;

Abstract. Due to the presence of algebraic constraints as well as existing of different modes of operation, a standalone solar power system consisting of photovoltaic arrays, battery bank, electrical load, and a converter is becoming a complex system that can no longer be modeled using the conventional block diagram approach. While the block diagram approach is based on causal interactions between a chain of the ordinary differential equations (ODE), a more appropriate acausal approach solves a flat model of the system consisting of hybrid differential algebraic equations (HDAE). In effect, this paper proposes a nonlinear HDAE-based model of a standalone solar power system. The proposed model is presented using the Modelica language that allows object-oriented and acausal modeling of the multi-mode systems. Next, a general purpose solver is employed to simulate the system. The results of the simulation shows proper match with the information available in the components datasheet. It is shown that the simulation provides a sufficiently accurate prediction of all the system behaviors, which is vital for any model-based controller, including mode transitions.

Introduction

Advances in photovoltaic (PV) technologies during the last decade have increased the share of the solar energy in the growing electricity market. The PV modules are nonlinear and complex still very popular components since they are easy to install and operate. A conventional solar power system employs an array of PV modules

as a generator to supply DC, AC, or mixed electrical load demands. It consists of a converter that normally boosts up the generated DC voltage to match the load characteristics. This converter is equipped with a maximum power point tracker (MPPT) to harvest the maximum available power. The system may optionally consist of an inverter for being connected to AC loads or the grid. Moreover, it consists of a battery bank to overcome the power fluctuation that is the result of the uncertainty in the solar irradiance. The battery also provides a constant voltage bus, which is essential for the MPPT algorithm. Although the voltage changes with regard to the amount of the current as well as the state of the charge (SOC) of the battery, it can be assumed constant during the time that the MPPT calculates the operating point. Figure 1 illustrates the topology of the standalone solar power system selected in this study.

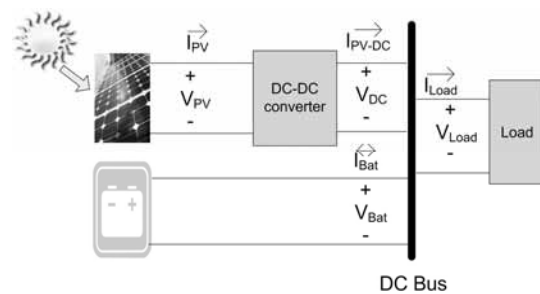


Figure 1: HRES topology in this study.

In order to study the behavior of the solar power system as well as to design model-based controllers, it is essential to model and simulate the system accurately. However, there are two major challenges, namely, the algebraic constraints introduced by the PV module and the battery, and the multimode operation of the battery. Altas and Sharaf [1] used the simplification techniques introduced by Buresch [2] as well as the Simulink basic

components to model and simulate the solar power system consisting of a PV module. Since the Simulink basic components, which provide classical block diagram simulation approach, does not support algebraic states, they introduced a weak dynamic as a short transport delay to their model. Villalva, Gazoli and Filho [3] proposed two simulation scenarios based on equivalent electrical circuits, to simulate the PV module. The proposed scenarios in [3] employ the SimPowerSystems that is one of the Simulink toolboxes for acausal simulation to overcome the algebraic loop introduced by the PV module. The acausal simulation needs the system to be modeled as differential algebraic equations (DAE). Petcut and Dragomir [4] introduced a first-order differential equation associated to the original algebraic equation of the PV module. They proved that the equilibrium point of this differential equation and the solution of the algebraic equation are exactly the same. Introducing this differential equation, they modelled and simulated a PV module with Simulink basic components. Carrero, Amador, and Arnaltes [5] simply fed back the output current of the PV module that introduces algebraic loop. Guasch and Silvestre [6] proposed a comprehensive model for lead-acid batteries along with an equivalent electrical circuit appropriate for simulation. However, their model needs an acausal approach to be simulated because of proposed algebraic loops. Moreover, they introduced a linear interpolation to simulate the mode transition period without considering it as a multi-mode (or in general hybrid) system. Tremblay and Dessaint [7] employed the Simulink SimPowerSystems toolbox to propose an acausal simulation of the lead-acid battery.

Due to the presence of algebraic constraints, the standalone solar power system cannot be decomposed into a chain of blocks with causal interaction and it requires being simulated employing an acausal approach [8]. Moreover, the different modes of operation introduced by battery cause discontinuity points in overall system mathematical model and classify it as a hybrid system [9]. In this paper, a mathematical model of the standalone solar power system is proposed as HDAEs. The Modelica is employed to create an acausal model of the system and the developed model is solved using a general purpose solver. The results of the simulation have been compared with information available in the PV and the battery datasheets provided by the manufacturers that indicate good accuracy.

An outline of the paper is as follows: In the next section, the mathematical models of the system components are presented. Section 2 provides the employed simulation method as well as an overview of the developed Modelica model. The results and their validations are discussed in Section 3. Finally, Section 4 presents a conclusion of the study.

1 Solar Power System Mathematical Model

1.1 PV Module and PV Array

A PV cell is a P-N junction that is typically modeled with an equivalent electrical circuit [10]. Figure 2 illustrates a singlediode equivalent electrical circuit of a PV cell, which is also applicable to model a PV module. The PV module consists of several PV cells connected together in series. A PV array, which is a combination of several photovoltaic modules in series and parallel arrangement, can be modeled with the same circuit, too. There are six electrical parameters in this model, namely, the stray capacity C that can be neglected, the ideality factor of the diode n_D , the photocurrent I_{ph} , the reverse saturation current of the diode I_0 , the series resistor R_s , and the shunt resistor R_{sh} . While the series resistor R_s is the sum of structural resistances, the shunt resistor R_{sh} represents the leakage current [11]. There are prior researches to estimate the electrical parameters from the PV module information available in datasheet [3], [11], [12]. Through this study, it is assumed that the equivalent electrical circuit of the PV module has already been identified using one of these methods. In Figure 2 applying the Kirchhoff current law (KCL) to the junction point of these two resistors gives the characteristic equation of the PV module, which is a nonlinear transcendental equation, as follows:

$$I_{pv} = I_{ph} - I_0 \left\{ \exp \left(\frac{V_{pv} + R_s I_{pv}}{n_d N_s} \frac{q}{KT_c} \right) - 1 \right\} - \frac{V_{pv} + R_s I_{pv}}{R_{sh}} \quad (1)$$

where V_{pv} and I_{pv} are, respectively, the output voltage and current of the PV module and all other symbols are defined as follows:

- q The electron charge ($1.60218 \cdot 10^{-19}$)
- K The Boltzman constant ($1.38066 \cdot 10^{-23}$)
- N_s The number of the PV cells in series as the PV module (-)
- T_c The current amount of the PV cell temperature (K)

The photocurrent I_{ph} and the reverse saturation current of the diode I_0 are calculated with (2) and (3) based on available parameters in the datasheets and the estimated electrical parameters [3], [13]. The performance of different PV modules are measured at a standard test condition (STC), which is defined as follows:

- The cell temperature of 25C
- The Global solar irradiance of $1000(W/m^2)$
- Air Mass of 1.5

Figure 3 shows the PV current-voltage ($I - V$) curve indicating the maximum power point (MPP). The MPP varies due to changes in operating conditions (i.e. the solar irradiance and the cell temperature) and requires being followed by a MPPT algorithm:

$$I_{ph} = \left(\frac{R_s + R_{sh}}{R_{sh}} I_{sc, stc} + k_I (T_c - T_{c, stc}) \right) \frac{S}{S_{stc}} \quad (2)$$

$$I_0 = \frac{I_{sc, stc} + k_I (T_c - T_{c, stc})}{\exp \left(\frac{V_{oc, stc} + k_V (T_c - T_{c, stc})}{n_d N_s} \frac{q}{KT_c} \right) - 1} \quad (3)$$

where:

- $I_{sc, stc}$ The short-circuit current of the PV module at the STC (A)
- k_I The temperature coefficient of the short-circuit current (A/C)
- k_V The temperature coefficient of the open-circuit voltage (V/C)
- S The current amount of the solar irradiance (W/m)
- S_{stc} The amount of the solar irradiance for the STC (W/m)
- $T_{c, stc}$ The amount of the cell temperature at the STC (K)
- $V_{oc, stc}$ The open-circuit voltage of the PV module at the STC (V)

The PV array of the solar power system introduces an algebraic nonlinear constraints between V_{PV} and I_{PV} , i.e., none of these variables are among degrees of freedom and the former is an algebraic state.

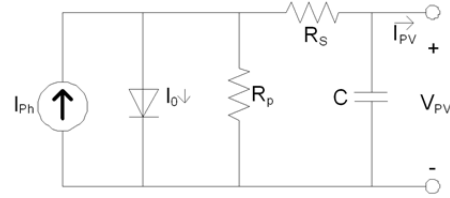


Figure 2: The single-diode equivalent electrical circuit of a PV module.

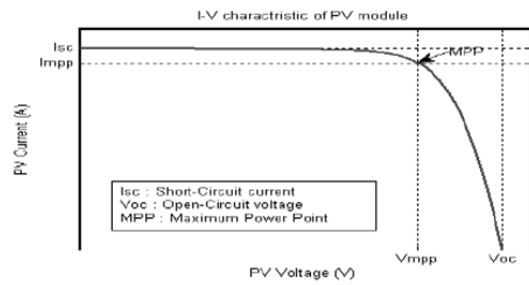


Figure 3: The $I - V$ curve of the PV module for the STC.

1.2 Boost-type DC-DC converter

The boost-type DC-DC converter is one of the converter options which is implemented based on the switching-mode circuit technology containing at least one energy storage and two semiconductor switches. However, in the ideal case a single-pole double-throw switch can be used. Figure 4 illustrates the electrical circuit for a boost-type DC-DC converter with an ideal switch.

Middlebrook and Cuk [14] proposed a state-space averaging approach to model the boost-type DC-DC converter. They suggested two i_L and v_C states for the continuous conduction mode (CCM) in which the instantaneous inductor current is always greater than zero. According to the proposed approach there is a state-space system for the converter at each state of the switch and the overall state-space model is the weighted average of these two models. The weighting factor is switching duty-cycle (D) as the only manipulating control signal. The switching duty-cycle is the duration of time that the circuit remains in each state. Equations (4) and (5) provide the average model of the boost-type DC-DC converter using ideal switch and diode.

$$\begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} = \begin{bmatrix} -\frac{R_L}{L} - \frac{R_C(1-D)}{L(1+\frac{R_C}{L})} & -\frac{1-D}{L(1+\frac{R_C}{L})} \\ \frac{1-D}{C(1+\frac{R_C}{L})} & -\frac{1}{RC(1+\frac{R_C}{L})} \end{bmatrix} \cdot \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} V_{in}(t). \quad (4)$$

$$V_{out} = \begin{bmatrix} \frac{R_C(1-D)}{1+\frac{R_C}{L}} & \frac{1}{1+\frac{R_C}{L}} \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} \quad (5)$$

1.3 Lead-acid battery

There are different types of batteries applicable for backup/storage of the solar power system. The authors in [15] [16] presented a comprehensive reviews and comparisons of different battery technologies for the renewable energy applications. The battery introduces an algebraic constraint between V_{bat} and I_{bat} where the former is an algebraic state.

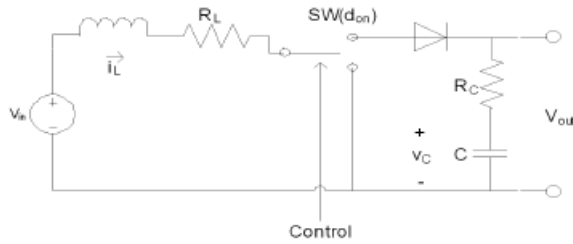


Figure 5: The electrical circuit of the boost-type DC-DC converter with an ideal switch.

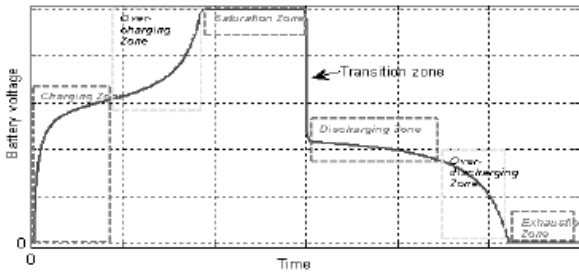


Figure 6: Different working zones of the lead-acid batteries.

Figure 5 illustrates a complete operating cycle of the battery starting from the charging zone up to completely being discharged, i.e., the exhaustion zone [6]. In Figure 5, it can be seen that the battery is a multi-mode component introducing discontinuity to the system. Tremblay and Dessaint [7] proposed an equivalent electrical circuit (Figure 6) as well as the following hybrid mathe-

matical presenting two modes of operation, namely, charging, and discharging:

$$V_{bat} = \begin{cases} V_0 - RI_{bat} + V_{exp} - \frac{P_1 C_{max}}{C_{max} - charge} charge - \frac{P_1 C_{max}}{charge + 0.1 C_{max}} I_{bat} & \text{mode=charging,} \\ V_0 - RI_{bat} + V_{exp} + \frac{P_1 C_{max}}{charge - C_{max}} charge + \frac{P_1 C_{max}}{charge - C_{max}} I_{bat} & \text{mode=discharging.} \end{cases} \quad (6)$$

$$mode = \begin{cases} \text{charging} & I_{bat} \leq 0, \\ \text{discharging} & I_{bat} > 0. \end{cases} \quad (7)$$

$$\frac{dcharge}{dt}(t) = \frac{1}{3600} I_{bat}(t). \quad (8)$$

$$\frac{dV_{exp}}{dt}(t) = \begin{cases} \frac{P_2}{3600} |I_{bat}| (P_3 - V_{exp}) & \text{mode=charging,} \\ -\frac{P_2}{3600} |I_{bat}| V_{exp} & \text{mode=discharging.} \end{cases} \quad (9)$$

where V_{bat} and I_{bat} are the voltage and the current of the battery, respectively. The parameters $P_1 - P_3$ are the experimental parameters requires being identified for each type of the battery and $V_{exp}(V)$ models the exponential behavior of the battery voltage during the mode transition period. While the current direction varies instantaneously, the voltage of the battery changes with an exponential factor after the mode transition. The C_{max} is the maximum amount of the battery capacity (Ah), R is the internal resistor of the battery (Ω), $charge(Ah)$ is the actual battery capacity, and V_0 is the battery constant voltage (V).

2 Simulation

According to the block diagram approach for modelling and simulation, the system is decomposed into distinguished blocks with causal interactions. In a sequential procedure starting from the first block through all oth-

ers, the simulator calculates output values with respect to the inputs and then feeds the outputs to the next block as their input variables. Although it is a straightforward approach, it still suffers from several drawbacks. In order to decompose a system into a chain of causal interacting blocks, the system must have just the differential states coupled with algebraic equations to calculate the outputs. In other words, it must be a system of ordinary differential equations (ODE) and any algebraic equation has to calculate only an algebraic variable that is a degree of freedom. However, most of the electrical and mechanical systems introduce algebraic states. An algebraic state is a state of the system which is restricted with an algebraic equation. Any system consisting of algebraic states, which are modeled with DAEs rather than ODEs, requires being simulated employing an acausal approach.

Despite a causal model which expresses an algorithm to calculate the variables, the acausal modeling is a declarative approach in which individual parts of the model are directly described as equations that are in

```

model HRES_SolarSystem
  Modelica.Blocks.Sources.Constant Sx(k = 300.0);
  Modelica.Blocks.Sources.Constant Tx(k = 298.15);
  Modelica.Electrical.Analog.Basic.Ground ground;
  HRES.PVArray pvarray(Npvp = 10);
  Modelica.Blocks.Sources.TimeTable DutyCycle(table = [...]);
  Modelica.Blocks.Sources.Step ramp(startTime = 5, offset = 10,
height = -6);
  Modelica.Electrical.Analog.Basic.VariableResistor load;
  HRES.BoostConverter converter(R1 = 0.001, Rc = 0.3, L = 0.0005,
C = 0.005);
  HRES.LeadAcidBattery battery1;
  HRES.LeadAcidBattery battery2;
  HRES.LeadAcidBattery battery3;
  HRES.LeadAcidBattery battery4;
equation
  connect(Tx, y.pvarray.Tx);
  connect(Sx, y.pvarray.Sx);
  connect(pvarray.n, ground.p);
  connect(pvarray.p, converter.p1);
  connect(converter.n1, ground.p);
  connect(converter.n2, ground.p);
  connect(DutyCycle, y.converter.D);
  connect(converter.p2, load.p);
  connect(load.n, ground.p);
  connect(converter.p2, battery1.p);
  connect(battery1.n, battery2.p);
  connect(battery2.n, battery3.p);
  connect(battery3.n, battery4.p);
  connect(battery4.n, ground.p);
end HRES_SolarSystem;

```

Figure 7: The first Plane of the Modelica Codes Modelling the Solar Power System.

general HDAEs. An HDAE represents both continuous-time (either differential or algebraic) and discrete-time behaviors [9]. Although there are general purpose numerical HDAE solvers such as tools in [17], [18], [19], the acausal modeling still needs to be described with a programming language. Modelica [20], which is an object-oriented and equation-based language providing the capability to model the hybrid systems [21], is an option to describe complex systems using an acausal approach. Among a number of available implementations of the Modelica compiler, the OpenModelica platform [22] has been selected for this study chiefly because it supports more features of the Modelica language comparing with the others. The platform translates the Modelica descriptions into a flat model in the form of HDAEs, then generates C or XML code from the flat model. The compiled model is simulated using the integrated DASSL that is a general purpose HDAE solver [19].

```

class LeadAcidBattery
  ...
equation
  chargeState = if noEvent(ibat ≤ 0) then true else false;
  der(charge) = 1/3600 * ibat;
  der(Vexp) = if chargeState then
    P2/3600 * abs(i) * (P3 - Vexp)
  else -(P2 * abs(i))/3600 * Vexp;
  when change(chargeState) and pre(chargeState) then
    tmp = if not chargeState then
      pre(vbat) - V0 - R * pre(ibat) -
      (P6 * Cmax)/(Cmax - pre(charge)) * pre(charge) -
      (P6 * Cmax)/(pre(charge) + 0.1 * Cmax) * pre(ibat)
    else 0;
    reinit(Vexp, tmp);
  end when;
  soc = 1 - charge/Cmax;
  vbat = if chargeState then
    V0 - R * ibat - (P1 * Cmax)/(Cmax - charge) * charge -
    (P1 * Cmax)/(charge + 0.1 * Cmax) * ibat + Vexp
  else V0 - R * ibat - (P1 * Cmax)/(Cmax - charge) * charge -
    (P1 * Cmax)/(Cmax - charge) * ibat + Vexp;
end LeadAcidBattery;

```

Figure 8: A summary of the lead-acid battery Modelica class.

Figure 7 shows the first plane of the developed Modelica codes that models the standalone solar power system. From Figure 7, it can be seen that the proposed model consists of a PV array connected to a resistive load through a converter.

A battery bank including four lead-acid batteries is employed to prevent power fluctuations. All these components are also modeled as separate classes of the Modelica language. For instance, Figure 8 summaries such a model for the lead-acid battery and its multi-mode operation.

3 Results, Validation and Discussion

The proposed model is used to simulate a solar power system consisting of an array of the Kyocera KC200GT PV modules [23] and a bank of the Panasonic LC-R127R2PG lead-acid batteries [24]. The authors in [3] and [7], respectively, presented the identified electrical parameters of the PV module and the lead-acid battery that are used in this study.

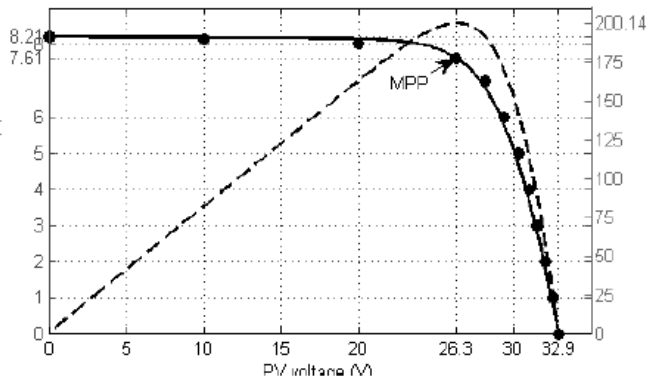


Figure 9: The simulated current-voltage curve of the KC200GT PV module at the STC.

Figure 9 shows the simulated current-voltage ($I - V$) and power-voltage ($P - V$) curves of the KC200GT PV module at the STC condition. It is observed that the proposed model predicts the curve very close to the empirical data provided by the manufacturer (the circle markers). The simulated MPP is in the same position of the experimental point as well.

Figure 10 illustrates the simulation results of the LC-R127R2PG lead-acid battery [24] for a full cycle as in Figure 5 that is a cycle of charging, over-charging, saturation, discharging, overdischarging, and exhaustion zones. While the battery is being charged for 100 minutes, it is discharged afterward. It also indicates that after 25 minutes it enters into the over-charging zone. Discharging with the current of 7.2A in average, it takes around 35 minutes for the battery, which matches with

the information available in datasheet [24], to reach the cut-off voltage that is around 10.2V.

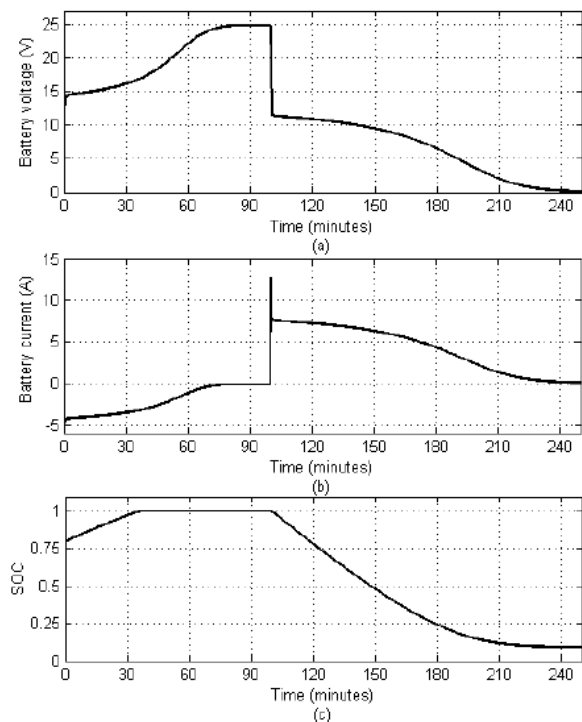


Figure 10: The simulated (a) battery voltage, (b) battery current, and (c) the SOC of the battery.

The solar power system is simulated using the proposed model for the following scenario:

- Simulation duration is 10 minutes.
- The solar irradiance is $1000W/m^2$ and the cell temperature is $25^{\circ}C$.
- The PV array consists of 10 connected KC200GT PV modules in parallel arrangement.
- The battery bank consists of 4 connected LC-R127R2PG lead-acid batteries in series arrangement.
- The manipulating control signal changes according to a stepwise pattern.
- The load demand suddenly increases at $t = 5$.
- After 6 minutes the generated power by the PV array declines.

In Figure 11, it can be seen that the battery absorbs excess energy when there is and provides the deficit energy when the PV module cannot provide adequate energy. For the first 5 minutes, the generated power is more than load demand and the battery, which absorbs the excess energy, is in charging mode.

The load demand suddenly increases to more than the generated power at $t = 5$ minutes that switches the state of the battery, which now provides the deficit energy, to the discharging mode. At $t = 6$ minutes the battery provides more power because the generated power by the PV array declines even less due to manipulating of the control signal. Figure 11 illustrates a power loss of around 10% in the DC-DC converter, which is the difference of the power generated by the PV array and the sum of the load and the battery powers, chiefly because of the internal resistance of the coil.

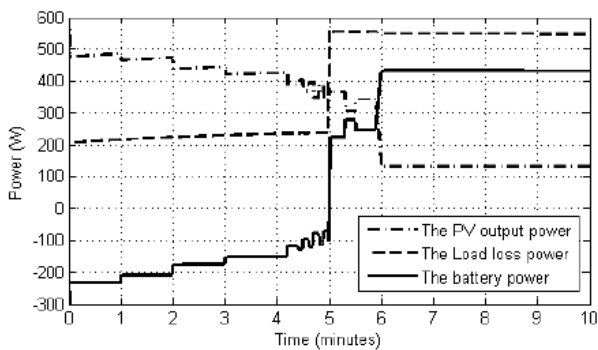


Figure 11: The simulated battery, load, and PV powers.

Figure 12 illustrates the details of battery bank. It is observed that during the period of charging of the battery bank ($time < 5$ minutes), the voltage and the SOC of the battery bank increase continuously. However, the voltage of the battery suddenly falls at $t = 5$ minutes to the nominal value by the transition from charging mode to discharging mode. There is another step change at $t = 6$ minutes as the result of decreasing the generated power by the PV array (Figure 11). The SOC also declines moderately after $t = 5$ minutes since the battery is being discharged.

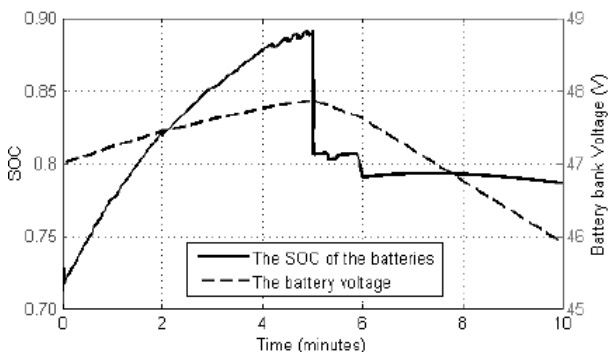


Figure 12: The simulated SOC and the voltage of battery bank.

Finally, Figure 13 depicts the variation of the battery bank current during the simulation period. While it is less than zero, which means the battery bank is in charging mode, before the $time = 5$ minutes, it suddenly becomes greater than zero after entering in the discharging mode. The small step changes are due to variation in the control signal causing changes in the harvested energy from the PV array. A load demand increase at $t = 6$ minutes causes a sharp climb of the battery bank current that matches the power variation in Figure 11.

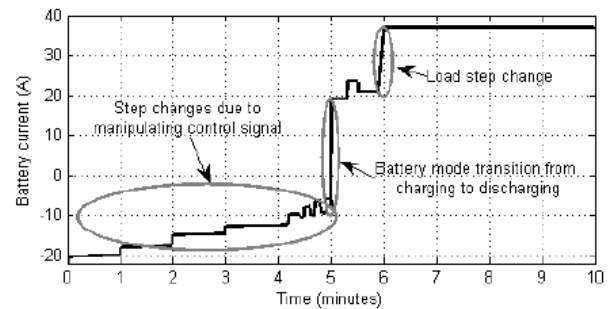


Figure 13: The simulated battery current.

4 Conclusion

This paper proposes an acausal model, which is based on the HDAE, to simulate the solar power system. The proposed model presents the nonlinear algebraic constraints, which are introduced by the PV array and the battery, as DAEs. Moreover, it models different modes of the battery operation as a hybrid system. The Modelica language is employed to describe the system as an acausal model organized as separate Modelica classes for different components. The OpenModelica environment as an integrated modeling and simulation Modelica tool-set is used to simulate the system with the DASSL general purpose integrator. The PV array and the lead-acid battery bank are separately simulated and validated with information available in datasheets that show very good accuracy. The whole solar power system is also simulated and discussed thoroughly indicating accurate prediction of all the system behaviors including mode transitions. The highest level Modelica codes as well as a summary of the battery Modelica class are presented.

Acknowledgment. The authors would like to thank the Synchron Technology Ltd. company for their partial financial support of this research.

This contribution is a post-conference publication from SIMS 2012 Conference (53rd SIMS Conference, Reykjavik, Iceland, October 4 - 6, 2012). The contribution is a (partly) modified publication from the paper published in the Proceedings of SIMS 2012, published by Orkustofnun, National Energy Authority Iceland, ISBN: 978-9979-68-318-6, electronically available at <http://www.scansims.org/sims2012/SIMS2012.pdf>.

References

- [1] Altas IH, Sharaf AM. A photovoltaic array simulation model for matlab-simulink GUI environment. In Proceedings of the ICCEP International Conference on Clean Electrical Power; 2007 May 21023; c2007. P.341–345. Doi:10.1109/ICCEP.2007.384234.
- [2] Buresch, M. *Photovoltaic Energy Systems Design and Installation*. New York: McGraw-Hill Book Co; 1983. 347 p.
- [3] Villalva MG, Gazoli JR, Filho ER. Comprehensive approach to modeling and simulation of photovoltaic arrays. *IEEE Transactions on Power Electronics*, 2009; 24(5): 1198–1208. Doi: 10.1109/TPEL.2009.2013862
- [4] Petcut FM, Dragomir TL. Solar cell parameter identification using genetic algorithms. *Journal of Control Engineering and Applied Informatics*, 2010; 12(1): 30–37
- [5] Carrero C, Amador J, Arnaltes S. A single procedure for helping PV designers to select silicon PV modules and evaluate the loss resistances. *Journal of Renewable Energy*, 2007; 32(15): 2579–2589. Doi: 10.1016/j.renene.2007.01.001
- [6] Guasch D, Silvestre S. Dynamic battery model for photovoltaic applications. *Progress in Photovoltaics: Research and Applications*, 2003; 11(3): 193–206. Doi: 10.1002/ppv.480
- [7] Tremblay O, Dessaint L. Experimental validation of a battery dynamic model for ev applications. *World Electric Vehicle Journal*, 2009; 3: 10–15
- [8] Najafi M, Nikoukhan R. Modeling and simulation of differential equations in scicos. In The Modelica Association, editor, *Modelica Conference*, 2006.
- [9] Lundvall H, Fritzson P, Bachmann B. Event handling in the openmodelica compiler and runtime system. In the *46th Conference on Simulation and Modeling (SIMS 2005)*, Trondheim, Norway, 2005.
- [10] Deshmukh MK, Deshmukh SS. Modelling of hybrid renewable energy system. *Renewable And Sustainable Energy Reviews*, 2008; 12(1): 235–249. Doi: 10.1016/j.rser.2006.07.011
- [11] Soon JJ, Low KS. Photovoltaic model identification using particle swarm optimization with inverse barrier constraints. *IEEE Transactions on Power Electronics*, 2012; 27(9): 3975–3983. Doi: 10.1109/TPEL.2012.2188818
- [12] Ishaque K, Salam Z, Taheri H. Simple, fast and accurate two-diode model for photovoltaic modules. *Journal of Solar Energy Materials and Solar Cells*, 2011; 95(2): 586–594. Doi: 10.1016/j.solmat.2010.09.023
- [13] Kou Q, Klein SA, Beckman WA. A method for estimating the long-term performance of direct-coupled PV pumping systems. *Journal of Solar Energy*, 1998; 64(1-3): 33–40. Doi: 10.1016/S0038-092X(98)00049-8
- [14] Middledbrook RD, Cuk S. A general unified approach to modelling switching-converter power stages. In: *Proceedings of IEEE Power Electronics Specialist conference*, 1976 June 08010; Cleveland; c1976. P. 18-34
- [15] Beaudin M, Zareipour H, Schellenberg A, Rosehart W. Energy storage for mitigating the variability of renewable electricity sources: An updated review. *Energy for Sustainable Development*, 2010; 14(4): 302–314. Doi: 10.1016/j.esd.2010.09.007
- [16] Divya KC, Ostergaard J. Battery energy storage technology for power systems an overview. *Electric Power Systems Research*, 2009; 79(4): 511–520. Doi: 10.1016/j.eprsr.2008.09.017
- [17] SUNDIALS. Suite of nonlinear and differential/algebraic equation solver [Internet]. Available from: <https://computation.llnl.gov/casc/sundials/>
- [18] DAETS. Defferential-algebraic equations by taylor series [Internet]. Available from: www.cas.mcmaster.ca/nedialk/dates/
- [19] DASSL. Defferential algebraic system solver [Internet]. Available from: www.cs.ucsb.edu/cse/
- [20] Modelica. Modelica association [Internet]. Available from: www.modelica.org
- [21] Fritzson, P. *Introduction to modelling and Simulation of Technical and Physical Systems with Modelica*. New York: John Wiley & Sons; 2011. 232p.
- [22] OpenModelica. Open-source modelica-based modelling and simulation env [Internet]. Available from: www.openmodelica.org
- [23] Kyocera. KC200GT, high efficiency multicrystal photovoltaic module [Internet]. [cited 2012]. Available from: www.kyocerasolar.com/assets/001/5195.pdf
- [24] Panasonic. LC-R127R2PG, panasonic batteries [Internet]. Available from: www.farnell.com/datasheets/1624915.pdf

Hydro Power Systems: Scripting Modelica Models for Operational Studies in Education

Dietmar Winkler^{*}, Bernt Lie

Faculty of Technology, Telemark University College, P.O. Box 203, N-3901 Porsgrunn, Norway;

^{*}*Dietmar.Winkler@hit.no*

Simulation Notes Europe SNE 23(3-4), 2013, 179 - 184
DOI: 10.11128/sne.23.tn.10214
Received: April 10, 2013 (Selected SIM 2012 Postconf. Publ.);
Revised Accepted: November 15, 2013;

Abstract. Telemark University College is offering a master degree program called 'Systems and Control Engineering'. Most students of that program have a background in either electrical, mechanical, control engineering or a combination of those. Since Norway covers about 99% of its electrical energy demand using hydro-electric power plants it is natural to also educate master students in the subject of hydro power systems.

About three years ago the Telemark University Colleges started a cooperation with the Norwegian power company 'Skagerak Energi' in order to offer real-life projects for students and to establish a new teaching course for second year master students called 'Modelling and Simulation of Hydro Power Systems'. That course teaches the students the basic principles of hydro-electric power generation starting the prediction of precipitation "down" to the distribution of electrical power in the grid with other loads and consumers connected to it.

This paper presents the teaching approach we have taken so far and our evaluations of opensource tools to be used within the 'Modelling and Simulation of Hydro Power Systems' course. The evaluations were also focused on possibilities of scripting model simulations.

1 Teaching Hydro Power Systems

1.1 Overview

Teaching hydro power systems gives one the great opportunity to deliver combined knowledge of at least three major engineering domains:

- Mechanical engineering
- Electrical engineering
- Control engineering

In detail the course deals with:

- Formulation of mathematical models across different physical domains (e.g., mechanical, electrical, hydrological).
- Introduction to the object-oriented modelling language Modelica.
- Development of a simple hydro power plant model which can be extended to more complex and accurate models.
- The benefit of using object orientation when implementing such models, with special emphasis on how the model can be gradually extended.

We use the modelling language Modelica which was especially designed for models which contain components from different physical domains. The benefit of using Modelica in teaching are for example:

Openness. Students can look at the exact equation based mathematical description of physical systems.

Multi-domain nature. In Modelica it is possible to connect the different domains (e.g., electrical, mechanical, control) within one model in order to get close representation of the real physical system.

Object-orientation. This Enhancing models in a “top-down” manner is very simple. This means students can start working on simple models and increase the level of detail later on.

1.2 Using modelling and simulation in projects

After having learnt about the mathematical and physical theory of hydro power systems, students can now apply that knowledge in working on operational studies. Those studies consist normally of real-life problems which need to be solved. In the past our students have for example worked on “Modelling and Optimisation of Deviation in Hydro Power Production”[1] and “Stability Analysis of AGC in the Norwegian Energy System”[2]. In the latter example it was especially important to use scripting and optimisation tools.

2 Modelling Tool Chain used so far

2.1 The modelling language Modelica

Modelica is a unified object-oriented language for systems modelling. It is developed by the *Modelica Association* which was founded in 1996 and consists of members from industry, university and research organisations.

The *Modelica Association* also develops the free and open-source *Modelica Standard Library* (MSL) which is currently at version 3.2 and consist of 1280 non-trivial models and 910 functions. The MSL makes it possible to generate models of complex systems in a simple and quick manner.

Since Modelica is especially suitable for multi-domain modelling and also because of the transparency of its models we decided to base our ‘Modelling and Simulation of Hydro Power Systems’ course on this powerful modelling language.

2.2 The modelling tools

In the course so far, we were using the commercial modelling tool *Dymola*. There are several reasons for this. One is that our students are mainly engineering students with little programming background. Since our hydro power systems course should mainly concentrate on the modelling and simulation tasks and not so much on the programming side we needed something that the students are comfortable working with and can learn within a reasonable short period of time. Basically this means that we needed a Modelica tool that allows to edit models graphically in a drag-and-drop manner rather than doing textual programming.

Another reason was that for the course we also liked to demonstrate the real power of Modelica with detailed models of a complete hydro power system. For this task we came across the *HydroPowerLibrary* which includes such complex models and easy to use examples.

2.3 Example from the HydroPowerLibrary

A typical example that the students model in the end of the course is a complete hydro power system as depicted in Figure 1 consisting of:

- Reservoir
- Waterway
- Turbine with turbine regulator
- Generator
- Power grid

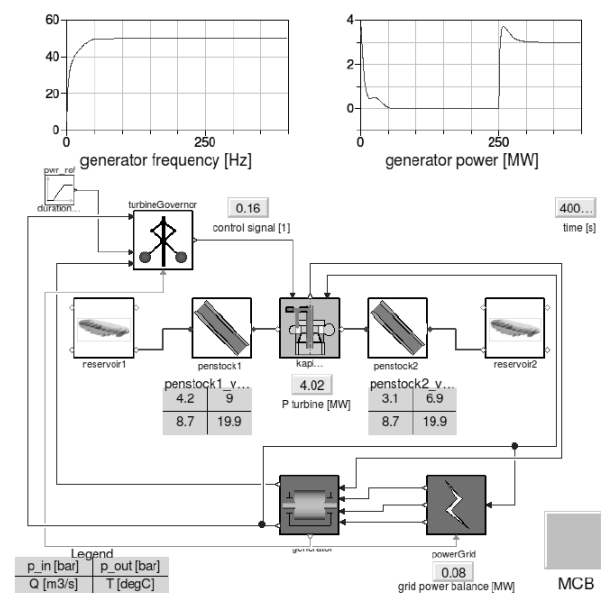


Figure 1: Screenshot from a HydroPowerLibrary example modelled in Dymola.

With such a model one can investigate the process of synchronising a generator that is driven by water turbine to the grid and then look at the power balance.

There are a lot of interesting aspects that the students can look into. E.g.,

- How aggressive should the turbine controller be?
- When can the electrical connection between the electrical generator and the electrical grid be made?
- What happens when it comes suddenly to a load change on the electrical grid?

And those are just some of the many scenarios that one can simulate with this model. One thing all of the different simulation scenarios have in common though is that one would like to automate the simulations with variations of some parameters, i.e., doing parameter sweeps.

2.4 Drawbacks of the commercial tools

Especially the automation of several simulations is something where *Dymola* was kind of weak or cumbersome to use. Also at this point the engineering students begin to see why it might be necessary and more convenient to be able to use a scripting language.

One of the most powerful scripting languages is *Python*. Unfortunately, the tool *Dymola* provides neither a convenient to use own scripting language nor does it provide a direct interface for *Python*. That is why we started to look at alternatives.

Another drawback, we as an academic institution see, is that students should be learning to use tools that they will also be able to use after they finished their degree at our university college. This might be a kind of moral aspect but a valid one none-the-less since many of our students come from countries where they basically can not afford to buy a licence (even when working at a company). It is also important for the students to be able to reproduce the results of their project and study work without restrictions after they have left higher education.

3 Going Open-Source in Modelling and Simulation

Using Modelica as an open modelling language is only the first step. We now looked into open-source tools that allows us to create the models in a convenient way, execute the simulation and do post-processing and optimisation. The most advanced open-source Modelica modelling and simulation tools are currently *OpenModelica* and *JModelica.org*.

First we looked at *OpenModelica* which already provides a graphical editor called “OMEdit”. Unfortunately that editor did not appear to be all that stable at the time of writing so we concentrated more on the script interface. Here *OpenModelica* provides the possibility to use *MetaModelica*, a special language that was developed not just for scripting but also programming the compiler itself [3]. As of version 1.8.1, *OpenModelica* also provides a beta version of *Python* Scripting.

JModelica.org is heavily reliant on *Python* and the whole simulation routine is controlled by using *Python*. Also does *JModelica.org* use the FMI standard [4] that offers the possibility to use exported models from other simulation tools. Thus we decided to start testing *JModelica.org* at first and wait with *OpenModelica* until the *Python* interface has become more mature.

Though not yet feature complete when it comes to the Modelica Language Specification [5] both tools are already powerful enough to simulate hydro power systems. However the remaining part of the paper shall present the experiences we made with *JModelica.org*.

3.1 Simplifying the models

The first thing we tried was exporting a *HydroPowerLibrary* model as a FMU and then later importing this into *JModelica.org*. Unfortunately this was not possible and we concluded this was possibly caused by either a nonstandard export on the one side or a not fully implemented import functionality on the other side.

However we continued with loading a simple *HydroPowerLibrary* model directly. Again this failed because of lacking support of certain functions used in the *HydroPowerLibrary* model.

In the end we decided to build a very simplified model that represents the functionality of a hydro power system consisting of a turbine and generator equivalent that is controlled by a turbine controller and is that then synchronised with the grid.

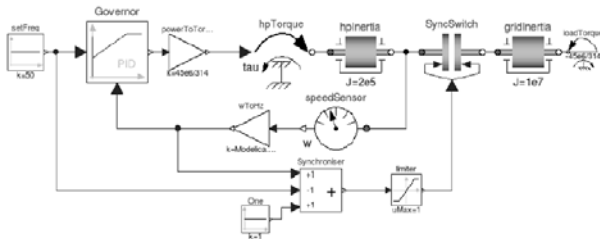


Figure 2: Screenshot of a simple system.

The SimpleSystem is depicted in Figure 2. The idea for this model is that one can look at the turbine and generator unit as torque source *hpTorque* that is used to accelerate their inertia *hpInertia*. The before an electrical generator can be connected with the electrical grid it needs to be synchronised. The process of synchronisation consists for several prerequisites:

- Same direction of rotation
- Same voltage level
- Same frequency

Now the simple model can only be used to simulate the frequency difference and the direction of rotation, i.e., the run-up of the generator. But this is actually sufficient for quite a lot of case studies.

When we only look at the active power balance then we can think of the electrical grid as a large inertia *gridInertia*. If the generated power and the load power are in balance then the grid inertia rotating at a constant frequency of 50Hz. Any electrical load can be represented via the *loadTorque* that can be calculated by:

$$T_{el} = \frac{P_{el}}{\omega^*} - 4\pi \quad (1)$$

where T_{el} stands for the electrical torque, P_{el} for the electrical power and ω^* for the specific rotational velocity (depending on the number of poles in an electrical generator the angular velocity can vary and needs to be taken into account when calculating the rotational energy)

The last central component in the SimpleSystem is the synchronisation switch which is represented by a mechanical clutch *SyncSwitch* which closes when the frequencies of the generator and the grid are near enough. In this case we are starting to close the “switch” when the frequencies are within 1Hz of each other.

3.2 Simulation with JModelica.org

The simplified system from Figure 2 could almost be loaded into *JModelica.org*. The only thing that we needed to fix was that *JModelica.org* did not cope with some of the more advanced initialisation options used in the clutch model but not actually needed in our case.

Error messages that we needed to fix were:

The binding expression of the variable *in-itType* does not match the declared type of the variable

and

String variables are not supported

The simple solution was to simply remove those parts from the models used from the *Modelica Standard Library*. This is best achieved by doing as so called “save total” of the model and then manipulating the used models there. The following script will then generate a successful simulation of the SimpleSystem in *JModelica.org*:

```
# Import the function for compilation
# of models and the FMUModel class
from pymodelica import compile_fmu
from pyfmi import FMUModel

# Import the plotting library
import matplotlib.pyplot as plt

# Define model file name and class name
mofile = 'SimpleSystemTotal.mo'
model_name = 'SimpleSystem'

# Compile model
fmu_name = compile_fmu(model_name, mofile)
```

```

# Load model
grid = FMUModel(fmu_name)

# Simulate the model
res = grid.simulate(final_time=600)
f_gen = res['wToHz.y']
f_grid = res['gridInertia.w']
t = res['time']

# Generating the Plot
plt.figure(1)
plt.title('Synchronising a generator')
plt.ylabel('Frequency [Hz]')
plt.xlabel('Time [s]')
plt.plot(t, f_gen, t, f_grid)
plt.grid()
plt.show()

```

and the resulting plot can be seen in Figure 3.

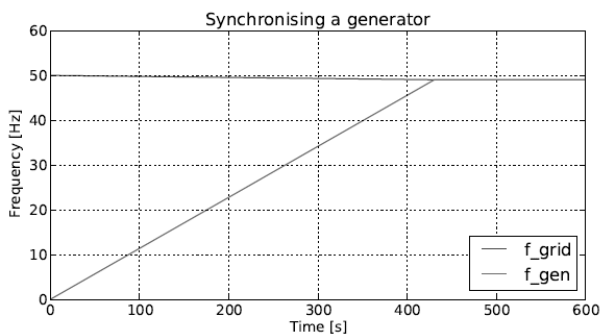


Figure 3: Simulation result from JModelica.org

3.3 Scripting and Optimisation

Now that we can run a simulation an extension for doing a parameter sweep can be easily achieved. It follows a variant of the previous simulation script only this time we run several simulations after each other in order to see the effect of having different hydro plant powers available (in the range of 40MW...140MW):

```

# Import the function for compilation
# of models and the FMUModel class
from pymodelica import compile_fmu
from pyfmi import FMUModel

# Import the plotting library
import matplotlib.pyplot as plt

# Import numpy
import numpy as np

# Define model file name and class name
mofile = 'SimpleSystemTotal.mo'
model_name = 'SimpleSystem'

# Compile model
fmu_name = compile_fmu(model_name, mofile)

# Load model
grid = FMUModel(fmu_name)

# Define initial conditions
p_var = 10
p_min = 40e6
p_max = 140e6
turbine_gain =
np.linspace(p_min, p_max, p_var) /
(2*np.math.pi*50)

# Setup of plot
plt.figure(1)
plt.hold(True)
plt.title('Synchronising a generator')
plt.ylabel('Frequency [Hz]')
plt.xlabel('Time [s]')

# Running the different simulations
for i in range(p_var):
    # Set initial conditions in model
    grid.set('turbineGain', turbine_gain[i])
    # Simulate
    res = grid.simulate(final_time=600)
    # Get Simulation result
    f_gen = res['wToHz.y']
    f_grid = res['gridInertia.w'] /
    (2*np.math.pi)
    t = res['time']
    plt.plot(t, f_gen, t, f_grid)
    plt.grid()
plt.show()

```

Using this code we will get a plot like shown in Figure 4 where the different rising graphs represent the frequencies of the accelerated turbinegenerator unit. For example can one see that the starting power of $P_{gen} = 40W$ is in this case not enough to bring back the grid frequency to 50Hz.

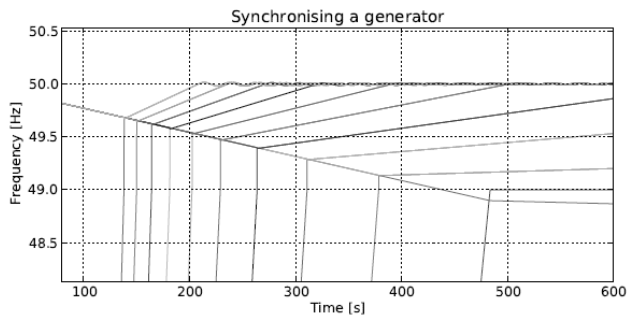


Figure 4: Simulation result of a simulation sweep with varying $P_{gen} = [40 \dots 140]MW$.

4 Conclusion

Our study has shown that it is possible to simulate Hydro Power Systems with open-source tools that also allow for convenient scripting. However there is still room for improvement both, on the compiler side in order to support more Modelica models, especially from the Modelica Standard Library. The other thing that is actually still lacking (but in development) in *JModelica.org* is a graphical editor. Without such a tool it will be hard to convince engineering students of the benefits and possibilities of Modelica and its rich modelling potentials. The scripting itself is thanks to Python very easy and quick to learn and the produced plots are thanks to Matplotlib also more advanced as what *Dymola* would be able to produce. To be honest, the open-source tools are not quite mature enough to allow us to completely switch our courses away from the commercial solutions we are currently using. But at least for student projects (i.e., where students can invest more time and energy) those offer a very interesting alternative and we are definitely continuing the evaluation as the tools keep improving all the time.

Acknowledgement

This contribution is a post-conference publication from SIMS 2012 Conference (53rd SIMS Conference, Reykjavik, Iceland, October 4 - 6, 2012). The contribution is a (partly) modified publication from the paper published in the Proceedings of SIMS 2012, published by Orkustofnun, National Energy Authority Iceland, ISBN: 978-9979-68-318-6, electronically available at <http://www.scansims.org/sims2012/SIMS2012.pdf>.

References

- [1] Winkler D, Thoresen HM, Andreassen I, Perera MAS, Sharefi BR. Modelling and Optimisation of Deviation in Hydro Power Production. In: Modelica Association and Technische Universität Dresden. *Proceedings of the 8th International Modelica Conference*; 2011 March 20022; Dresden, Germany. Doi: 10.3384/ecp1106318
- [2] Andreassen I, Winkler D. Stability Analysis of AGC in the Norwegian Energy System. In: Dahlquist E, editor. *SIMS 2011. Proceedings of the 52nd Scandinavian Conference on Simulation and Modeling*; 2011 Sep 29030; Västerås, Sweden. Scandinavian Simulation Society, Mälardalen University; c2011. P.133–143
- [3] Pop A, Fritzson P. Metamodelica: A unified equation-based semantical and mathematical modeling language. In: Lightfoot D and Szyperski C, editors. *Modular Programming Languages, Lecture Notes in Computer Science*, 4228. Berlin/Heidelberg: 2006. P.211–229.
- [4] Blochowicz T, Otter M, Arnold M, Bausch C, Clauß C, Elmqvist H, Junghanns A, Mauss J, Monteiro M, Neidhold T, Neumerkel D, Olsson H, Peetz J, Wolf S and Atego Systems GmbH, Qtronic Berlin, Fraunhofer Scai, St. Augustin. The functional mockup interface for tool independent exchange of simulation models. In: *Proceedings of the 8th International Modelica Conference*, 2011 March 20022; Dresden, Germany. Linköping Electronic Conference Proceedings; c2011. P.105–114
- [5] Modelica Association, Modelica – A Unified Object-Oriented Language for Physical Systems Modeling – Language Specification, version 3.3 ed., 2012 Sep 05.

Simulation of Heat Radiation Asymmetry with Maple

Ildikó Perjési – Hámori

Department of Mathematics, Pollack Mihály Faculty of Engineering, University of Pécs, Boszorkány u. 2.
7624 Pécs, Hungary, perjesi@pmmik.pte.hu

Simulation Notes Europe SNE 23(3-4), 2013, 185 - 188
DOI: 10.11128/sne.23.sn.10215
Received: October 15, 2012; (Selected MATHMOD 2012
Postconf. Publ.); Revised Accepted: July 15, 2013;

Abstract. A new method was developed for determination of radiated temperature asymmetry. Application of the method resulted in more accurate data, where the plane dividing the two half spaces of the surface element at the test point separates the thermally active (coldest and warmest) surfaces in all cases and determines the irradiation factor through drawing. The present method of asymmetry calculation enables a more accurate calculation of the 'one side radiation asymmetry' parameter and provides a further characteristic parameter of the comfort of rooms in case of cold and warm surfaces being within the same plane.

Introduction

Former calculation procedures of radiation temperature asymmetry are only of approximate validity because of the fixed, i.e. vertical or horizontal position of the plane separating the two half spaces and because of the fact, that in practical calculations the room terminating surfaces beyond those being thermally active and their temperatures are taken into account with their average values [1].

A new method has been developed for determination of radiation temperature asymmetry "giving a result more closer to the reality", where the plane dividing the two half spaces of the surface element at the test point separates the thermally active (coldest and warmest) surfaces in all cases and determines the irradiation factor through drawing [2].

The present method of asymmetry calculation enables a more accurate calculation of the "one side radiation asymmetry" parameter defined by us and being a further characteristic parameter of the comfort of rooms in case of cold and warm surfaces being within the same plane.

Using the Maple computer algebra system there are several possibilities to simulate the radiation asymmetry values in space.

1 Educational Aspects

In the engineering education the students always ask about the usefulness of learning Mathematics. The question is usually justifiable because of the themes of basic Mathematics courses. The lack of time and the weak pre-education of the freshmen students only a few engineering applications are mentioned during the main courses. The solution of a complex engineering problem needs the knowledge of those kinds of – mainly numerical – methods, which are not taught in the courses of Mathematics.

The next model uses only a very few mathematical concepts. These concepts are taught in the first semester of all engineering courses. If there are any possibilities to get to know the elements of a computer algebra system, the simulation of heat radiation asymmetry is a good example to bring Mathematics closer to the engineering students.

2 Problem Definition

2.1 Analytical description

The radiation temperature asymmetry in a given room, in a given test point is the difference of the radiation temperature of the two half spaces, where the space is separated into two half spaces depending the shape and the place of the test body:

$$\Delta t_a = |t_{rad,1} - t_{rad,2}| \quad (1)$$

where Δt_a is the radiation asymmetry (°C), $t_{rad,i}$ is the radiation temperature of the i -th hemisphere (°C).

The radiation temperature is the temperature of the surrounding surfaces at which the irradiative heat exchange of a body is the same, as the irradiative heat exchange of the different platforms.

The equation of the radiation temperature for a half spaces as follows:

$$t_{rad,i} = \sqrt[4]{\sum_{j=1}^n \varphi_{i,j} \cdot T_j^4} - 273 \quad (2)$$

where $\varphi_{i,j}$ is the irradiation coefficient of i -th surface toward j -th surface, n is the number of surfaces in the half sphere, T_j is the temperature of the j -th surface in K.

For the calculation of radiation temperature asymmetry we have to determine the irradiation coefficients of the different boundary surfaces with different temperature from complicated equations.

In the case of different surface temperatures carrying out the calculations is very complicated, and if we get into account the great number of test points in a room, such as this procedure is rather time-consuming.

2.2 Geometrical constructional process

This constructional process uses the hemisphere shape space above the elemental surface to determine the irradiation coefficient (Fig. 1).

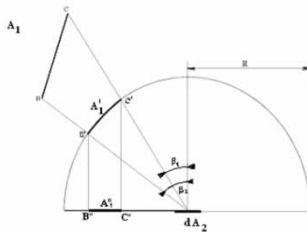


Figure 1: Determination of the irradiation coefficient of dA_2 elemental surface to A_1 surface in a two dimensional model.

In the first step in this process we generate a so called radiation hemisphere above the elemental surface, which is a semicircle in the two dimensional model. The dA_2 elemental surface is the geometrical centre of the basic surface. The radius of the sphere is arbitrary, in our calculation is equal to 1. In the next step we project the A_1 surface across the semicircle into the basic plane.

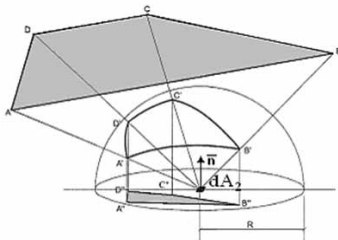


Figure 2: The dA_2 elemental surface irradiation coefficient determination in the three dimensional model.

The geometrical constructional process for determination of radiation coefficient in three dimensional cases is shown on the Fig. 2. The arbitrary corner elements of a room are projected to the hemisphere first, and the intersections and arcs are projected orthogonally to the circle shape basic surface in the next step.

3 Application of Geometrical Construction Process

We used the geometrical process in a special case, where the plane, which divides the space, divides the two surfaces with extreme temperature as well.

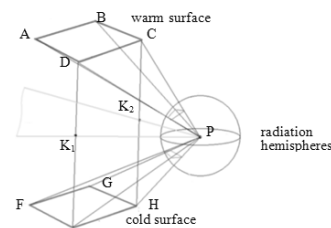


Figure 3: The plane which divided the space into two half spaces.

The plane which divides the space into two half spaces is defined by the observation point P and the K_1 and K_2 middle points of EI and CH sections.

Our goal was to develop a computer simulation, which can determine and visualise the radiation asymmetry values. The used Maple computer algebra system is able to visualise the steps of the construction, to visualise the results and calculate the irradiation coefficients in as many points as we want.

For the calculation we used only the elements of elementary vector algebra (equation of plane, line, determination of intersection of line and plane, equation of the sphere, territory of the rectangle).

For modelling mathematically the procedure, the marks, what will be used in the following part are:

$a, b, c, d, e, f, g, h, p$ are the radius vectors of $A, B, C, D, E, F, G, H, P$ points, respectively.

The steps of the process as follows,

- Determination of plane equation through the K_1, K_2, P points:

$$\mathbf{n}(\mathbf{r} - \mathbf{p}) = 0 \quad (3)$$

where $\mathbf{n} = \overrightarrow{PK_1} \times \overrightarrow{PK_2}$ and \mathbf{r} is the radius vector of the arbitrary point of the plane. The equation of the sphere with radius R , and centre P :

$$(\mathbf{r} - \mathbf{p})^2 = R^2 \quad (4)$$

- Projection of the cold and warm active surfaces and the boundary walls to the sphere ($A'B'C'D'$ points): For A' point firstly the equation of AP line is written down:

$$\mathbf{r} = \mathbf{p} + t(\mathbf{a} - \mathbf{p}), t \in \mathbb{R} \quad (5)$$

The point A' is the intersection of (4) and (5).

- The spherical projected points projection to the K_1K_2P plane ($A''B''C''D''$ points):

Such as A'' point it is the intersection of the plane (3) and line, which is orthogonal to this plane and go through to A' .

- Determination of the $\varphi = \frac{\text{area}(A''B''C''D'')}{R^2 \pi}$ values.

- Calculation of the $t_{rad,i} = \sqrt[4]{\sum_{j=1}^n \varphi_{i,j} \cdot T_j^4} - 273$

and the $\Delta t_a = |t_{rad,1} - t_{rad,2}|$ values.

Using the Maple computer algebra system we simulate the radiation asymmetry values in space, in three different cases.

3.1 Thermally active surfaces in the same plane – radiator under the window

In arbitrary observation points of a room we can calculate all irradiation coefficients, not only for the thermally active surfaces, but for all boundary surfaces as well.

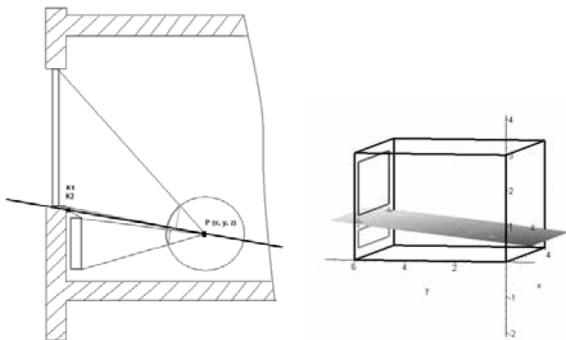


Figure 4. Observation plane in the case of radiator under the window.

The boundary values where the calculations were carried out were as follows: the temperature of the warm surface was 36 OC, the cold surface had 12 oC, and the walls had 18 oC. The sizes of the room in meter were $x=4$, $y=6$, $z=3$. The area of the warm surface (red rectangle) was 2.9×0.5 m², the cold surface (blue rectangle) was 2.9×1.5 m².

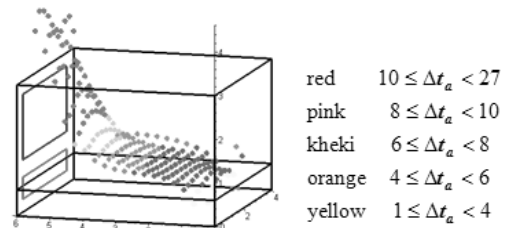


Figure 5. The radiation temperature asymmetry values in 0.6 m height.

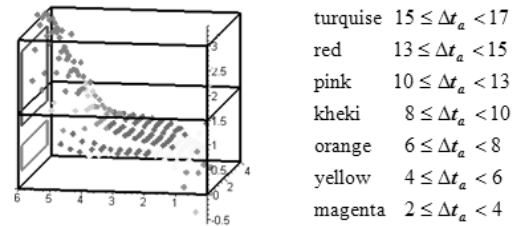


Figure 6. The radiation temperature asymmetry values in 1.5 m height.

In the Figure 5, Figure 6, and the Table 1 are demonstrated, that closer to the active surfaces the asymmetry values increase. The values are in °C. For humans these high values mean discomfort feeling. From this reason, it is not useful to place the bed or the desk close to the window.

	Δt_a (°C) $z=0.6$					
x	y=0.9	y=1.8	y=2.4	y=3.6	y=4.8	y=5.7
0.6	2.11	5.43	5.84	6.72	12.50	24.73
1.2	4.05	5.16	5.71	7.31	13.88	26.70
1.8	4.87	5.18	5.60	7.48	15.34	24.06
2.4	4.66	5.15	5.59	7.38	14.41	22.42
3.0	3.08	5.02	5.65	6.96	12.12	23.72
3.6	6.21	5.11	6.25	7.16	9.40	16.63

Table 1. Each figure and table caption is to be put below the figure, typeset like this caption. Insert each table *inline* and compose good, comprehensive captions.

3.2 Thermally active, parallel surfaces

In this case the thermally active, parallel surfaces are on the opposite side (cold glass wall and wall-heating on winter time or warm glass and cooler on summer time).

In Fig. 7. and 8. it is shown that the absolute values are much more smaller, then it were in the first case, where the active surfaces were at the same plane. It is the reason, why the window and the cooler are usually on the opposite side of a room. The most comfortable place is in the middle of the room.

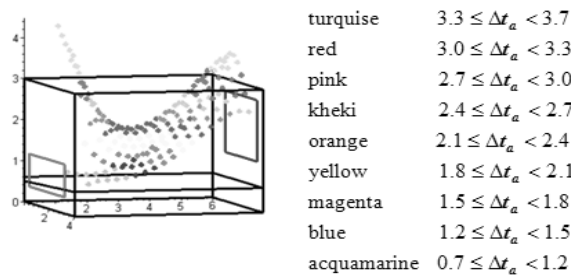


Figure 7: The radiation temperature asymmetry values in case of parallel active surfaces in 0.6 m height.

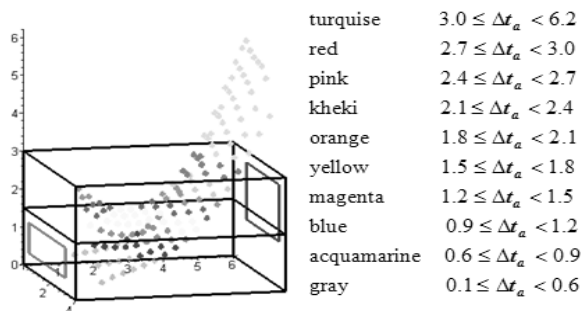


Figure 8: The radiation temperature asymmetry values in case of in case of parallel active surfaces in 1.5 m height.

3.3 Thermally active surfaces in orthogonal arrangements

In this case warm (heating) ceiling or floor and cold (cooler) window (winter period), or cold ceiling or floor and warm window (summer period) are in the studied room (Fig. 9, 10).

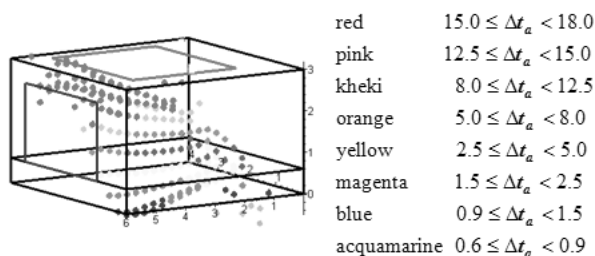


Figure 9: The radiation temperature asymmetry values in case of orthogonal arrangement in 0.6 m height. The values are between 0.6 and 18 °C.

It is interesting to observe, that the values and the differences in Fig. 10 are the highest one, nevertheless the floor heating nowadays are very popular.

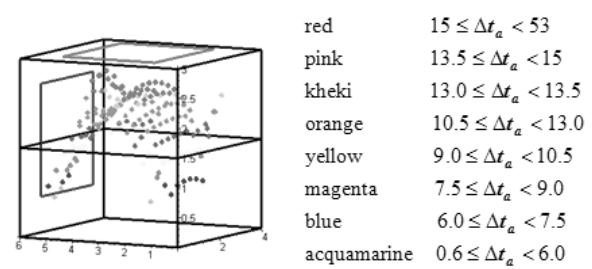


Figure 10: The radiation temperature asymmetry values in case of orthogonal arrangement in 1.5 m height. The values are between 0.6 and 50.3 °C.

4 Conclusion

According to the newest data of literature [4], [5], [6] humans feel unpleasant the asymmetrical radiation, when the differences of the heat stream on different part of the body are higher than 35-40 W/m². When the difference between the temperatures of the surfaces (ceiling, walls, windows) are higher than 5-6 K, there are a danger of discomfort feeling.

Our simulations are useable for house planning. The present values are somewhat different from the values, which were calculated the earlier used methods [3]. It does not cause any problem, because we can standardize the values, and for planning the trends, the fluctuations are interesting for the comfort feeling.

With help of this simulation there are possibilities to determine the zones in a room, where the humans feel themselves much more comfortable.

References

- [1] Bánhidí L, Bakó-Biró Zs. Design and Calculation Possibilities for the Heat Exchange Conditions of the Human Body. *Periodica Polytechnica*. 2000; 44(2):185-193
- [2] Frohner I, Perjési-Hámori I, Bánhidí L. Lassen sich die Diskomfortzonen bei Passivhauser zurückgewinnen? *Internationale Passivhaustagung*, Ludwigshafen, 2005
- [3] Frohner I. PhD Thesis Technical University of Budapest, 2004.
- [4] Jäger A. Optimale Raumtemperaturen liegen zwischen 20 °C, und 26 °C, In: 10 goldene Regeln für gute Lüftung, Klima und Behaglichkeit, von Manfred Stahl, *CCI-Promotor Verlag*, Karlsruhe, 2006.
- [5] Vajda J, Perjési-Hámori I. Two dimensional mathematical model of heat-transmission of one- and double-layer building. *Pollack Periodica*. 2007; 2(3):25-34
- [6] Vajda J. Keine Alternative zur Passivhaustechnik. *HLH*. 2008; 59

Short Remark on Lateral Vibration of Functionally Graded Beams

György Maróti

Department of Mathematics, Pollack Mihály Faculty of Engineering, University of Pécs, Hungary,
Boszorkány u. 2., H-7624 Pécs, Hungary; maroti@epigramma.hu

Simulation Notes Europe SNE 23(3-4), 2013, 189 - 192
DOI: 10.11128/sne.23.sn.10216
Received: October 15, 2012; (Selected MATHMOD 2012 Post-conf. Publ.); Revised Accepted: July 15, 2013;

Abstract. In his study Aydogdu analyzes the vibration of axially functionally graded simply supported beams. The main idea behind his calculation is that the vertical displacement is supposed to have a special form, which transforms the Euler-Bernoulli differential equation for the motion for the lateral vibrations into an exact linear differential equation which depends on the axial coordinate only. In this paper we generalize the method used by Aydogdu and determine the largest function class of the form $w(x,t) = F(x) \cdot G(t)$ for which the same method is applicable.

Introduction

Finding closed-form solutions for the vibration and buckling of the beams has been in the focus of scientific engineering research for a long time. In general case, to reach this goal appears to be not realistic. Several special cases have been examined. A comprehensive summary of the subject can be found in [2]. Further examples can be found in [3], and [4]. The detailed description of usage and programming of Maple can be found in [5] and [6].

1 Basic Model

This article refers to the study of Aydogdu ([1]) in which the equation of motion for the lateral vibrations of axially functionally graded simply supported beams is examined by using the semi-inverse method. The Euler-Bernoulli differential equation for the lateral vibrations of FG beams has the form:

$$\frac{\partial^2}{\partial x^2} \left(E(x) J \left(\frac{\partial^2}{\partial x^2} w(x,t) \right) + \rho A \left(\frac{\partial^2}{\partial t^2} w(x,t) \right) \right) \quad (1)$$

where ρ is the density, A is the cross sectional area, w is the transverse deflection, J is the moment of inertia and $E(x)$ is the elasticity modulus of the beam and t is the time. The cross section area A and the moment of inertia J are assumed to be constant.

In [1] Aydogdu supposes that the vertical displacement has a special form $w(x,t) = W_m \sin(\beta x) \sin(\omega t)$ and points out that the substitution of this form into the Euler-Bernoulli differential equation above transforms it into an exact linear differential equation which depends on the axial coordinate only.

The reconstruction of this calculation is straightforward. Consider the Euler-Bernoulli differential equation of motion for the lateral vibrations and perform the partial derivations.

$$\begin{aligned} &> \frac{\partial^2}{\partial x^2} \left(E(x) J \left(\frac{\partial^2}{\partial x^2} w(x,t) \right) + \rho A \left(\frac{\partial^2}{\partial t^2} w(x,t) \right) \right) = 0 \\ &> \text{value}(\%) \\ &\frac{d^2}{dx^2} (E(x)) J \left(\frac{\partial^2}{\partial x^2} w(x,t) \right) + 2 \left(\frac{d}{dx} E(x) \right) J \left(\frac{\partial^3}{\partial x^3} w(x,t) \right) + \\ &\quad + E(x) J \left(\frac{\partial^4}{\partial x^4} w(x,t) \right) + \rho A \left(\frac{\partial^2}{\partial t^2} w(x,t) \right) = 0 \end{aligned} \quad (2)$$

Next suppose that the vertical displacement is the product of two sinus functions, more specifically let

$$> w(x,t) = W_m \sin\left(\frac{m\pi x}{L}\right) \sin(\omega t) \quad (3)$$

where W_m is the amplitude of the vibrations, m is the half wave number, L is the length and ω is the radial natural frequency of the FG beam. Introducing the notation $\beta = \frac{m\pi}{L}$ we obtain:

> algsubs($\frac{m\pi}{L} = \beta, \%$)

$$w(x, t) = W_m \sin(\beta x) \sin(\omega t) \quad (4)$$

Substituting (4) into (2) the resulting differential equation does not depend on the variable t .

> eval((2), (4))

$$\begin{aligned} & - \left(\frac{d^2}{dx^2} E(x) \right) J W_m \sin(\beta x) \sin(\omega t) - \\ & - 2 \left(\frac{d}{dx} E(x) \right) J W_m \cos(\beta x) \beta^3 \sin(\omega t) + \\ & + E(x) J W_m \sin(\beta x) \beta^4 \sin(\omega t) - \rho A \sin(\beta x) \sin(\omega t) \omega^2 = 0 \end{aligned} \quad (5)$$

> expand($\left(\frac{(5)}{J W_m \sin(\omega t) \beta^2} \right)$)

$$\begin{aligned} & - \left(\frac{d^2}{dx^2} E(x) \right) \sin(\beta x) - 2\beta \left(\frac{d}{dx} E(x) \right) \cos(\beta x) + \\ & + \beta^2 E(x) \sin(\beta x) - \frac{\rho A \sin(\beta x) \omega^2}{J \beta^2} = 0 \end{aligned} \quad (6)$$

Note that the function $w(x, t)$ in (3) has the form $w(x, t) = F(x) \cdot G(t)$. This observation naturally raises the question: how should we choose the functions $F(x)$ and $G(t)$ so that the choice $w(x, t) = F(x) \cdot G(t)$ results in a differential equation which does not depend on variable t ? In other words, denote the class of all functions of the form $F(x) \cdot G(t)$ by C and determine the largest subset of C whose elements transform the Euler-Bernoulli differential equation into a DE which depends on variable x only.

2 Generalization

Proposition 1. For arbitrary function $F(x)$ and for the function

$$G(t) = A \sin(\omega t) + B \cos(\omega t)$$

the choice

$$w(x, t) = F(x) (A \sin(\omega t) + B \cos(\omega t))$$

transforms the Euler-Bernoulli differential equation into a DE, which does not depend on variable t .

Proof. Suppose that $w(x, t)$ has the desired form and substitute it into the Euler-Bernoulli differential equation.

$$> w(x, t) = F(x) (A \sin(\omega t) + B \cos(\omega t)) : \quad (7)$$

> eval((1), (7))

$$\begin{aligned} & \left(\frac{d^2}{dx^2} E(x) \right) J \left(\frac{d^2}{dx^2} F(x) \right) (A \sin(\omega t) + B \cos(\omega t)) + \\ & + 2 \left(\frac{d}{dx} E(x) \right) J \left(\frac{d^3}{dx^3} F(x) \right) (A \sin(\omega t) + B \cos(\omega t)) + \\ & + E(x) J \left(\frac{d^4}{dx^4} F(x) \right) (A \sin(\omega t) + B \cos(\omega t)) - \\ & - \rho A F(x) \omega^2 (A \sin(\omega t) + B \cos(\omega t)) = 0 \end{aligned} \quad (8)$$

Freeze the subexpression $(A \sin(\omega t) + B \cos(\omega t))$ and divide the resulting equation by $J\alpha$, provided $\alpha \neq 0$.

> algsubs($A \sin(\omega t) + B \cos(\omega t) = \alpha, (8)$)

$$\begin{aligned} & \left(\frac{d^2}{dx^2} E(x) \right) J \left(\frac{d^2}{dx^2} F(x) \right) \alpha + 2 \left(\frac{d}{dx} E(x) \right) J \left(\frac{d^3}{dx^3} F(x) \right) \alpha + \\ & + E(x) J \left(\frac{d^4}{dx^4} F(x) \right) \alpha - \rho A F(x) \omega^2 \alpha = 0 \end{aligned} \quad (9)$$

> expand($\frac{\alpha}{J\alpha}$)

$$\begin{aligned} & \left(\frac{d^2}{dx^2} E(x) \right) \left(\frac{d^2}{dx^2} F(x) \right) + 2 \left(\frac{d}{dx} E(x) \right) \left(\frac{d^3}{dx^3} F(x) \right) + \\ & + E(x) \left(\frac{d^4}{dx^4} F(x) \right) - \frac{\rho A F(x) \omega^2}{J} = 0 \end{aligned} \quad (10)$$

This proves Proposition 1.

Although this is not in the focus of our investigations the next proposition determines the general solution of DE.

Proposition 2. The general solution of DE (10) is

$$E(x) = \frac{-C_2 + -C_1 x + \frac{\rho A F(x) \omega^2}{J} \int \int F(x) dx dx}{\frac{d^2}{dx^2} F(x)}$$

Proof. The proof is a simple three step calculation. Maple is used to evaluate the differential equation above after the substitution the supposed value of the function $E(x)$. The resulting expression is huge and far from being perspicuous. This does not mean, however, that Maple cannot simplify it to zero.

$$E(x) = \frac{-C_2 + -C_1 x + \frac{\rho A F(x) \omega^2}{J} \int \int F(x) dx dx}{\frac{d^2}{dx^2} F(x)} : \quad (11)$$

> eval((11), %)

$$\begin{aligned}
 & \left(\frac{\rho A F(x) \omega^2}{\left(\frac{d^2}{dx^2} F(x) \right) J} - \frac{2 \left(-C1 J + \rho A \omega^2 \left(\int F(x) dx \right) \right) \left(\frac{d^3}{dx^3} F(x) \right)}{\left(\frac{d^2}{dx^2} F(x) \right)^2 J} + \right. \\
 & + \frac{2 \left((-C2 + -C1 x) J + \rho A \omega^2 \left(\int \int F(x) dx dx \right) \right) \left(\frac{d^3}{dx^3} F(x) \right)^2}{\left(\frac{d^2}{dx^2} F(x) \right)^3 J} - \\
 & \left. - \frac{\left((-C2 + -C1 x) J + \rho A \omega^2 \left(\int \int F(x) dx dx \right) \right) \left(\frac{d^4}{dx^4} F(x) \right)}{\left(\frac{d^2}{dx^2} F(x) \right)^2 J} \right) \cdot \left(\frac{d^2}{dx^2} F(x) \right) + 2 \left(\frac{-C1 J + \rho A \omega^2 \left(\int F(x) dx \right)}{\left(\frac{d^2}{dx^2} F(x) \right) J} - \right. \\
 & \left. - \frac{\left((-C2 + -C1 x) J + \rho A \omega^2 \left(\int \int F(x) dx dx \right) \right) \left(\frac{d^3}{dx^3} F(x) \right)}{\left(\frac{d^2}{dx^2} F(x) \right)^2 J} \right) \cdot \left(\frac{d^3}{dx^3} F(x) \right) + \\
 & + \frac{\left((-C2 + -C1 x) J + \rho A \omega^2 \left(\int \int F(x) dx dx \right) \right) \left(\frac{d^4}{dx^4} F(x) \right)}{\left(\frac{d^2}{dx^2} F(x) \right) J} \cdot \left(\frac{d^2}{dx^2} F(x) \right) - \frac{\rho A F(x) \omega^2}{J} = 0 \\
 & > \text{simplify}(\%) \\
 & 0 = 0 \quad (13)
 \end{aligned}$$

In the end we show the reverse of Proposition 1. In other words, we prove that the form $G(t) = A \sin(\omega t) + B \cos(\omega t)$ is not only sufficient but also necessary condition for fact that the choice $w(x, t) = F(x) \cdot G(t)$ transforms the Euler-Bernoulli differential equation into a DE, which does not depend in variable t .

Proposition 3. If

$$w(x, t) = F(x) \cdot G(t)$$

and its substitution transforms the Euler-Bernoulli differential equation into a DE, which does not depend in variable t , then $G(t)$ must have the form

$$G(t) = A \sin(\omega t) + B \cos(\omega t).$$

Proof. Consider the function

$$> w(x, t) = F(x) G(t)$$

$$w(x, t) = F(x) G(t) \quad (14)$$

and let us substitute it into

$$> \text{eval}((1), (14))$$

$$\begin{aligned}
 & \left(\frac{d^2}{dx^2} E(x) \right) J \left(\frac{d^2}{dx^2} F(x) \right) G(t) + 2 \left(\frac{d}{dx} E(x) \right) J \left(\frac{d^3}{dx^3} F(x) \right) \cdot \\
 & \cdot G(t) + E(x) J \left(\frac{d^4}{dx^4} F(x) \right) G(t) + \rho A F(x) \left(\frac{d^2}{dt^2} G(t) \right) = 0 \quad (15)
 \end{aligned}$$

All terms on the left hand side of this equation is divisible by $G(t)$ except for the last one. Divide the equation by $J G(t)$ provided that $G(t) \neq 0$.

$$> \text{expand}(\frac{\%}{J G(t)})$$

$$\begin{aligned}
 & \left(\frac{d^2}{dx^2} E(x) \right) \left(\frac{d^2}{dx^2} F(x) \right) + 2 \left(\frac{d}{dx} E(x) \right) \left(\frac{d^3}{dx^3} F(x) \right) + \\
 & + E(x) \left(\frac{d^4}{dx^4} F(x) \right) + \frac{\rho A F(x) \left(\frac{d^2}{dt^2} G(t) \right)}{J G(t)} = 0 \quad (16)
 \end{aligned}$$

The first three terms and the coefficient of the second derivative of $G(t)$ in the numerator of the fourth term do not depend on variable t , which yields that the equation above can be written in the form

$$> A + \frac{B \left(\frac{d^2}{dt^2} G(t) \right)}{G(t)} = 0 : \quad (17)$$

The solution of this differential equation can be easily determined by means of procedure *dsolve*.

$$> \text{dsolve}(\%, G(t))$$

$$G(t) = -C1 \sin \left(\frac{\sqrt{A}}{\sqrt{B}} t \right) + -C2 \cos \left(\frac{\sqrt{A}}{\sqrt{B}} t \right) \quad (18)$$

Introducing the notation $\omega^2 = \frac{A}{B}$ we obtain the desired form.

$$> G(t) = \text{algsubs}(\frac{\sqrt{A}}{\sqrt{B}} t = \omega t, \text{rhs}(\%))$$

$$G(t) = A \sin(\omega t) + B \cos(\omega t)$$

This proofs Proposition 3.

3 Conclusion

The aim of this paper has been to show the usage of Maple general purpose computer algebraic system in the scientific engineering calculations. We have entrusted the performance of all calculation step to Maple. In this way we have used it not only to convert different complex expressions but Maple turned out to be a useful tool in the proofs of propositions.

We have pointed out that the largest class of functions of the form $w(x,t) = F(x) \cdot G(t)$, which transforms the Euler-Bernoulli differential equation for the lateral vibrations of FG beams into an exact linear differential equation depending on the axial coordinate only, consists of the functions $G(t) = F(x)(A \sin(\omega t) + B \cos(\omega t))$. We have also determined closed form solution of the transformed DE.

Acknowledgement

The author expresses his gratitude to Prof. Isaac Elishakoff for his valuable remarks.

References

- [1] Aydogdu M. Semi-inverse Method for Vibration and Buckling of Axially Functionally Graded Beams. *Journal of Reinforced Plastics and Composites*. 2008; 27(7): 683-691. doi: 10.1177/0731684407081369 .
- [2] Elishakoff I. *Eigenvalues of inhomogeneous structures*. City of Publisher: CRC Press; 2000. 752 p.
- [3] Maróti Gy. Finding Closed-Form Solutions of Beam Vibration. *Pollack Periodica*. 2011; 6(1).
- [4] Maróti Gy, Elishakoff E. On Buckling of Axially Functionally Graded Beams. *Pollack Periodica*. 2012; 7(1).
- [5] Maróti Gy, Klincsik, M. *Maple*. Békéscsaba: Livermore; 2006. 392 p.
- [6] Monagan MB, Geddes KO, Heal KM, Laban G, Vorkoetter J, Mccarron J, DeMarco P. *Maple User's Guide*. 2009.

An Agent-based Approach to ARGESIM Benchmark C6 'Emergency Department – Follow-up Treatment' using Anylogic

Katharina Ölsböck^{1*}, Michael Noya¹, Patrick Einzinger^{1,2}

¹ Inst. of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *katharina.oelsboeck@tuwien.ac.at

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

Simulation Notes Europe SNE 23(3-4), 2013, 193 - 194

DOI: 10.11128/sne.23.bn06.10217

Received: September 20, 2013; Revised: October 10, 2013;

Accepted: November 30, 2013;

Abstract. ARGESIM Benchmark C6 'Emergency Department – Follow-up Treatment' on the first glance is a classic discrete process system – with servers (treatment units) and entities (patients). But the different behaviour of doctors causes some modelling difficulties. This contribution presents an agent-based modelling approach, which is much more flexible than the classic DES approach, implemented in the Java-based simulator AnyLogic, which is capable of DES, agent-based, ODE, system dynamics and programming approach.

Simulator. AnyLogic is an object-oriented simulation tool that supports discrete-event, agent-based, and system dynamics simulation. It includes a graphical modeling language and allows the user to extend simulation models with Java code.

Basic Model. According to the C06 specification, several interacting active object classes were defined: Patient, Doctor, InexpDoctor (inexperienced doctors), Registration, XRayUnit and PlasterStaff.

There are 250 predefined patients, belonging to one of 4 different types. The arrival of the patients is modelled as a cyclic timeout-triggered event. On average every 0.3 sec a random patient arrives and receives a registration number. A patient's state reflects his current position and whether he is currently waiting or being treated. The corresponding statechart is depicted in Figure 1.

At every treatment point patients are being treated in the order of their arrival. For example, as a patient enters the emergency department, he receives a new registration number.

If there is at least one patient waiting, the registration object changes its state from "idle" to "working". With a certain probability the patient is assigned either to CW1 or CW2 and receives a new number for the corresponding waiting queue. After processing the patient (which takes a certain amount of time), the registration unit changes its state back to "idle" and checks if there are other patients waiting.

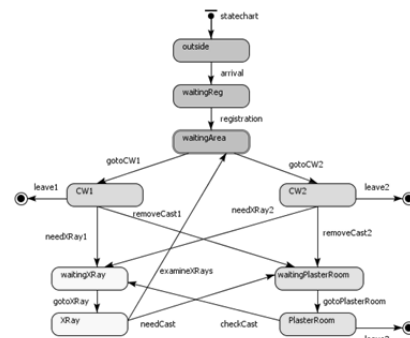


Figure 1: Statechart of a patient. The different states reflect the current position of the patient in the emergency department or whether he is outside.

The other treatment points work in a similar way, but treat patients differently depending on their type. A doctor checks the type of a patient and whether he has already been treated and, consequently, may assign him either to the X-ray room, the plaster room or send him directly home.

The simulation process (Task 1 – Basic Simulation) is visualized on a layout plan of the emergency department (see Figure 2). The agents representing a patient are positioned according to their current state. Every type of patient is depicted with a different color. The two types of doctors are also displayed in either dark (experienced) or light colors (inexperienced).

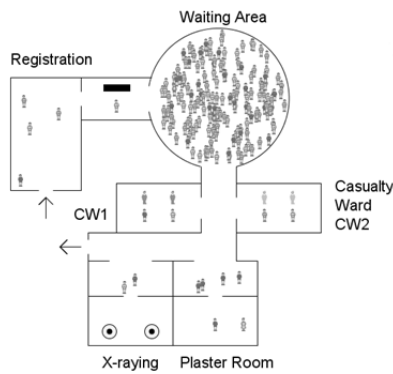


Figure 2: Layout plan of the emergency department showing the current location of patients, doctors and other staff.

Model for Complex Tasks. To provide a solution for the 2nd task *Change of Experienced Doctors*, every doctor has four possible states instead of two: 'idleCW1', 'workingCW1', 'idleCW2' and 'workingCW2' (see Figure 3). One of the experienced and one of the inexperienced doctors is declared 'changing doctor'. The experienced changing doctor changes his state from 'idleCW1' to 'idleCW2', when there are at least 20 patients waiting for CW2 and the corresponding parameter for task b is activated. He then is treating patients assigned to CW2 instead of CW1. Apart from that, his work is unchanged. Only the treatment time per patient of the inexperienced doctor is increasing, when he starts to work in CW1. If the numbers of patients waiting for CW2 is down to 5, the doctors change back to their original workplace.

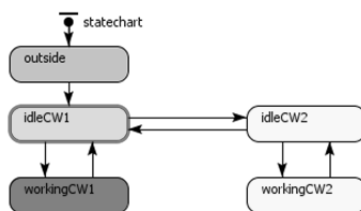


Figure 3: Statechart of a doctor. The states reflect his current workplace and whether he is idle or working.

The 3rd task – *Priority Ranking for Patients* – is implemented by creating a new priority waiting queue at each casualty ward. Patients of type 1 and 3 waiting to enter the casualty ward for the second time receive a number in one of these priority queues. An idle doctor will always prioritize patients waiting in the priority queue over patients in regular queues.

Results. The simulation was run 100 times for each task. Table 1 below shows the results.

1st Task – Basic Simulation: The average overall treatment time is 172 minutes, with significant differences between the patient types.

2nd Task – Change of Experienced Doctors: Exchanging doctors between the casualty wards increases the average overall treatment time while at the same time increasing the standard deviation and the working hours for the emergency department staff. This strategy yields no advantages for patients of any type.

3rd Task – Priority Ranking for Patients: Through the implementation of a priority ranking the average treatment time for patients of type 1 and 3 is reduced by around 70 minutes per patient while the average treatment time for type 2 and 4 is increased by around 30 minutes, thus lowering the standard deviation, while only slightly increasing the working hours for the emergency department staff. Even though patients of type 1 and 3 only constitute 40% of the total, the reduction in their treatment time is significant enough to lower the overall treatment time for all patients.

	Task a	Task b	Task c
Average overall treatment time per patient	172	184	162
Av. treatment time, type 1	235	251	163
Av. treatment time, type 2	135	144	165
Av. treatment time, type 3	246	265	179
Av. treatment time, type 4	126	135	157
Standard deviation	80	92	76
Average closing hour	13:43	14:23	13:52

Table 1: Simulation results.

Conclusions. Agent-based modelling is a useful approach for the simulation of humans working and interacting in a place such as an emergency department. The persons are modelled by individual agents. Their state and progress can therefore be examined at any time and it is easy to visualize all activities.

Model Sources / References. AnyLogic Model files (.alp) and short file documentation can be downloaded (zip format) by EUROSIM societies' members from SNE website, or are available from the author.

- [1] Breitenacker F. Emergency Department – Follow-up Treatment - ARGESIM Comparison C6 – Definition. *SNE Simulation News Europe*. 1992; 2(3):30.

Comparison of MATLAB, Simulink and AnyLogic Approach to ARGESIM Benchmark C9 'Fuzzy Control of a Two Tank System'

Sebastian Hödlmoser*, Florian Kitzler

Inst. of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; * sebastian.hoedlmoser@tuwien.ac.at

Simulation Notes Europe SNE 23(3-4), 2013, 195 - 200

DOI: 10.11128/sne.23.bn09.10219

Received: September 20, 2013; Revised: October 10, 2013;

Accepted: November 30, 2013;

Abstract. This contribution compares modelling and simulation of the ARGESIM Benchmark C9 'Fuzzy Control of a Two-Tank System' with three approaches: (1) programming directly in MATLAB (2) using SIMULINK and the MATLAB Fuzzy Toolbox, and (3) using AnyLogic, a Java-based graphic simulation environment.

The MATLAB implementation required direct programming, whereby the nonlinear ODE model for the two-tank system was simulated by MATLABs ODE solvers, and fuzzification, inference, and defuzzification was programmed by 'pure' vector handling feature. The Simulink implementation is straightforward: graphical blocks for the ODE model, and use of the Fuzzy Toolbox, which supports graphical design of the fuzzy controller. Anylogic offers various graphical modelling methods, also classic block diagrams. But for this comparison the System Dynamics modelling capability was used, which allows a genuine mapping of 'tanks' as reservoir variables; the fuzzy controller was programmed directly in Java embedded into the simulation environment.

The contribution discusses advantages and disadvantages of the modelling approaches – in modelling, in implementation and in simulation and efficiency.

1 Model Description

We consider a two tank system as showed in Figure 1. Tank 1 has an inflow u and is coupled with tank two by a valve v_1 . Tank 2 has a second valve v_2 , acting as a sink. The system is characterized by the nonlinear ODE set:

$$f = 0.06624v_1\sqrt{|x_1 - x_2|}\text{sign}(x_1 - x_2)$$

$$\dot{x}_1 = 0.067u - f$$

$$\dot{x}_2 = f - 0.0605rv_2|x_2|^{0.48}$$

with

$$r = \begin{cases} 1.2 & \text{if } x_2 < 16 \text{ cm} \\ 1 & \text{if } x_2 \geq 16 \text{ cm} \end{cases}$$

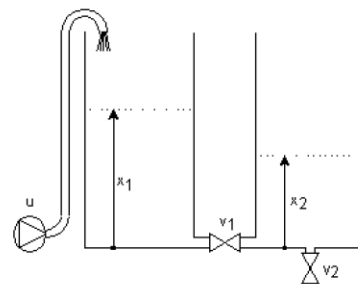


Figure 1: Two tank system with source u , liquid levels x_1, x_2 and valves v_1, v_2 .

The valve positions are $v_1=0.4$ and $v_2=0.3$. The aim of the model is to control the liquid level of the second tank with a fuzzy controller. The model can be subdivided into two different systems. The fuzzy controller (FC1, FC2, FC3) calculates the inflow parameter u for the given membership functions for x_1 (liquid level of tank 1) and ex_2 (difference to the desired liquid level in tank 2). These are given as stepwise linear functions. The membership function of u is calculated combining the given rules (MIN for AND, MAX for OR) using MAX-PROD-Inference.

Defuzzification is done by the centroid method. For calculating the center of gravity, two integrals are approximated using trapezoid rules with stepsize 0.25 (considering the membership functions of u , which change their slopes on vertices *mod* 0.25).

With the calculated value of u we can then solve the differential equation for x_1 and x_2 for the next timestep.

2 Implementation

As mentioned before the two tank system can be subdivided into two different subsystems that can be implemented independently.

First the fuzzy controller has to compute the variable u for given values of x_1 and ex_2 . With x_1 , ex_2 and u we can then solve the ODE system for the next time step. This was implemented using the explicit Euler method with fixed time step of 1 in all three environments.

2.1 MATLAB Model and Implementation

To implement the fuzzy controller we needed to define the membership functions for the variables x_1 , ex_2 and u . These are generalized indicator functions that can reach any value between 0 and 1 . For the variable x_1 we used trapezoid membership functions (Figure 2), whereas triangular membership functions are used for the variables ex_2 and u (FC1). In FC2 and FC3, the membership functions of u are singletons, i.e. functions with a pointwise value of 1 and 0 in between (imagine triangular functions with an 'infinte slope').

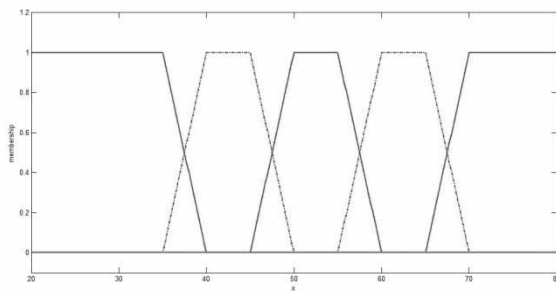


Figure 2: Trapezoidal membership functions for variable x_1 .

		x_1				
		nl	p1	p2	p3	p4
ex_2	p1	p8	p7	p5	p3	nl
	p2	p7	p6	p4	p3	nl
	p3	p7	p5	p3	p2	nl
	nl	p4	p3	p2	p1	nl
	n1	nl	nl	nl	nl	nl

Table 1: Linguistic rules for the output membership function for u of the fuzzy controller.

In Table 1, a set of linguistic rules is given to show how to calculate the membership function for the output variable u . To combine two or more of this rules we need the combination rules for AND (Minimum) and OR (Maximum). Under these combination rules, we reach the membership function p_2 for u as follows:

IF ($ex_2 = nl$ AND $x_1 = p_2$) OR
($ex_2 = p_1$ AND $x_1 = p_3$) THEN $u = p_2$

is translated in MATLAB to

```
mb_FC1(3) =
max([min(mb_ex2(2), mb_x1(3)), ...
min(mb_ex2(3), mb_x1(4))]);
```

Given values for x_1 and ex_2 can be contained in more than just one membership function, with different grades of membership. These values imply more than one membership function for the variable u , which are combined using the MAX-PROD-inference. This means that the different membership functions of u get multiplied by the grade of membership and then combined using the maximum of all these functions. Figure 3 shows an example of such a membership function for u when $x_1=40$ and $ex_2=13$.

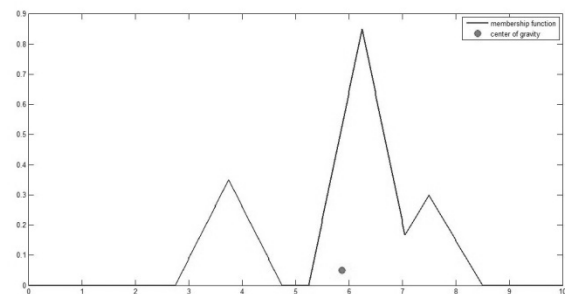


Figure 3: Calculated membership function for output variable u and center of gravity.

The last step of the fuzzy controller is the defuzzification of the membership function of u to compute a value to set the inflow valve. This was done by the centroid method, which sets the sharp value of u to the x-coordinate of the center of gravity of the membership function.

We computed the center of gravity using the trapezoid rule to calculate the occurring integrals (FC1). All other steps were implemented in a MATLAB routine and only required evaluation of membership functions and the MIN and MAX functions. For solving the ODE the Euler-method was used.

2.2 SIMULINK Model / Implementation

In Figure 4 the SIMULINK implementation is displayed. One can obtain the controlling circuit with the control block 'Fuzzy Logic Controller' and the tank system itself (in the 'tank system' subsystem, which we won't discuss any further).

The 'Fuzzy Logic Controller' block calls the fuzzy controller, which is generated with the Fuzzy Toolbox (see below) and saved as a '.fis' file. The coupling of SIMULINK and '.fis' files is just one way to use the Fuzzy Logic Toolbox.

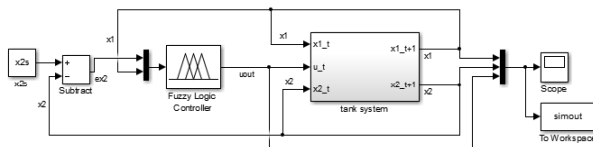


Figure 4: Model structure of the two tank system with fuzzy controller implemented in SIMULINK using the Fuzzy Toolbox.

With the Fuzzy Toolbox it is very easy to define and visualize the used membership functions and combination rules. Figure 5 depicts the GUI of the Fuzzy Toolbox (FIS-Editor, 'Fuzzy Inference System') opened via the command 'fuzzy' in the MATLAB workspace. In the first shell one can specify the number of input and output variables and set the Fuzzy Control options like rules for AND and OR, type of inference (aggregation and implication) and method for defuzzification.

The membership functions can be drawn via a graphic interface as seen in Figure 6. The linguistic rules don't have to be programmed but directly declared in a respective window.

To generate the controllers FC2 and FC3 we had to change the type of the controller to 'sugeno' (instead of default 'mamdani'). This type of controller allows singleton membership functions for the output variable.

The Fuzzy Toolbox Editor allows changing the controller options and comparing the effects on the model without changing the source code, like it would be necessary in the MATLAB version.

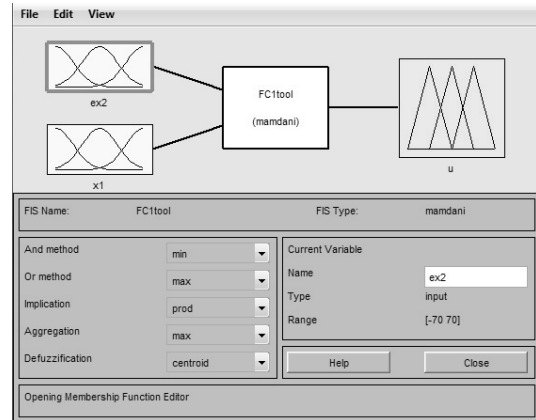


Figure 5: GUI of the Fuzzy Logic Toolbox.

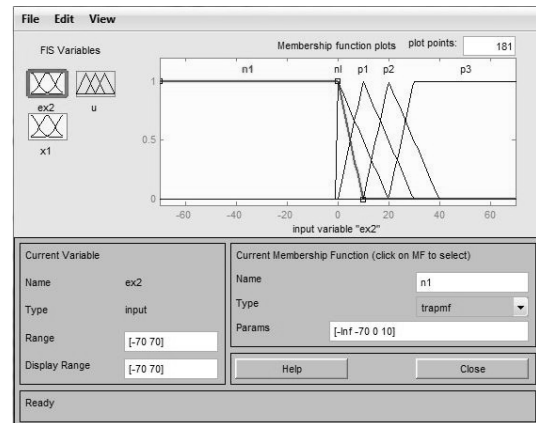


Figure 6: Interface to draw the Membership Functions

2.3 AnyLogic Model / Implementation

As mentioned above, AnyLogic contains a variety of simulation methods. For our purposes, AnyLogic's system dynamics palette offers an intuitive, graphic approach to implement the tank system. Here a flow chart is generated, where the ODEs and other parts of the system lie within.

Our system dynamics model is depicted in Figure 7. The clouds are a source and a sink, in between are two so called stocks, which represent the two tanks (x_1 and x_2). These are coupled with flows which depend on the signs of equations in the ODE set, e.g. the inflow into x_1 is set to $0.067u$ and the flow from x_1 to x_2 is f , giving the change over time \dot{x}_1 (compare the second equation of the ODE set).

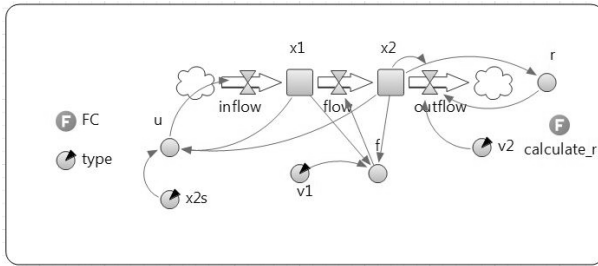


Figure 7: System Dynamics Model of the Two Tank System.

In general, circles are dynamic variables (like u , r and f), circles with black arrows are parameters ($v1$, $v2$ and $x2s$).

The controller is added as the function FC, written in JAVA, and lies within the dynamic variable u . The parameter 'type' denotes which controller should be used (FC1, FC2 or FC3). Besides that, the function FC is basically the JAVA translation of the controller functions used in MATLAB.

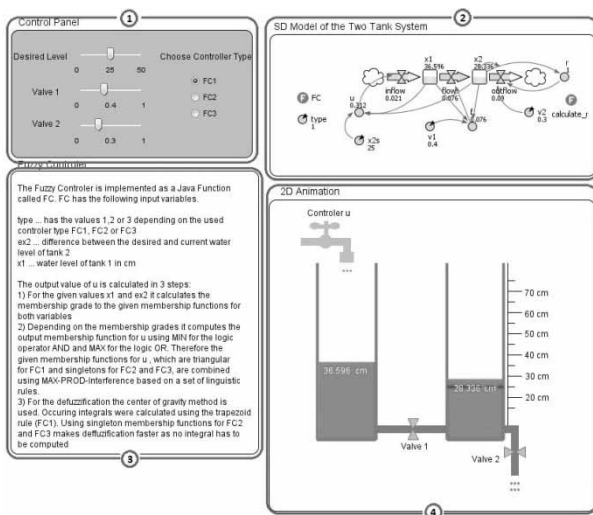


Figure 8: Presentation of the simulation in AnyLogic: (1) control panel, (2) system dynamics model, (3) brief description, (4) 2D visualization.

Here we built a 2D visualization of the two tanks and linked the liquid levels to the sizes of the blue rectangles which represent the liquid. Now instead of just showing plots of the liquid levels over time (compare Figure 11), one can actually see a model of the tanks and how the system changes in real time. Also by adding control units like sliders and radio buttons, we built a panel where one can change the parameters and controllers during simulation.

Figure 8 shows a possible presentation of the simulation. In the left upper corner we have the control panel to set the desired liquid level and valve parameters and choose the type of controller, beneath a brief description of the simulation. On the right side we see the system dynamics model and the corresponding 2D model.

3 Tasks and Results

The following section deals with the tasks of ARGESIM Benchmark C9. Here the performances of the MATLAB and SIMULINK implementation are compared. There are no qualitative differences between the following results and the AnyLogic simulations. However the computation times in AnyLogic are significantly higher.

Task a: Computation of Control Surfaces. The characteristic surfaces of the two fuzzy controllers FC1 and FC2 where computed. Since the characteristic surfaces are the same in both versions (up to little differences due to defuzzification), this was done only in MATLAB (here a plot command had to be programmed; in the Fuzzy Toolbox a surface view command is already implemented, see Figure 10).

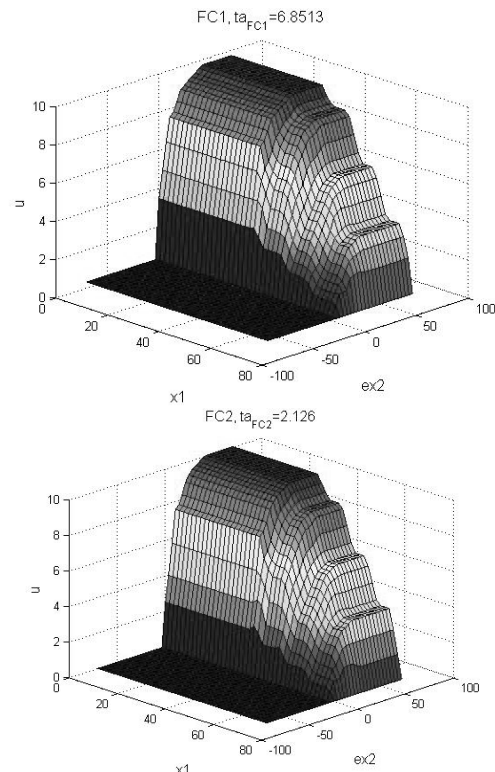


Figure 9: Surface plots for FC1 and FC2 in MATLAB; the characteristic surface with the fuzzy toolbox (nearly) coincides.

Figure 9 shows the visualization for $ex_2 = [-70, 70]$ and $x_1 = [0, 70]$ with both intervals subdivided 40 times. ta_{FC1} and ta_{FC2} denote the computation times for the controller FC1 and FC2, respectively. With our machine we got $ta_{FC1} = 6.8513$ s and $ta_{FC2} = 2.126$ s, giving a ratio of $ta_{FC1}/ta_{FC2} = 3.2226$.

It should be mentioned that the computation time of FC1 increases with decreasing stepsize for the defuzzification (calculation of the center of gravity). However, with smaller stepsizes no significant changes can be observed in the behaviour of FC1 (in this computation the stepsize was 0.25). Since FC2 uses singleton membership functions, no center of gravity has to be calculated, thus making defuzzification easier. A remark: as mentioned, the plots in Figure 9 were programmed in MATLAB. The Fuzzy Toolbox surface viewer opens - without any further programming - a window as depicted in Figure 10.

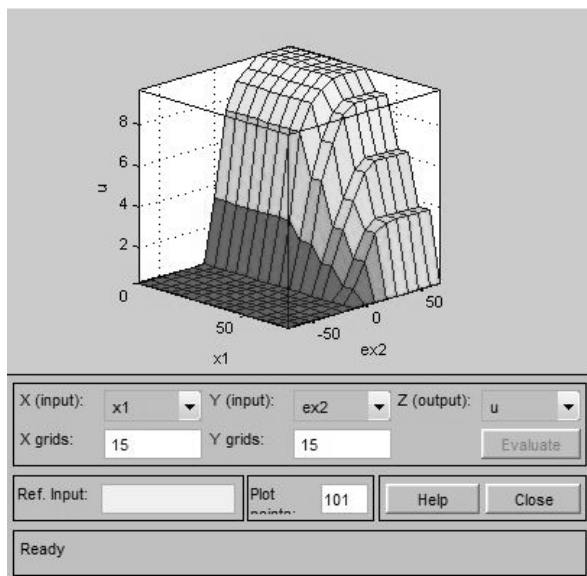


Figure 10: Surface viewer with the Fuzzy Logic Toolbox.

Task b: Simulation with Fuzzy Control. The system was simulated for $t = 1000$ seconds and a desired level $x_{2s} = 25$ cm for the second tank. All calculations were done with both methods. Corresponding computation times where $tb_{FC1} = 3.5353$ s, $tb_{FC2} = 0.78904$ s in MATLAB and $tb_{FC1,Tool} = 1.7009$ s, $tb_{FC2,Tool} = 0.79509$ s in the SIMULINK/Toolbox version. The two ratios are $tb_{FC1}/tb_{FC2} = 4.4805$ and $tb_{FC1,Tool}/tb_{FC2,Tool} = 2.1392$.

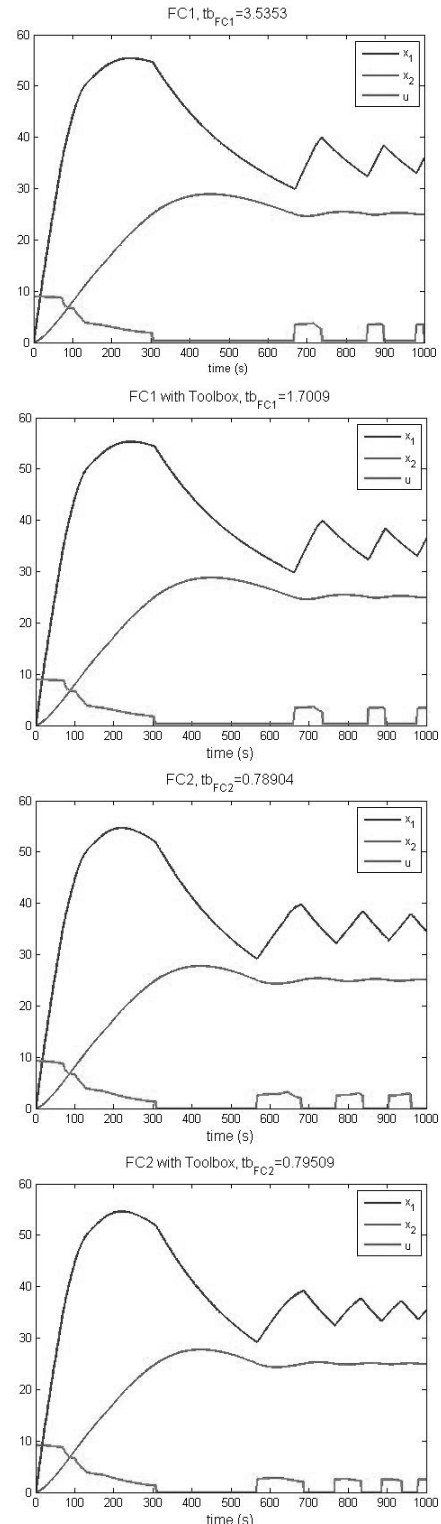


Figure 11: Simulation with FC1/FC2 with MATLAB and SIMULINK/Fuzzy Toolbox; the different performances of the controllers can be observed.

In Figure 11 we can obtain the different performances of the controllers. Although the controller output u is slightly different (due to differences in the calculation of the center of gravity), the behaviour of FC1 stays the same in MATLAB and in SIMULINK. The differences in FC2 are clearer (note the changes at $t \sim 900$). Their origin lies in the inference algorithm and defuzzification of the toolboxes sugeno-type-controller. This controller supports singleton membership functions, but no MAX-PROD-interference. Also the defuzzification is different.

An interesting observation is that the choice of singletons in FC2 instead of triangular membership functions increases the controller's performance. It takes controller FC1 about 100 seconds more to reach a stable liquid level of about 25 cm.

Task b: Simulation with Weighted Fuzzy Control. The rules of FC2 were weighted to gain the controller FC3. In our MATLAB program this happens in the inference algorithm, where corresponding minima were multiplied by the factor 0.1. The Fuzzy Toolbox supports weighting in the rule editor. Repeating the calculation of task a, the computation time for FC3 was $t_{FC3} = 2.1236$.

4 Conclusion

Applying fuzzy logic to control systems bears a lot of advantages. The formulation of membership functions and rules is an intuitive and comparatively accessible way to formalize a problem.

Out of the three methods, implementing a fuzzy controller in MATLAB without the use of any toolbox or software takes the most programming effort. Also some basic knowledge about ODE solving and a good understanding of fuzzy controlling are necessary. Thus the pure MATLAB approach takes the most time to realize, on one hand because of the programming itself, on the other because a certain theoretical background is needed. However this isn't necessarily a disadvantage since this method gives the best understanding of the underlying processes. The pure programming also leads to a high transparency and flexibility of the system.

If the theoretical background is of less interest, the MATLAB Fuzzy Toolbox delivers a convenient way to generate fuzzy controllers.

Via the FIS editor, one can define and edit the membership functions easily, specify the linguistic rules, inference and defuzzification and leave all the programming to the toolbox. All these options and functions can be altered by just a few clicks to compare different controllers and the rule and surface viewer can help get a better understanding of the system. The fuzzy toolbox enables also users with lesser knowledge about fuzzy control theory a rather easy access. Coupled with SIMULINK this is an easy way to implement a fuzzy control system. But while the SIMULINK model gives a nice overview about the control circuit (Figure 4), the implementation of the ODE set can be a rather confusing.

In comparison to MATLAB and SIMULINK, the big strength of AnyLogic lies in its countless ways to present the data and simulations. Instead of relying on usual plots, one can create interactive sheets, build dynamic objects, add external graphics, explanations, comments etc. The parameter manipulation can be done during simulation to get a better understanding about the effects on controlling. Also the system dynamics model is way clearer than the ODE system in SIMULINK (in our case the components of the SD model can be linked - more or less - directly to the schematic model from Figure 7). When it comes to the fuzzy controller itself, in AnyLogic the parts have to be programmed from scratch in JAVA in the same manner as in the MATLAB version. This leads to the same amount of work but again also to a good insight in the control unit.

In summary we believe it is a question of useage which environment is to be preferred. For pure controlling, the Fuzzy Toolbox/SIMULINK offers the most convenient way. For educational or presentation purposes, AnyLogic is the better alternative.

Model sources

All model sources (MATLAB m-files, Simulink .mdl files, and AnyLogic .alp files) and a short file documentation can be downloaded (zip format) by EUROSIM societies' members from SNE website, or are available from the author.

References

- [1] Goldynia J W, Marinits J M. Comparison 9: Fuzzy Control of a Two-Tank System Definition. *SNE Simulation News Europe*. 1996; 6(2): p28 – p29.
- [2] Jaouad S, Ottens M. *Einführung in die Regelungstechnik mit Fuzzy Logik*, Skriptum, Technische Fachhochschule Berlin, 2009.

SNE Simulation News

EUROSIM Data and Quick Info



EUROSIM 2016

9th EUROSIM Congress on Modelling and Simulation

City of Oulu, Finland, September 16-20, 2016

www.eurosim.info

Contents

Info EUROSIM	2
Info EUROSIM Societies	3 - 8
Info ASIM, CAE-SMSG	3
Info CROSSIM, CSSS, DBSS, FRANCOSIM	4
Info HSS, ISCS, LIOPHANT	5
Info LSS, PSCS, SIMS, SLOSIM	6
Info UKSIM, KA-SIM, ROMSIM	7
Info RNSS, Info SNE	8

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 IS SN 2 305-9974) and in a online version (Online IS SN 2 306-0271). With Online SNE the publisher ARGESIM follows the Open Access strategy for basic SNE contributions. Since 2012 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 an additional SNE contributions are available via membership in a EUROSIM society.

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

SNE Reports Editorial Board

EUROSIM Esko Juuso, esko.juuso@oulu.fi
 Borut Zupančič, borut.zupancic@fe.uni-lj.si
 Felix Breitenecker, Felix.Breitenecker@tuwien.ac.at
 ASIM Thorsten Pawletta, pawel@mb.hs-wismar.de
 CAE-SMSG Emilio Jimenez, emilio.jimenez@unirioja.es
 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
 LIOPHANT F. Longo, f.longo@unical.it
 LSS Yuri Merkuryev, merkur@itl.rtu.lv
 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
 KA-SIM Edmond Hajrizi, info@ka-sim.com
 ROMSIM Florin Stanculescu, sflorin@ici.ro
 RNSS Y. Senichenkov, sneyb@dcn.infos.ru

SNE Editorial Office /ARGESIM

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

✉ office@sne-journal.org (info, news)

✉ eic@sne-journal.org Felix Breitenecker (publications)

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. For scientific publications, please contact the EiC.



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 simulation societies and groups to promote 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 fourteen *Full Members* and three *Observer Members*:

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>
LIOPHANT	LIOPHANT Simulation Club <i>Italy & International, Observer Member</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>
KA-Sim	Romanian Society for Modelling and Simulation, <i>Romania, Observer Member</i>
ROMSIM	Romanian Society for Modelling and Simulation, <i>Romania, Observer Member</i>
RNSS	Russian National Simulation Society <i>Russian Federation, 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	Esko Juuso (SIMS) <i>esko.juuso@oulu.fi</i>
Past President	Khalid Al.Begain (UKSIM) <i>kbegain@glam.ac.uk</i>
Secretary	Borut Zupančič (SLO-SIM) <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 2016 will be organised by SIMS in Oulu, Finland, September 16-20, 2016.

Chairs / Team EUROSIM 2016

Esko Juuso EUROSIM President, *esko.juuso@oulu.fi*
Erik Dahlquist SIMS President, *erik.dahlquist@mdh.se*
Kauko Leiviskä EUROSIM 2016 Chair,
kauko.leiviska@oulu.fi

→ www.eurosim.info

✉ office@automaatioseura.fi



EUROSIM Member Societies



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.

→ 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	Ch. Deatcu, christina.deatcu@hs-wismar.de
Treasurer	Anna Mathe, anna.mathe@tuwien.ac.at
Membership Affairs	S. Wenzel, s.wenzel@uni-kassel.de W. Maurer, werner.maurer@zhwin.ch Ch. Deatcu, christina.deatcu@hs-wismar.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 J. Wittmann, wittmann@htw-berlin.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	A. Körner, andreas.koerner@tuwien.ac.at N. Popper, niki.popper@drahtwarenhandlung.at Katharina Nöh, k.noeh@fz-juelich.de
International Affairs	A. Körner, andreas.koerner@tuwien.ac.at 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 December 2013

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@dwh.at A. Körner, andreas.koerner@tuwien.ac.at
Working Groups for Simulation in Business Administration, in Traffic Systems, for Standardisation, for Validation, etc.	

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	M. A. Piera Eroles, MiquelAngel.Piera@uab.es
Vice president	Emilio Jimenez, emilio.jimenez@unirioja.es
Repr. EUROSIM	Emilio Jimenez, emilio.jimenez@unirioja.es
Edit. Board SNE	Emilio Jimenez, emilio.jimenez@unirioja.es
Web EUROSIM	Mercedes Peres, mercedes.perez@unirioja.es

Last data update December 2013



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	Vesna Dušak, vdusak@foi.hr
Vice president	Jadranka Božikov, jbozikov@snz.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 December 2012



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 Alexík, 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 Alexík, alexik@frtk.fri.utc.sk
Edit. Board SNE	Mikuláš Alexík, alexik@frtk.fri.utc.sk
Web EUROSIM	Petr Peringer, peringer@fit.vutbr.cz

Last data update December 2012

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, smnitnet@wxs.nl
Treasurer	W. Smit, smnitnet@wxs.nl
Secretary	W. Smit, smnitnet@wxs.nl
Repr. EUROSIM	A. Heemink, a.w.heemink@its.tudelft.nl
Deputy	W. Smit, smnitnet@wxs.nl
Edit. Board SNE	A. Heemink, a.w.heemink@its.tudelft.nl

Last data update April 2006

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 Pierrevall, pierrevall@imfa.fr*
- Pole Modelling and simulation of continuous systems. Pole Contact: *Yskandar Hamam, y.hamam@esiee.fr*

→ www.eurosim.info

✉ y.hamam@esiee.fr

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

FRANCOSIM Officers

President	Karim Djouani, djouani@u-pec.fr
Treasurer	François Rocaries, f.rocaries@esiee.fr
Repr. EUROSIM	Karim Djouani, djouani@u-pec.fr
Edit. Board SNE	Karim Djouani, djouani@u-pec.fr

Last data update December2012

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

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.at

✉ ISCS / Mario Savastano,
c/o CNR - IRSIP,
Via Claudio 21, 80125 Napoli, Italy

ISCS Officers

President	M. Savastano, mario.savastano@unina.it
Vice president	F. Maceri, Franco.Maceri@uniroma2.it
Repr. EUROSIM	F. Maceri, Franco.Maceri@uniroma2.it
Secretary	Paola Provenzano, paola.provenzano@uniroma2.it
Edit. Board SNE	M. Savastano, mario.savastano@unina.it

Last data update December2010



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

LIOPHANT Officers

President	A.G. Bruzzone, agostino@itim.unige.it
Director	E. Bocca, enrico.bocca@liophant.org
Secretary	A. Devoti, devoti.a@iveco.com
Treasurer	Marina Masseimassei@itim.unige.it
Repr. EUROSIM	A.G. Bruzzone, agostino@itim.unige.it
Deputy	F. Longo, f.longo@unical.it
Edit. Board SNE	F. Longo, f.longo@unical.it
Web EuroSIM	F. Longo, f.longo@unical.it

Last data update December2013



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
Secretary	Artis Teilans, Artis.Teilans@exigenservices.com
Repr. EUROSim	Yuri Merkuryev, merkur@itl.rtu.lv
Deputy	Artis Teilans, Artis.Teilans@exigenservices.com
Edit. Board SNE	Yuri Merkuryev, merkur@itl.rtu.lv
Web EUROSim	Oksana Sosho, oksana@itl.rtu.lv

Last data update December 2013

PSCS – Polish Society for Computer Simulation

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	Tadeusz Nowicki, Tadeusz.Nowicki@wat.edu.pl
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	Tadeusz Nowicki, tadeusz.nowicki@wat.edu.pl
Edit. Board SNE	Zenon Sosnowski, z.sosnowski@pb.edu.pl
Web EUROSim	Magdalena Topczewska m.topczewska@pb.edu.pl

Last data update December 2013

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 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
Vice president	Erik Dahlquist, erik.dahlquist@mdh.se
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, vadim.engelson@mathcore.com

Last data update December 2013



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	Vito Logar, vito.logar@fe.uni-lj.si
Vice president	Božidar Šarler, bozidar.sarler@ung.si
Secretary	Aleš Belič, ales.belic@sandoz.com
Treasurer	Milan Simčič, milan.simcic@fe.uni-lj.si
Repr. EUROSIM	B. Zupančič, borut.zupancic@fe.uni-lj.si
Deputy	Vito Logar, vito.logar@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 December 2013

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
Vice president	A. Orsoni, A.Orsoni@kingston.ac.uk
Secretary	Richard Cant, richard.cant@ntu.ac.uk
Treasurer	A. Orsoni, A.Orsoni@kingston.ac.uk
Membership chair	K. Al-Begain, kbegain@glam.ac.uk
Univ. liaison chair	R. Cheng, rhc@maths.soton.ac.uk
Repr. EUROSIM	Richard Zobel, r.zobel@ntlworld.com
Deputy	K. Al-Begain, kbegain@glam.ac.uk
Edit. Board SNE	Richard Zobel, r.zobel@ntlworld.com

Last data update December 2013

EUROSIM OBSERVER MEMBERS

KA-SIM Kosovo Simulation Society

Kosova Association for Modeling and Simulation (KA – SIM, founded in 2009), is part of Kosova Association of Control, Automation and Systems Engineering (KA – CASE). KA – CASE was registered in 2006 as non Profit Organization and since 2009 is National Member of IFAC – International Federation of Automatic Control. KA-SIM joined EUROSIM as Observer Member in 2011.

KA-SIM has about 50 members, and is organizing the international conference series International Conference in Business, Technology and Innovation, in November, in Durrhës, Albania, an IFAC Simulation workshops in Pristina.

→ www.ubt-uni.net/ka-case

✉ ehajrizi@ubt-uni.net

✉ MOD&SIM KA-CASE
Att. Dr. Edmond Hajrizi
Univ. for Business and Technology (UBT)
Lagjja Kalabria p.n., 10000 Prishtina, Kosovo

KA-SIM Officers

President	Edmond Hajrizi, ehajrizi@ubt-uni.net
Vice president	Muzafer Shala, info@ka-sim.com
Secretary	Lulzim Beqiri, info@ka-sim.com
Treasurer	Selman Berisha, info@ka-sim.com
Repr. EUROSIM	Edmond Hajrizi, ehajrizi@ubt-uni.net
Deputy	Muzafer Shala, info@ka-sim.com
Edit. Board SNE	Edmond Hajrizi, ehajrizi@ubt-uni.net
Web EUROSIM	Betim Gashi, info@ka-sim.com

Last data update December 2013

ROMSIM – Romanian Modelling and Simulation Society

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

→ www.ici.ro/romsim/

✉ sflorin@ici.ro

✉ ROMSIM / Florin Stanculescu,
National Institute for Research in Informatics, Averescu
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
Repr. EUROSIM	Florin Stanciulescu, sflorin@ici.ro
Deputy	Marius Radulescu, mrادulescu@ici.ro
Edit. Board SNE	Florin Stanciulescu, sflorin@ici.ro
Web EUROSIM	Zoe Radulescu, radulescu@ici.ro

Last data update December 2012

RNSS – Russian Simulation Society

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.

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

Last data update February 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 *Modeling 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. In 2013, SNE introduced an extended submission strategy i) individual submissions of scientific papers, and ii) submissions of selected contributions from conferences of EUROSIM societies for post-conference publication (suggested by conference organizer and authors) – both with peer review.

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 2012 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.).

SNE Editorial Office /ARGESIM

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

✉ office@sne-journal.org (info, news)

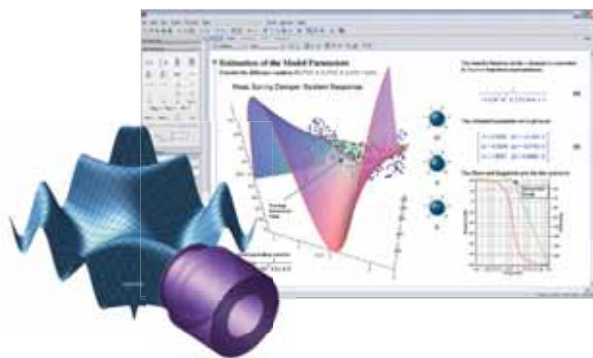
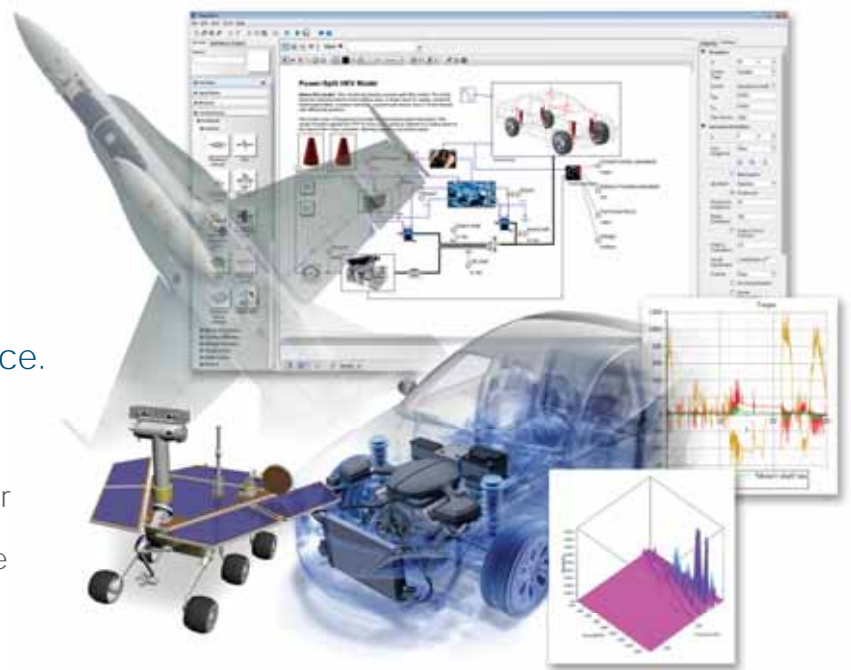
✉ eic@sne-journal.org Felix Breitenacker
(publications)

A modern approach to modeling and simulation



With MapleSim, educators have an industry-proven tool to help bridge the gap between theory and practice.

- MapleSim illustrates concepts, and helps students learn the connection between theory and physical behavior
- A wide variety of models are available to help get started right away



Maple™

MapleSim is built on Maple, which combines the world's most powerful mathematical computation engine with an intuitive, "clickable" user interface.

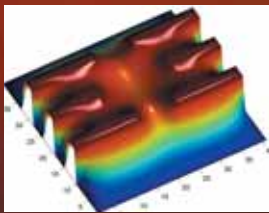
To learn more about how you can reinforce engineering concepts using a combination of theory, simulation, and hardware, view this webinar.

www.maplesoft.com/SNEWebinar

Contact us: +49 (0)241/980919-30

Parlez-vous MATLAB?

Über eine Million Menschen weltweit sprechen MATLAB. Ingenieure und Wissenschaftler in allen Bereichen – von der Luft- und Raumfahrt über die Halbleiterindustrie bis zur Biotechnologie, Finanzdienstleistungen und Geo- und Meereswissenschaften – nutzen MATLAB, um ihre Ideen auszudrücken. Sprechen Sie MATLAB?



Modellierung eines elektrischen Potentials in einem Quantum Dot.

Dieses Beispiel finden Sie unter:
www.mathworks.de/ltc

MATLAB[®]
The language of technical computing

