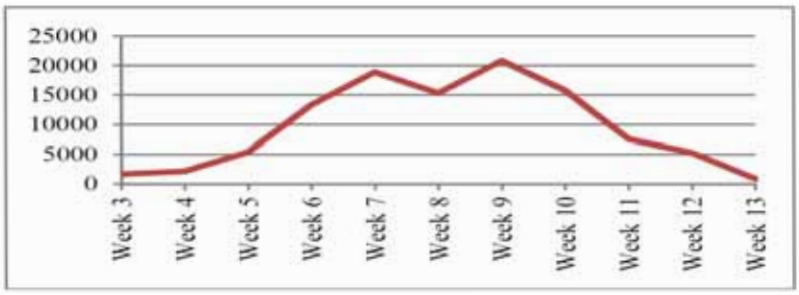
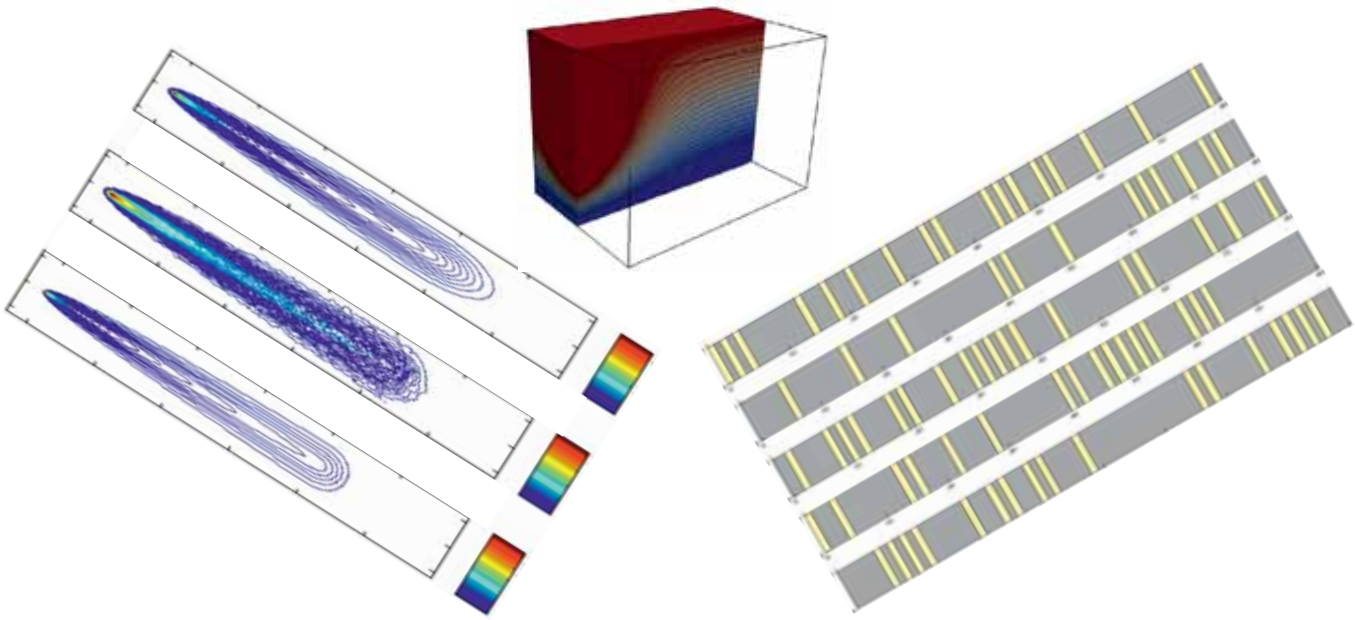
SNE SIMULATION NOTES EUROPE





Volume 24 No.1 April 2014
doi: 10.11128/sne.24.1.1022

Print ISSN 2305-9974
Online ISSN 2306-0271

Journal on Developments and Trends in Modelling and Simulation

Membership Journal for Simulation Societies and Groups in EUROSIM

ARGESIM

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 first issue of SNE Volume 24 continues the extended submission strategy introduced in 2013 - individual submissions of scientific papers, and submissions of selected contributions from conferences of EUROSIM societies for post-conference publication (suggested by conference organizer and authors). This issue publishes post-conference publications from SIMS Conference 2013 (SIMS – Scandinavian Simulation Society), from ERK 2013 Conference (Portoroz, Slovenia) – where SLOSIM, the Slovenian simulation society is running regularly a successful modelling and simulation track, and from ICBTS Conference 2014 (Durres, Albania), where EUROSIM's newest member, KA-SIM, the Kosovo simulation society starts organising sections on modelling and simulation. The variety of content reaches from simulation methodology via mechatronics and fluid dynamics applications to compartment modelling and e-learning.*

Individual submissions complete the range of modelling and simulation themes identification and calibration in health care systems, and two extended benchmark solutions: a comparison of random walk and finite difference approach for fluid dynamics for C19 'Groundwater Flow Pollution', and Petrinet implementations in classical simulators for C4 'Dining Philosophers'

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).

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

Contents SNE 24(1)

SNE doi: 10.11128/sne.24.1.1022

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 submissons of scientific papers are welcome, as well as post-conference publications of contribtions 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 10.1112 8). **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 24(1) April 2014 doi: 10.11128/sne.24.1.1022

→ *www.sne-journal.org*

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

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

SNE Editorial Board

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

Author's Info

Authors are invited to submit contributions which have not been published and have not being 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, eic@sne-journal.og*
✉ 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,* ✉ *eic@sne-journal.org*

Layout / Administration: J. Tanzler, F. Preyser, T. Wobruba; C. Wytrzens, R. Leskovar et al.; Math. Modelling and Simulation Group, Vienna Univ. of Technology, Wiedner Haupstrstasse 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*

Intelligent Methods in Modelling and Simulation of Complex Systems

Esko K. Juuso*

Control Engineering Laboratory Department of Process and Environmental Engineering, P.O.Box 4300, 90014 University of Oulu, Finland; *Esko.Juuso@oulu.fi*

Simulation Notes Europe SNE 24(1), 2014, 1 - 10
DOI: 10.11128/sne.24.on.10221
Received: Jan.10, 2014 (Selected SIMS 2013 Postconf. Publ.);
Accepted: February 15, 2014;

Abstract. Data mining with a multitude of methodologies is a good basis for the integration of intelligent systems. Small, specialised systems have a large number of feasible solutions, but developing truly adaptive, and still understandable, systems for highly complex systems require domain expertise and more compact approaches at the basic level. This paper focuses on the integration of methodologies in the smart adaptive applications. Statistical methods and artificial neural networks form a good basis for the data-driven analysis of interactions and fuzzy logic introduces solutions for knowledge-based understanding the system behaviour and the meaning of variable levels.

Efficient normalisation, scaling and decomposition approaches are the key methodologies in developing large-scale applications. Linguistic equation (LE) approach originating from fuzzy logic is an efficient technique for these problems.

The nonlinear scaling methodology based on advanced statistical analysis is the corner stone in representing the variable meanings in a compact way to introduce intelligent indices for control and diagnostics. The new constraint handling together with generalised norms and moments facilitates recursive parameter estimation approaches for the adaptive scaling.

Well-known linear methodologies are used for the steady state, dynamic and case-based modelling in connection with the cascade and interactive structures in building complex large scale applications. To achieve insight and robustness the parameters are defined separately for the scaling and the interactions.

Introduction

Models understood as relationships between variables are used for predicting of properties or behaviours of the system. Variable interactions and nonlinearities are important in extending the operation areas of control and fault diagnosis, where the complexity is alleviated by introducing software sensors (Figure 1).

Adaptive systems can be developed for nonlinear multivariable systems by various statistical and intelligent methodologies, which are in sensor fusion combined with data pre-processing, signal processing and feature extraction [14]. Fault diagnosis is based on symptoms generated by comparing process models and measurements [15], signal analysis [38], limit checking of measurements [14] and human observations [19]. All these are used in intelligent control and detection of operating conditions, which introduce reasoning and decision making to the smart adaptive systems, whose hybrid nature is seen in literature where these topics are combined from different perspectives.

The linguistic equation (LE) approach originates from fuzzy set systems [35]: rule sets are replaced with equations, and meanings of the variables are handled with scaling functions which have close connections to membership functions [27]. The nonlinear scaling technique is needed in constructing nonlinear models with linear equations [28]. New development methodologies [32,23], improve possibilities to update the scaling functions recursively [33,24].

This paper classifies methodologies and focuses on combining advanced statistical analysis and soft computing in developing LE based applications for complex systems.

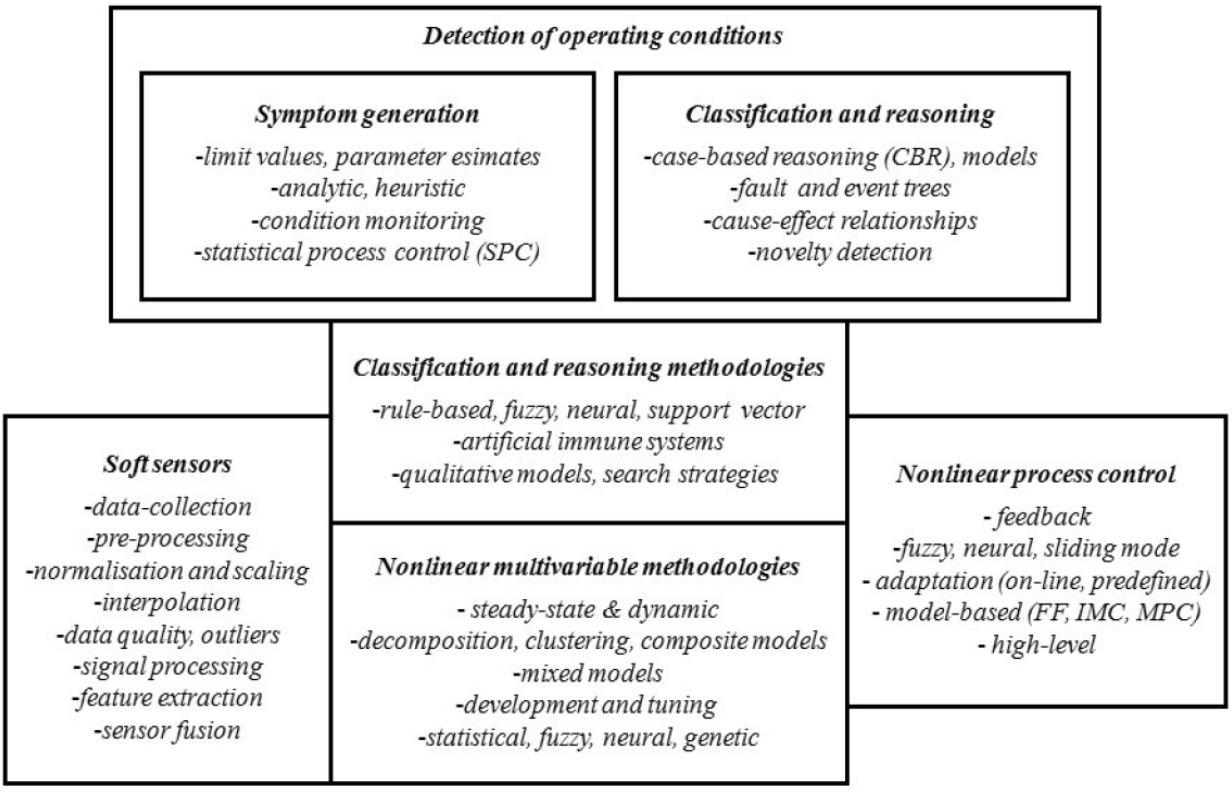


Figure 1: Methodologies for modelling of complex system.

1 Steady-State Modelling

The steady-state simulation models can be relatively detailed nonlinear *multiple input, multiple output* (*MIMO*) models $\boldsymbol{y} = F(\boldsymbol{x})$, where the output vector $\boldsymbol{y} = (y_1, y_2, \dots, y_n)$ is calculated by a nonlinear function F from the input vector $x = (x_1, x_2, \dots, x_m)$. More generally, the relationship could also be a table or a graph. Fuzzy set systems, artificial neural networks and neurofuzzy methods provide additional methodologies for the function $F(x)$.

Statistical modelling in its basic form uses linear regression for solving coefficients for linear functions. In the *response surface methodology* (*RSM*), the relationships are represented with *multiple input, single output* (*MISO*) models, which contain linear, quadratic and interactive terms [5]. Application areas of the linear modelling can also be extended by arbitrary nonlinear models, e.g. semi-physical models, developed by using appropriate calculated variables as inputs, see [39]. Principal component analysis (PCA) compresses the data by reducing the number of dimensions: each principal component is a linear combination of the original variables, usually the first few principal components are used. Various extensions of PCA are referred in [21]. Partial least squares regression (PLS) uses potentially collinear variables [17].

Fuzzy logic emerged from approximate reasoning, and the connection of fuzzy rule-based systems and expert systems is clear, e.g. the vocabulary of AI is kept in fuzzy logic [13]. *Fuzzy set theory* first presented by Zadeh (1965) form a conceptual framework for linguistically represented knowledge. *Extension principle* is the basic generalisation of the arithmetic operations if the inductive mapping is a monotonously increasing function of the input. The interval arithmetic presented by Moore (1966) is used together with the extension principle on several membership α-cuts of the fuzzy number x_j for evaluating fuzzy expressions [6-8]. The fuzzy sets can be modified by intensifying or weakening modifiers [11]. *Type-2 fuzzy* models introduced by Zadeh in 1975 take into account uncertainty about the membership function [42]. Most systems based on interval type-2 fuzzy sets are reduced to an interval-valued type-1 fuzzy set.

Linguistic fuzzy models [12], where both the antecedent and consequent are fuzzy propositions, suit very well to qualitative descriptions of the process as they can be interpreted by using natural language, heuristics and common sense knowledge. *Takagi-Sugeno (TS) fuzzy models* [48], where each consequent $y_i, i = 1, \ldots, n$ is a crisp function of the antecedent variables $\boldsymbol{x}$, can be interpreted in terms of local models. For linear functions, the standard weighted mean inference must be extended with a smoothing technique [2]. *Singleton models*, where the consequents are crisp values, can be regarded as special cases of both the linguistic fuzzy models and the TS fuzzy models. *Fuzzy relational models* [44] allow one particular antecedent proposition to be associated with several different consequent propositions.

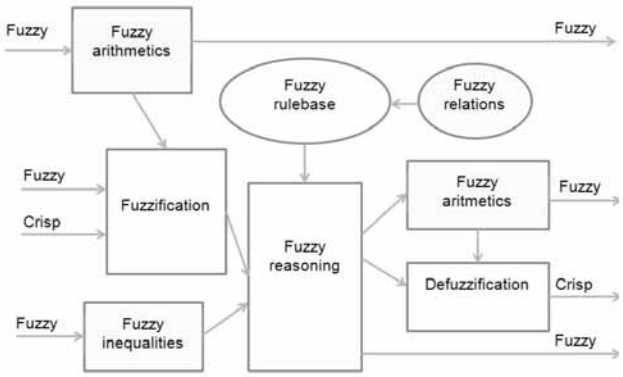


Figure 2: Combined fuzzy modelling.

Several fuzzy modelling approaches are combined in Figure 2: fuzzy arithmetics is suitable both for processing fuzzy inputs for the rule-based fuzzy set system and the fuzzy outputs; fuzzy inequalities produce new facts like $A \leq B$ and $A = B$ for fuzzy inputs A and B; fuzzy relations can be represented as sets of alternative rules, where each rule has a degree of membership.

Artificial neural networks (*ANN*) are used as behavioural input-output models consisting of neurons. Network architectures differ from each other in their way of forming the net input, use of activation functions and number of layers. *Linear networks* correspond to the models with linear terms in RSM models. The most popular neural network architecture is the *multilayer perceptron* (*MLP*) with a very close connection to the *backpropagation learning* [45].

Neurofuzzy systems use fuzzy neurons to combine the weight factors and the inputs (Figure 3). The activation function is handled with the extension principle from the fuzzy input, which is obtained by the fuzzy arithmetics [16]. Also cascade architectures of fuzzy set systems and neural networks are often called neurofuzzy systems. Neural computation is used for tuning fuzzy set systems which can be represented by neural networks, see [20].

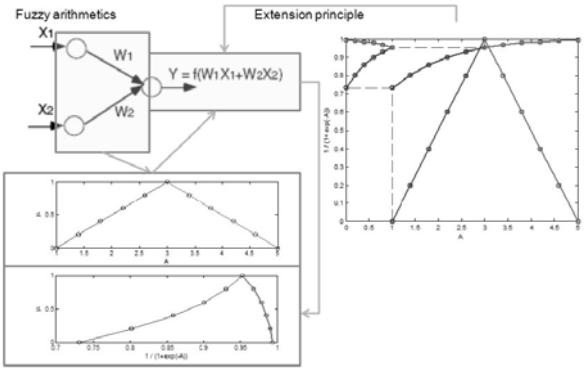


Figure 3: A fuzzy neuron.

A *function expansion* presented in [40] provides flexible way to present several types of black box models by using basis functions, which are generated from one and the same function characterised by the scale (dilation) and location (translation) parameters. The expansion can contain, for example, radial basis functions, one-hidden-layer sigmoidal neural networks, neurofuzzy models, wavenets, least square support vector machines, see [39].

Approximate reasoning based on T-norms and S-norms, also called T-conorms, is an essential part of combining antecedents and rules in fuzzy logic [12]. T-norms and S-norms can be used in neurofuzzy systems if the inputs are normalised to the range [0,1] [16].

2 Decomposition Methodologies

A modelling problem can be divided into smaller parts by developing separate models for independent subprocesses (Figure 4). Cluster analysis can be used in the data-driven approach. Composite local models can be used, and fuzzy set systems provide feasible techniques for handling the resulting partially overlapping models. The system may also include models based on the first principles.

A process plant consists of several processing units interconnected with process streams. *Decomposition* can be continued within process units. In an electric furnace presented in [22], a cylindrically symmetrical one-electrode model was based on two-dimensional areas defined by overlapping rectangular grids where the amount of detail can be increased in selected parts [25]. In addition to spatial or logical blocks decomposed modelling can be based on different frequency ranges.

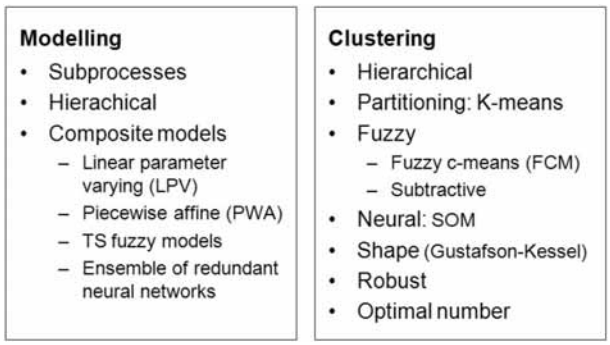


Figure 4: Decomposition for modelling.

Hundreds of clustering algorithms have been developed for the data-driven analysis by researchers from a number of different scientific disciplines. *Hierarchical clustering* groups data by creating a cluster tree, where clusters at one level are joined as clusters at the next higher level. *Partitioning-based clustering algorithms*, e.g. K-means, minimise a given clustering criterion by iteratively relocating data points between clusters until a (locally) optimal partition is attained.[1].

Numerous *fuzzy clustering* algorithms have been proposed and applied to a variety of real-world problems [4]. *Fuzzy c-means (FCM)* clustering is a partitioning-based method: each data point belongs to a cluster to some degree membership. *Subtractive clustering* [10] is an algorithm for estimating the number of clusters and the cluster centres according to the parameters of the algorithm. *Neural clustering* use competitive networks based on competitive layers, e.g. self-organising maps (SOM) [36] have several alternatives for calculating the distance in the competitive layer. The response of a radial basis functions (RBF) neuron is obtained from an exponential function [9].

The clustering algorithms have limitations in shape, cluster centres and generalisation of the results. The algorithm with the standard Euclidean norm imposes a spherical shape on the clusters, regardless of the actual data distribution [2]. Gustafson and Kessel (1979) extended the standard by employing an adaptive distance norm to detect clusters of different geometrical shapes. *Robust clustering*, which is based on a spatial median, is aimed for problems where classical clustering methods are too sensitive to erroneous and missing values [1]. Optimal *number of clusters* is selected iteratively by using some quality criteria, see [49].

Composite local model approach constructs a global model from local models, which usually are linear approximations of the nonlinear system in different neighbourhoods. If the partitioning is based on a measured regime variable, the partitioning can be used in weighting the local models. *Linear parameter varying (LPV) models*, where the matrices of the state-space model depend on an exogeneous variable measured during the operation, are close related to local linear models [40]. *Piecewise affine (PWA) systems* are based on local linear models, more specifically in a polyhedral partition [47]. The models can be state-space models or parametric models. The model switches between different modes as the state variable varies over the partition [40].

Fuzzy models can be considered as a class of local modelling approaches, which solve a complex modelling problem by decomposing into number of simpler understandable subproblems [2,3]. The smoothing problem around the submodel borders of TS fuzzy models needs special techniques, e.g. smoothing maximum [2], or by making the area overlap very strong. *Multiple neural network systems* improve generalisation through task decomposition and an ensemble of redundant networks [46].

A *mixed approach* using both the rigorous first principles and the black box modelling in an integrated environment is an interesting alternative for complex systems, see [41]. [40] classifies the models as a palette of grey shades from white to black into six categories: first principles, identified parameters, semi-physical models, composite models, block oriented models, and black box models. In semi-physical models, linear modelling used together with nonlinear transformations which are based on process insight.

3 Adaptive Nonlinear Scaling Membership

Membership definitions provide nonlinear mappings from the operation area of the (sub)system, defined with feasible ranges, to the linguistic values represented inside a real- valued interval $[-2,2]$. The feasible range is defined by a membership function, and membership functions for finer partitions can be generated from membership definitions [34]. The basic scaling approach presented in [28] has been improved later: a new constraint handling was introduced in [32], and a new skewness based methodology was presented for signal processing in [23].

3.1 Working point and feasible ranges

The concept of feasible range is defined as a trapezoidal membership function. In the fuzzy set theory [51], support and core areas are defined by variable, x_i, specific subsets,

$$\mathrm{supp}(F_j) = \left\{x_j \in U_j | \mu_{F_j}(x_j) > 0\right\}, \quad (1)$$

$$\mathrm{core}(F_j) = \left\{x_j \in U_j | \mu_{F_j}(x_j) = 1\right\}, \quad (2)$$

where U_j is an universal set including F_j; $\mu_{F_j}(x_j)$ is the membership value of x_j in F_j. The main area of operation is the core area, and the whole variable range is the support area.

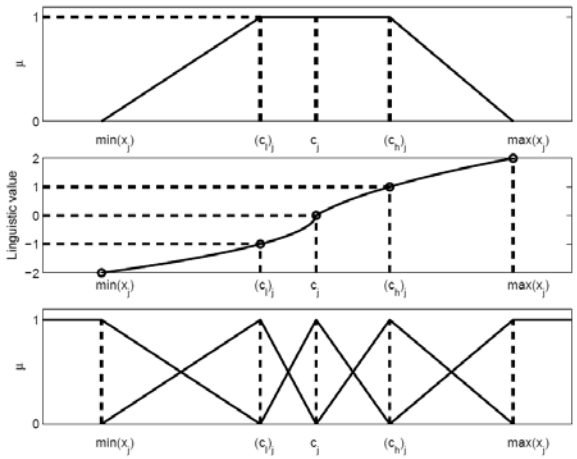


Figure 5: Nonlinear scaling [28]

For applications, a trapezoidal function providing linear transitions between 0 and 1 is sufficient (Figure 5). The corner parameters can be defined on the basis of expert knowledge or extracted from data.

The slope can be different on upper and lower part depending on the linearity or nonlinearity of the system. The complement of a fuzzy set is defined as a subset [51]

$$\bar{F}_j = \left\{x_j \in U_j | \mu_{\bar{F}_j}(x_j) = 1 - \mu_F(x_j)\right\}, \quad (3)$$

where $\mu_{\bar{F}_j}(x_j)$ is the membership value of x_j in $\bar{F}_j$. The membership function of the complement corresponds to the highest and lowest membership functions (Figure 5).

The support area is defined by the minimum and maximum values of the variable, i.e. the support area is $[\min(x_j), \max(x_j)]$ for each variable $j, j = 1,\ldots,m$. The central tendency value, c_j, divides the support area into two parts, and the core area is defined by the central tendency values of the lower and the upper part, $(c_l)_j$ and $(c_h)_j$, correspondingly. This means that the core area of the variable j defined by $[(c_l)_j, (c_h)_j]$ is within the support area.

The corner points can be extracted from existing rule-based fuzzy systems or defined manually. Feasible ranges should be consistent with membership definitions, and therefore they are defined together in the data-driven approach. Earlier the analysis of the corner points and the centre point has been based on the arithmetic means or medians of the corresponding data sets [28]. The norm defined by

$$\left\|{}^{\tau}M_j^p\right\|_p = \left({}^{\tau}M_j^p\right)^{1/p} = \left[\frac{1}{N}\sum_{i=1}^{N}(x_j)_i^p\right]^{1/p}, \quad (4)$$

where $p \neq 0$, is calculated from N values of a sample, τ is the sample time. With a real-valued order $p \in \Re$ this norm can be used as a central tendency value if $\left\|{}^{\tau}M_j^p\right\|_p \in \Re$, i.e. $x_j > 0$ when $p < 0$, and $x_j \geq 0$ when $p > 0$. The norm (4) is calculated about the origin, and it combines two trends: a strong increase caused by the power p and a decrease with the power $1/p$. All the norms have same dimensions as x_j. The norm (4) is a Hölder mean, also known as the power mean. The generalised norm for absolute values $|x_j|$ was introduced for signal analysis in [37].

For variables with only negative values, the norm is the opposite of the norm obtained for the absolute values. If a variable has both positive and negative values, each norm is an average of two norms obtained from data sets made positive and negative. [33]

The operating area of each variable is defined by a feasible range represented with a trapezoidal membership function whose corner points are $\min(x_j), (c_l)_j, (c_h)_j$ and $\max(x_j)$. Warnings and alarms can be generated directly from the degrees of membership of the complement (3).

3.2 Membership definitions

A membership definition is defined as a (nonlinear) mapping of variable values x_j inside its range to $X_j \in [-2,2]$, denoted as *linguistic range*. It more or less describes the distribution of variable values over its range. The range $[-2,2]$ includes the normal operation in the range $[-1,1]$ and the areas with warnings and alarms. The values X_j are called *linguistic values* since the scaling idea originates from the fuzzy set systems: values $-2, -1, 0, 1$ and 2 associated to the linguistic labels are defined with membership functions (Figure 5). The number of membership functions is not limited to five: the values between these integers correspond to finer partitions of the fuzzy set system. The early applications of the linguistic equations used only integer values [27].

In present systems, membership definitions are used in a continuous form consisting of two second order polynomials: one for negative values, $X_j \in [-2, 0)$, and one for positive values, $X_j \in [0, 2]$. The polynomials

$$\begin{aligned} f_j^-(X_j) &= a_j^- X_j^2 + b_j^- X_j + c_j, \quad X_j \in [-2,0), \\ f_j^+(X_j) &= a_j^+ X_j^2 + b_j^+ X_j + c_j, \quad X_j \in [0,2], \end{aligned} \tag{5}$$

should be monotonous, increasing functions in order to result in realisable systems. The upper and lower parts should overlap at the linguistic value 0. [28].

The functions are monotonous and increasing if the ratios

$$\begin{aligned} \alpha_j^- &= \frac{(c_l)_j - \min(x_j)}{c_j - (c_l)_j}, \\ \alpha_j^+ &= \frac{\max(x_j) - (c_h)_j}{(c_h)_j - c_j}, \end{aligned} \tag{6}$$

are both in the range $\left[\frac{1}{3}, 3\right]$ see [25]. If needed, the ratios are corrected by modifying the core $[(c_l)_j, (c_h)_j]$ and/or the support $[\min(x_j), \max(x_j)]$.

Errors are checked independently for f^- and f^+: each error can always be corrected either by moving the corner of the core or the support. In some cases, good results can also be obtained by moving c_j. If these constraints allow a non-empty range, the maximum of the lower limits and the minimum of the upper limit are chosen to define the limits for continuous definitions.

The coefficients of the polynomials can be represented by

$$\begin{aligned} a_j^- &= \frac{1}{2}(1 - \alpha_j^-)\Delta c_j^-, \\ b_j^- &= \frac{1}{2}(3 - \alpha_j^-)\Delta c_j^-, \\ a_j^+ &= \frac{1}{2}(\alpha_j^+ - 1)\Delta c_j^+, \\ b_j^+ &= \frac{1}{2}(3 - \alpha_j^+)\Delta c_j^+, \end{aligned} \tag{7}$$

where $\Delta c_j^- = c_j - (c_l)_j$ and $\Delta c_j^+ = (c_h)_j - c_j$. Membership definitions may contain linear parts if some coefficients α_j^- or α_j^+ equals to one.

For each variable, the membership definitions are configured with five parameters, which can be presented with three consistent sets. The working point (centre point) c_j belongs to all these sets, where the other parameters are:

- the corner points (Figure 5) are good for visualisation;
- the parameters $\{\alpha_j^-, \Delta c_j^-, \alpha_j^+, \Delta c_j^+\}$suit for tuning;
- the coefficients $\{a_j^-, b_j^-, a_j^+, \Delta b_j^+\}$ are used in the calculations.

3.3 Adaptation of nonlinear scaling

Recursive data analysis facilitates the adaptation of the functions to changing operating conditions, also the orders of the norms are re-analysed if needed. The existing scaling functions provide a basis for assessing the quality of new data: outliers should be excluded, but the suspicious values may mean that the operating conditions are changing. In this research, the scaling functions are extended for analysing outliers and suspicious values to select data for the adaptive scaling. Different operating areas can be analysed with clustering, and statistical process control provide additional tools for detecting changes, anomalies and novelties.

The parameters of the nonlinear scaling functions can be recursively updated by including new equal sized subblocks into calculations. The number of samples can be increasing or fixed with some forgetting, and weighting of the individual samples can be used in the analysis. If the definitions should cover all the operating areas, also suspicious values are included as extensions of the support area. In each adaptation step, the acceptable ranges of the shape factors α_j^- and α_j^+ are checked and corrected if needed. The orders $(p_l)_j, (p_0)_j$ and $(p_h)_j$ of the corresponding norms are re-analysed if the distribution is changing considerably with new measurements.

4 Intelligent Systems

Nonlinear models can be constructed by using scaled values in linear modelling based on data and expertise [27,28]. Compact model structures are beneficial in building and tuning dynamic and case-based models for complex systems. The recursive analysis provides new tools for both the adaptation of the scaling functions and the model interactions to changing operating conditions. Linear interactions are used in steady-state models and extended to dynamic systems by parametric structures used in identification. Decomposition of the modelling area is used for case-based systems which can include both steady state and dynamic models. The nonlinear scaling is performed twice: first scaling from real values to the interval $[-2,2]$ before applying linguistic equations, and then scaling from the interval $[-2,2]$ to real values after applying linguistic equations. Variable selection is needed in large-scale systems.

4.1 LE models

The nonlinear scaling with membership definitions transforms the nonlinear model $\boldsymbol{y} = F(\boldsymbol{x})$ to a linear problem. The basic element of a linguistic equation (LE) model is a compact equation

$$\sum_{j=1}^{m} A_{ij} X_j\left(t - n_j\right) + B_i = 0 \tag{8}$$

where X_j is a linguistic value for the variable $j, j = 1 \ldots m$. Each variable j has its own time delay n_j compared to the variable with latest time label. Linguistic values in the range $[-2, 2]$ are obtained from the actual data values by membership definitions. The directions of the interaction are represented by interaction coefficients $A_{ij} \in \Re$.

In the original system [35], the linguistic labels $\{very\ low, low, normal, high,\ \ very\ high\}$ were replaced by numbers $\{-2, -1, 0, 1, 2\}$. The approach was generalized for finer fuzzy partitions in [34]. The bias term $B_i \in \Re$ was first introduced as an additional component in fuzzy LE models [26], and later extended for fault diagnosis systems [28].

The coefficients A_{ij} and B_i in (8) have a relative meaning, i.e. the equation can be multiplied or divided by any nonzero real number without changing the model. A LE model with several equations can be represented as a matrix equation

$$A\boldsymbol{X} + \boldsymbol{B} = 0, \tag{9}$$

where the interaction matrix A contains all coefficients $A_{ij}, i = 1, \ldots, n, j = 1, \ldots, m,$ and the bias vector $\boldsymbol{B}$ all bias terms $B_i, i = 1, \ldots, n$. The time delays of individual variables are equation specific. As linear equations, each model can be used in any direction, i.e. the output variable can be chosen freely.

4.2 Hybrid LE systems

Statistical analysis is an essential part of the development and tuning of the LE systems: data-driven development of the scaling functions, which is based on advanced generalised norms and moments, is suitable for different statistical distributions. The linguistic equation approach originates from the fuzzy set systems which keeps the connections of the methodologies strong. Compact LE models provide a good basis for multi-model systems, where local LE models are combined with fuzzy logic, to handle transitions between models, some special situations and uncertainty with fuzzy set systems. Fuzzy reasoning is an important part of the LE based fault diagnosis and the decision making in the recursive adaptation.

The LE based development of fuzzy systems on any partition can be done if a sufficient number of variables are known or variated by selecting membership locations. Fuzzy set systems, which represent gradual changes by interpolating with membership functions, can be handled by membership definitions and linguistic equations, i.e. the system does not necessarily need any uncertainty or fuzziness. Fuzzy set systems have been moved to higher levels in applications, when first modelling and control, and later also the detection of the operating conditions was realised with the LE approach.

Fuzzy numbers can be handled in LE models by the *extension principle* [29-31. LE models are extended to fuzzy inputs with this approach if the membership definitions, i.e. functions f_j^- and f_j^+ defined by (5) and the corresponding inverse functions, are replaced by the corresponding extensions of these functions. The square root functions are used in the *linguistification* part. The extension principle is needed for fuzzy inputs. The result of the fuzzy extension is a nonlinear membership function for the output even if the membership function of the input is linear. The number of membership levels should increase with the growing fuzziness of the input.

5 Conclusions

Data mining need to be combined with domain expertise to develop practical systems. The LE approach provides a feasible integration framework for practical intelligent applications. The process insight is maintained since all the modules can be assessed by expert knowledge and the membership definitions relate measurements to appropriate operating areas. The nonlinear scaling methodology based on statistical analysis enhanced with domain expertise is the corner stone of the approach, which represents the variable meanings in a compact way to introduce intelligent indices for control and diagnostics.

Different statistical and intelligent methods are used together with the LE approach. The weighting of submodels also is based on the scaled values and fuzzy logic. The cascade and interactive model structures are used in building more complex large scale applications.

Acknowledgement

This contribution is a post-conference publication from SIMS 2013 Conference (54th SIMS Conference, Bergen University College, Norway, October 16-18, 2013). The contribution is a modified publication from the paper published in the Proceedings of SIMS 2013, to be found at http://www.scansims.org/sims2013/SIMS2013.pdf.

References

[1] Äyrämö S, Kärkkäinen T. (2006). *Introduction to partitioning-based clustering methods with a robust example.* Reports of the Department of Mathematical Information Technology Series C. University of Jyväskylä, Software and Computational Engineering; 2006; No. C. 1/2006.

[2] Babuška R. *Fuzzy Modeling and Identification*. Boston: Kluwer Academic Publisher; 1998.

[3] Babuška R, Setnes M, Kaymak U, Verbruggen HB. (1997). Fuzzy Modelling: a Universal and Transparent Tool. In: Yliniemi L, Juuso E, editors. TOOLMET'97. Proceedings of Tool Environments and Development Methods for Intelligent Systems; 1997 April 17-18, Oulu. P.1-27

[4] Bezdek JC. *Pattern Recognition with Fuzzy Objective Function*. New York: Plenum Press. 1981.

[5] Box GEP, Wilson KB. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society*. 1951; Series B, 13(1): 1–45.

[6] Buckley JJ, Feuring T. (2000). Universal approximators for fuzzy functions. *Fuzzy Sets and Systems*. 2000; 113: 411–415.

[7] Buckley JJ, Hayashi Y. (1999). Can neural nets be universal approximators for fuzzy functions? *Fuzzy Sets and Systems*. 1999; 101: 323–330.

[8] Buckley JJ, Qu Y. On using α-cuts to evaluate fuzzy equations. *Fuzzy Sets and System.* 1990 December; 38(3): 309–312. doi: 10.1016/0165-0114(90)90204-J

[9] Chen S, Cowan C, Grant PM. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks.* 1991; 2(2): 302–309. doi: 10.1109/72.80341

[10] Chiu S. Fuzzy Model Identification Based on Clustering Estimation. *Journal of Intelligent & Fuzzy Systems.* 1994; 2(3): 267–278

[11] De Cock M, Kerre EE. Fuzzy modifiers based on fuzzy relations. *Information Sciences*, 2004; 160(1- 4): 173–199. doi: 10.1016/j.ins.2003.09.002

[12] Driankov D, Hellendoorn H, Reinfrank M. *An Introduction to Fuzzy Control*. Berlin: Springer; 1993.

[13] Dubois D, Prade H, Ughetto L. Fuzzy logic, control engineering and artificial intelligence. In: Verbruggen HB, Zimmermann HJ, Babuska R, editors. Fuzzy Algorithms for Control, International Series in Intelligent Technologies; 1999; Kluwer, Boston. P.17–57

[14] Fortuna L, Graziani S, Rizzo A, Xibilia MG. *Soft Sensors for Monitoring and Control of Industrial Processes: Advances in Industrial Control.* New York: Springer. 2007; 270 pp.

[15] Frank PM, Garcia EA, Köppen-Seliger B. Modelling for fault detection and isolation versus modelling for control. Mathematics and Computers in Simulation. 2000; 53: 259–271

[16] Fuller R. Introduction to Neuro-Fuzzy Systems. Advances in Soft Computing. Springer. 2000. 289 pp.

[17] Gerlach RW, Kowalski BR, Wold HOA. Partial least squares modelling with latent variables. Anal. Chim. Acta; 1979; 112(4): 417–421

[18] Gustafson DE, Kessel WC. (1979). Fuzzy clustering with a fuzzy covariance matrix. *In Proceedings of IEEE CDC*, 1979; San Diego. 761–766

[19] Isermann R. Supervision, fault-detection and fault-diagnosis methods - an introduction. Control Engineering Practice. 1997; 5(5): 639–652

[20] Jang JSR. ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Transactions on Systems, Man, and Cybernetics.* 1993; 23(3): 665–685

[21] Jolliffe IT. Principal Component Analysis. 2nd ed. New York: Springer. 2002. 487 pp.

[22] Juuso E. A computer analysis of temperature distribution and energy consumption in a submerged arc furnace used in the production of high-carbon ferrochromium. Acta Universitatis Ouluensis, Series C Technica No. 17, Technica Processionum. 1980; No. 3. 42 pp.

[23] Juuso E, Lahdelma S. Intelligent scaling of features in fault diagnosis. In: Proceedings of the 7th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies; 2010 June 22-24; Stratford-upon-Avon, UK. 2: 1358–1372

[24] Juuso E, Lahdelma S. Intelligent trend indices and recursive modelling in prognostics. In: Proceedings The 8th International Conference on Condition Monitoring and Machinery Failure Prevention Technologies; 2011 June 20-22; Cardiff, UK. 1: 440– 450. BINDT

[25] Juuso EK. Multilayer simulation of heat flow in a submerged arc furnace used in the production of ferroalloys. In: Amouroux M, Jai AE, editors. Control of Distributed Parameter Systems 1989. Selected Papers from the 5th IFAC Symposium, Perpignan, France. 1989 June 26-29; IFAC Symposia Series; 1990; Pergamon, Oxford, UK. 3: 459–464

[26] Juuso EK. Computational intelligence in distributed interactive synthetic environments. In: Bruzzone AG, Kerckhoffs EJH, editors. Simulation in Industry. Proceedings of the 8th European Simulation Symposium, Simulation in Industry, ESS'96; 1996 October 2-5; Genoa, Italy; 157–162

[27] Juuso EK. Fuzzy control in process industry: The linguistic equation approach. In: Verbruggen HB, Zimmermann HJ, Babuska R, editors. *Fuzzy Algorithms for Control, International Series in Intelligent Technologies;* 1999; Kluwer, Boston; 243–300. Doi: 10.1007/978-94-011-4405-6 10

[28] Juuso EK. Integration of intelligent systems in development of smart adaptive systems. *International Journal of Approximate Reasoning;* 2004; 35: 307–337. doi: 10.1016/j.ijar.2003.08.008

[29] Juuso EK. Forecasting batch cooking results with intelligent dynamic simulation. In: Zupančič B, Karba R, Blažič S, editors. Proceedings of the 6th EUROSIM Congress on Modelling and Simulation; 2007 September 9-13; Ljubljana, Slovenia. 2: 8 pp.

[30] Juuso EK. Intelligent modelling of a fluidised bed granulator used in production of pharmaceuticals. In: Bunus P, Fritzson D, Führer C, editors. Conference Proceedings of SIMS 2007 - The 48th Scandinavian Conference on Simulation and Modeling; 2007 October 30-31; Göteborg (Särö); 101–108. Linköping University Electronic Press, Linköping, Sweden.

[31] Juuso EK. Intelligent dynamic simulation of a fed-batch enzyme fermentation process. In: 10th International Conference on Computer Modelling and Simulation, EUROSIM/UKSi; 2008 April 13; Cambridge, UK. 301–306. The Institute of Electrical and Electronics Engineers IEEE. Doi: 10.1109/UKSIM.2008.133

[32] Juuso EK. Tuning of large-scale linguistic equation (LE) models with genetic algorithms. In: Kolehmainen M, editor. Revised selected papers of the International Conference on Adaptive and Natural Computing Algorithms – ICANNGA; 2009; Kuopio, Finland; Lecture Notes in Computer Science, volume LNCS 5495: 161–170. Springer-Verlag, Heidelberg. Doi: 10.1007/978- 3-642-04921-7_17

[33] Juuso EK. Recursive tuning of intelligent controllers of solar collector fields in changing operating conditions. In: Bittani S, Cenedese A, Zampieri S, editors. Proceedings of the 18th World Congress The International Federation of Automatic Control; 2011 August 28 - September 2; Milano; P.12282–12288.

[34] Juuso EK, Bennavail J, Singh M. Hybrid knowledge-based system for managerial decision making in uncertainty environment. In: Carrete NP, Singh MG, editors. Qualitative Reasoning and Decision Technologies. QUARDET'93. Proceedings of the IMACS International Workshop on Qualitative Reasoning and Decision Technologies. 1993 June 16 – 18; Barcelnoa; P. 234–243

[35] Juuso EK, Leiviskä K. Adaptive expert systems for metallurgical processes. In: Jämsä-Jounela SL, Niemi AJ, editors. Expert Systems in Mineral and Metal Processing. Proceedings of the IFAC Workshop; 1991 August 26-28; Espoo, Finland; IFAC Workshop Series; 1992; Pergamon, Oxford, UK; Number 2: 119–124

[36] Kohonen T. *Self-Organizing Maps*. Berlin: Springer. 1995.

[37] Lahdelma S, Juuso E. Signal processing in vibration analysis. In: Proceedings of the Fifth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies; 2008 July 15-18; Edinburgh, UK; 867–878. BINDT.

[38] Lahdelma S, Juuso E. Signal processing and feature extraction by using real order derivatives and generalised norms. Part 1: Methodology. The International Journal of Condition Monitoring; 2011; 1(2), 46–53. doi: 10.1784/204764211798303805.

[39] Ljung L. System Identification - Theory for the User. 2nd edition. Upper Saddle River, N.J.: Prentice Hall. 1999.

[40] Ljung L. Perspectives on system identification. In: Chung MJ, Misra P, editors. Plenary papers, milestone reports & selected survey papers, 17th IFAC World Congress; 2008 July 6-11; Seoul, Korea; P.47–59

[41] Macias-Hernandez J. Opportunities of smart adaptive systems in the oil refining industry. In: Proceedings of Eunite 2004 - European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems; 2004 June 10-12; Aachen, Germany. Wissenschaftsverlag Mainz, Aachen; P.1–12.

[42] Mendel JM. Advances in type-2 fuzzy sets and systems. *Information Sciences*. 2007; 177(1): 84–110

[43] Moore RE. Interval Analysis. Englewood Cliffs, NJ: Prentice Hall. 1966.

[44] Pedrycz W. An identification algorithm in fuzzy relational systems. *Fuzzy Sets and Systems*. 1984; 13(2): 153– 167.

[45] Rummelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Rummelhart DE, McClelland J, editors. Parallel Data Processing; 1986; M.I.T. Press, Cambridge, MA.; P.318–362

[46] Shields MW, Casey MC. A theoretical framework for multiple neural network systems. *Neurocomputing*, 2008; 71: 1462–1476

[47] Sontag E. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions Automatic Control*. 1981; 26(2): 346–358

[48] Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*. 1985; 15(1): 116–132

[49] Windham MP. Cluster validity for fuzzy clustering algorithms. *Fuzzy Sets and Systems*. 1981; 5(2): 177–185

[50] Zadeh LA. Fuzzy sets. *Information and Control*. 1965; 8(June): 338–353

[51] Zimmermann HJ. *Fuzzy set theory and its applications*. Kluwer Academic Publishers. 1992.

PDE Modeling with Modelica via FMI Import of HiFlow3 C++ Components

Kristian Stavåker[1*], Stafan Ronnås[2], Martin Wlotzka[2], Vincent Heuveline[2], Peter Fritzson[1]

[1]Programming Environments Lab, Linköping University, Sweden;
*kristian.stavaker@liu.se

[2]Interdisciplinary Center for Scientific Computing (IWR) Heidelberg University, Germany

Simulation Notes Europe SNE 24(1), 2014, 11 - 20
DOI: 10.11128/sne.24.tn.10223
Received: Jan.10, 2014 (Selected SIMS 2013 Postconf. Publ.)
Accepted: February 15, 2014

Abstract. Despite an urgent need and desire in academia as well as in industry for modeling Partial Differential Equations (PDEs) using the increasingly popular Modelica modeling and simulation language, there is limited support for this available at the moment. In this work, we propose a solution based on importing PDE models with PDE solvers implemented using the general-purpose parallel finite element library HiFlow3 as models into the Modelica environment using the standard Functional Mock-up Interface. In contrast to methods based on language extensions or automatic semidiscretizations in space, this approach requires no change to the language, and enables the use of specialized PDE solvers. Furthermore, it allows for full flexibility in the choice of geometry, model parameters, and space discretization between simulation runs without recompilation needed. This makes it possible to exploit advanced features of the PDE solver, such as adaptive mesh refinement, and to build complex multi-physics simulations by coupling different models, of both PDE and DAE type, in a straightforward manner using Modelica. We illustrate our method with an example that couples a Modelica Proportional-Integral-Derivative controller to a PDE solver for the unsteady heat equation in a 3D domain.

Introduction

We discuss numerical simulation of models that couple Partial Differential Equations (PDEs) and Differential-Algebraic Equations (DAEs) in the context of the Modelica modeling and simulation language [15, 7]. Modelica originated around the idea of solving complex, coupled dynamical systems, which can be described by systems of Ordinary Differential Equations (ODE) or DAE. Up to now, there is only limited support for working with PDEs, despite the fact that the number of Modelica users in academia and in industry has grown significantly lately. One of the first attempts to incorporate PDE support into Modelica is described in [19], [20], and in Chapter 8 of [7], which investigates two different approaches: (1) expressing the PDEs using a combination of new language constructs and a supporting Modelica PDE library using the methodof-lines; (2) exporting the PDE part to an external PDE FEM C++ tool which solves the PDE part of the total problem. Based on this work, an experimental implementation of PDE support was added to the OpenModelica [4] compiler. However, this implementation has not been maintained, even though there have recently been discussions in the OpenModelica community about re-activating these features. Only one simple PDE operator is currently in the Modelica language specification: spatial distribution for 1D PDEs.

In [3] a Modelica library with basic building blocks for solving onedimensional PDE with spatial discretizations based on the method of lines or finite volumes is described. Although this approach is attractive due to its simplicity, it is not clear how it could be extended to higher dimensions, without increasing the complexity significantly. Another approach is described in [11], which extends the modeling language with primitives for geometry description and boundary/initial conditions, and uses an external pre-processing tool to convert the PDE model to a DAE based on the method of lines. In both of these two works, the PDE system is expanded early on in the compilation process. In this way important information of the PDE structure is lost, information that could have been used for mesh refinement and adjustment of the run-time solver.

Another similar option is to use the commercial MapleSim environment [14]: It means to write the PDEs in a Maple component, to export this component to DAE form using a discretization scheme 1 and to use the resulting component in MapleSim, which supports the Modelica language. An overview of how to use Maple and MapleSim together for PDE modeling can be found in [10]. Apart from the cost for licenses, this method again has the same drawback, arising from the loss of information regarding the original model.

In this work, we propose a way to allow for PDE modeling with Modelica by importing C++ components, written with the HiFlow3 multi-purpose finite element software [22], into Modelica using the Functional Mock-Up Interface (FMI) [6] import. FMI is a standard for model exchange and co-simulation between different tools. FMI supports only C but with correct linking it is possible to execute with C++ code. We use the OpenModelica [4] development environment but the same approach can be adapted to other Modelica environments. A similar approach was used in [13, 12], which describes a simulation of the energy supply system of a house using Dymola, ANSYS CFD and the TISC co-simulation environment. Some of the products used in that work are proprietary, however, whereas our environment is based on open standards and open source software. Furthermore, we use the 'model import' approach for the coupling between components, instead of the 'cosimulation' approach applied in those works.

The method described in this paper has several advantages:

1. HiFlow3 is well maintained and has strong support and capabilities for PDE modeling and solving;
2. HiFlow3 and OpenModelica are free to download and use;
3. The PDE structure is not lost but is maintained throughout the actual run-time simulation process. This allows for mesh refinement, solver run-time adjustments, etc.;
4. It is possible to mix PDE and DAE systems in the same system setting. This is also possible in [19].

As a proof-of-concept to demonstrate the Modelica-HiFlow3 integration, we have implemented and tested a coupled model which consists of solving the heat equation in a 3D domain and controlling its temperature via an external heat source. This source consists of a Modelica Proportional-Integral Derivative controller (PID controller), which is taken directly from the Modelica standard library.

The outline of the paper is as follows. In Section 1, we introduce the physical scenario we wish to simulate, and provide a mathematical overview of the two main components in the model. Section 2 describes in detail the realization of the simulation based on coupling of existing software components. In Section 3, we provide numerical simulation results for the example model, followed by a discussion in Section 4, and conclusions in Section 5.

1 Simulation Scenario

We consider the evolution of the temperature distribution in a rectangular piece of copper. Figure 1 shows the setup of the system. A heat sensor is attached on the right side of the copper bar, and on the bottom side there is an adjustable heat source. The system is exposed to environmental influences through time-varying boundary conditions at the top and left sides. The goal is to control the temperature in the material by adjusting the heat source, so that a desired temperature is reached at the point of measurement. The regulation of the heat source is done by a PID controller. It uses the sensor value and a reference temperature to compute the heat source strength. In our simulation, the two entities in this system PID controller copper bar measuring point heat source are realized by reusing existing software components.

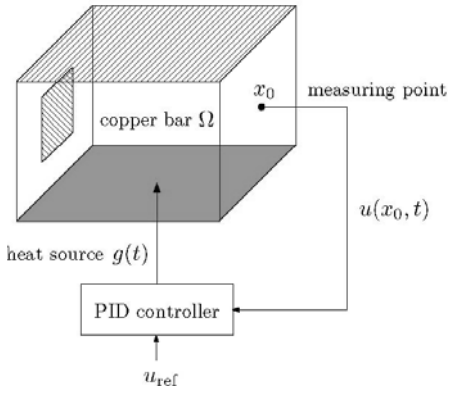


Figure 1: System consisting of a copper bar connected to a temperature regulator based on a PID controller.

The temperature of the copper bar is computed using a HiFlow3 solver, and the 'LimPID' controller using a model from the Modelica standard library. The components are coupled by importing the HiFlow3 solver as a Modelica model using FMI, and then creating a third Modelica model, which connects these two components as well as some additional components.

1.1 Computing the temperature distribution

The evolution of the temperature distribution is modeled by the unsteady heat equation. In this subsection, we give the mathematical problem formulation, discuss the numerical treatment of the heat equation, and describe the discretization we used in the computations.

Heat Equation

We consider the copper to occupy a domain

$$\Omega := (0,0.045) \times (0,0.03) \times (0,0.03) \subset \mathbb{R}^3,$$

where the boundary of Ω is denoted by $\partial\Omega$. The heat problem formulation for our simulation scenario is the following:

Find a function $u : \Omega \times (0,T) \to \mathbb{R}$ as the solution of

$$\partial_t u - \alpha \Delta u = 0 \text{ in } \Omega \times (0,T), \tag{1a}$$

$$u(0) = u_0 \text{ in } \Omega \tag{1b}$$

$$u = g \text{ on } \Gamma_{src} \times (0,T), \tag{1c}$$

$$u = u_{top} \text{ on } \Gamma_{top} \times (0,T), \tag{1d}$$

$$u = u_{left} \text{ on } \Gamma_{left} \times (0,T), \tag{1e}$$

$$\frac{\partial u}{\partial n} = 0 \text{ on } \Gamma_{iso} \times (0,T) \tag{1f}$$

The unsteady heat equation (1a) is a parabolic PDE. Its solution, the unknown function u, describes the evolution of the temperature in the copper bar Ω during the time interval $(0,T)$. Here, $\alpha = 1.11 \times 10^{-4} \left[\frac{m^2}{s}\right]$ denotes the thermal diffusivity of the copper. u_0 in equation (1b) is the initial state at time $t = 0$. The sensor is placed at the point $x_0 := (0.045, 0.015, 0.015)$, where the temperature $u(x_0, t)$ is taken as measurement value for time $t \in (0,T)$. The heat source is modeled by the Dirichlet boundary condition (1c). The controlled temperature $g(t)$ is prescribed on the source part of the boundary

$$\Gamma_{src} := [0,0.045] \times [0,0.03] \times \{0\} \subset \partial\Omega$$

The environmental influence is modeled by the Dirichlet boundary conditions (1d) and (1e). At the top boundary part

$$\Gamma_{top} := [0,0.045] \times [0,0.03] \times \{0.03\} \subset \partial\Omega$$

and the left boundary part

$$\Gamma_{left} := \{0\} \times [0.01,0.02] \times [0.01,0.02] \subset \partial\Omega$$

the temperature is given by the functions $u_{top}(t)$ and $u_{left}(t)$ for $t \in (0,T)$, respectively. The rest of the boundary

$$\Gamma_{iso} := \partial\Omega \setminus \left(\Gamma_{\text{top}} \cup \Gamma_{\text{left}} \cup \Gamma_{\text{src}}\right)$$

is isolated through the homogeneous Neumann boundary condition (1f).

Variational Formulation

A well-established method for numerically solving PDEs like the heat equation is the finite element method. Here we briefly describe the numerical treatment of problem (1). The methods of this section are taken from [2] and [5].

The finite element discretization is based on a variational formulation, which can be derived as follows. We denote by $C^k(X;Y)$ the set of all k times continuously differentiable functions from X to Y, and by $C_0^\infty(X;Y)$ the set of all smooth functions with compact support. In the common case $X = \Omega, Y = \mathbb{R}$, we just write $C^k(\Omega)$. Assuming that there is a classical solution

$$u \in C^1(0,T; C^2(\Omega) \cap C(\bar{\Omega}))$$

of problem (1), equation (1a) is multiplied by a test function $v \in C_0^\infty(\Omega)$ and integrated over Ω:

$$\int_\Omega (\partial_t u) v \, dx - \alpha \int_\Omega (\Delta u) v \, dx = 0 \tag{2}$$

Green's first identity [9] is applied to the second term of (2), giving

$$\int_\Omega (\Delta u) v \, dx = \int_{\partial\Omega} \frac{\partial u}{\partial n} v \, ds - \int_\Omega \nabla u \cdot \nabla v \, dx$$

$$= -\int_\Omega \nabla u \cdot \nabla v \, dx,$$

where n is the outer unit normal on $\partial\Omega$. The boundary integral vanishes since v is zero on $\partial\Omega$. This leads to

$$\int_\Omega (\partial_t u) v \, dx + \alpha \int_\Omega \nabla u \cdot \nabla v \, dx = 0 \tag{3}$$

For equation (3) to be well-defined, weaker regularity properties of u and v than in the classical context are sufficient. The problem can be formulated in terms of the Lebesgue space $L^2(\Omega)$ of square-integrable functions and the Sobolev space $H^1(\Omega)$ of functions in $L^2(\Omega)$ with square-integrable weak derivatives. We define the solution space

$$V := \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_{left} \cup \Gamma_{top} \cup \Gamma_{src}\}$$

and the bilinear form

$$a : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R},$$

$$a(u,v) := \alpha \int_\Omega \nabla u \cdot \nabla v$$

Note that a is symmetric, continuous and V-elliptic. We denote by $(u,v)_{L^2} = \int_\Omega uv\,dx$ the standard inner product in $L^2(\Omega)$. Now we can state a variational formulation of problem (1):
Find $u \in \bar{u} + C^1(0,T;V)$ as the solution of

$$(\partial_t u, v)_{L^2} + a(u,v) = 0 \qquad \forall v \in V \tag{4a}$$

$$u(0) = u_0 \tag{4b}$$

where $\bar{u} \in C^1(0,T;H^1(\Omega))$ is a given function fulfilling the Dirichlet boundary conditions

$$\bar{u} = g \quad \text{on } \Gamma_{left} \times (0,T),$$
$$\bar{u} = u_{top} \quad \text{on } \Gamma_{top} \times (0,T),$$
$$\bar{u} = u_{src} \quad \text{on } \Gamma_{src} \times (0,T),$$

This variational formulation admits a unique solution u, which is called a weak solution of the heat equation.

Finite element discretization in space

Let $T_h := \{K_1, \dots, K_N\}$ be a triangulation of Ω with N tetrahedral cells $K_i (i = 1, \dots, N)$. We define the finite element space of piecewise linear functions

$$V_h := \{v \in V : v|_K \text{is linear } (K \in T_h)\}. \tag{5}$$

V_h has the finite dimension $n := \dim(V_h)$. We give the problem formulation for a conforming finite element approximation of (4):
Find $u_h \in \bar{u}_h + C^1(0,T;V_h)$ as the solution of

$$(\partial_t u_h, v_h)_{L^2} + a(u_h, v_h) = 0 \qquad \forall v_h \in V_h, \tag{6a}$$

$$(u_h(0), v_h)_{L^2} - (u_0, v_h)_{L^2} = 0 \qquad \forall v_h \in V_h. \tag{6b}$$

Let $\{\varphi_1, \dots, \varphi_n\}$ be a basis of V_h. We define the ansatz

$$u_h(x,t) := \sum_{i=1}^{n} w_i(t)\varphi_i(x)$$

and insert it into (6a), yielding

$$\sum_{i=1}^{n} \dot{w}\,(\varphi_i, \varphi_j)_{L^2} + \sum_{i=1}^{n} w_i a(\varphi_i, \varphi_j) = 0 \quad (j = 1, \dots, n).$$

This can be written as

$$M\dot{w} + Aw = 0,$$

where

$$M := \left((\varphi_j, \varphi_i)_{L^2}\right)_{i,j=1,\dots,n} \in \mathbb{R}^{n\times n}$$

is the mass matrix and

$$A := \left(a(\varphi_i, \varphi_j)\right)_{i,j=1,\dots,n} \in \mathbb{R}^{n\times n}$$

is the stiffness matrix of the problem, and

$$w: (0,T) \to \mathbb{R}^n$$

is the vector of the time-dependent coefficients. From (6b), an initial condition for w is derived as

$$(Mw_0)_i = (u_0, \varphi_i)_{L^2} \quad (i = 1, \dots, n)$$
$$\Leftrightarrow w_0 = M^{-1}b,$$

where $b_i = (u_0, \varphi_i)_{L^2}$ $(i_1, \dots, n)$. Thus, the finite element discretization in space leads to the initial value problem

$$M\dot{w} + Aw = 0, \tag{7a}$$
$$w(0) = w_0 \tag{7b}$$

for the coefficient vector w.

Implicit Euler discretization in time

As will be discussed in Section 2.6, limitations in the used technology restrict us to use a relatively simple ODE solver for the time discretization. For the heat equation the implicit Euler scheme is suitable, due to its good stability properties [8]. Let $\{0 = t_0 < t_1 < \dots < t_m = T\}$ be a partition of the time interval with step sizes $\partial t_k = t_{k+1}$ $(k = 0, \dots, m-1)$. The implicit Euler time stepping method for problem (7) is defined as

$$[M + \partial t_k A]w(t_{k+1}) = Mw(t_k) \tag{8}$$

for $k = 0, \dots, m-1$. This method is convergent and has first-order accuracy in terms of the step size ∂t_k.

1.2 PID controller

A Proportional-Integral-Derivative controller (PID controller) is a form of loop feedback controller, which is widely used to control industrial processes. The controller takes as input a reference value u_{ref}, which represents the desired temperature, and the measured value $u(x_0,t)$. It uses the error $e(t) := u_{ref} - u(x_0,t)$ to compute the output signal $g(t)$. As the name PID suggests, there is a proportional part that accounts for present errors, an integral part that accounts for the accumulation of past errors, and a derivative part that predicts future errors:

$$g(t) = w_P e(t) + w_I \int_0^t e(\tau)\,d\tau + w_D \frac{d}{dt} e(t)$$

Here, w_P, w_I and w_D are weight parameters. By tuning these parameters the performance of the controller can be adapted to a specific process. A PID controller is widely regarded as the best controller when information of the underlying process is lacking, but the use of a PID controller does not guarantee optimal control.

The tuning of the parameters can be done manually or by using a formal method such as Ziegler-Nichols or Cohen-Coon. There are also software tools available. Sometimes one or several of the parameters might have to be set to zero. For instance, derivative action is sensitive to measurement noise thus this part of the controller might have to be omitted in some situations, resulting in a PI controller. PID controllers are linear and can therefore have problems controlling non-linear systems, such as Heating, Ventilation and Air Conditioning (HVAC) systems. [1]

2 Coupled Implementation

In this section, we first briefly introduce the technologies used in the present work. We then describe the coupled simulation setup, as well as its two main constituent components in more detail.

2.1 The Modelica language

Modelica is a language for equation-based object-oriented mathematical modeling which is being developed and standardized through an international effort in the Modelica Association [15]. The equation parts of Modelica requires a lot of the compiler developer: knowledge in compiler construction, symbolic manipulation of equations and associated mathematics, as well as knowledge of numerical mathematics. The simulation run-time system is also an important part and can be complex; various solver techniques for solving the differential equations can be applied. Modelica allows high-level concepts such as object-oriented modeling and component composition. Multi-domain modeling is also possible in Modelica with the possibility of combining model components from a variety of domains within the same application. There exist several mature and well-maintained Modelica development environments, such as Dymola, OpenModelica, MapleSim,Wolfram SystemModeler, Simulation X, and JModelica.org.

2.2 The OpenModelica environment

OpenModelica is a modeling and simulation environment, which is developed and supported by an international consortium, the Open Source Modelica Consortium (OSMC) [4]. This effort includes an open-source implementation of a Modelica compiler, a simulator and a development environment for research, education and industrial purposes.

2.3 The HiFlow3 Finite Element Library

HiFlow3 [22] is a multi-purpose finite element software providing powerful tools for efficient and accurate solution of a wide range of problems modeled by partial differential equations (PDEs). Based on object-oriented concepts and the full capabilities of C++ the Hi- Flow3 project follows a modular and generic approach for building efficient parallel numerical solvers. It provides highly capable modules dealing with mesh setup, finite element spaces, degrees of freedom, linear algebra routines, numerical solvers, and output data for visualization. Parallelism - as the basis for high performance simulations on modern computing systems - is introduced at two levels: coarse-grained parallelism by means of distributed grids and distributed data structures, and fine-grained parallelism by means of platform-optimized linear algebra back-ends.

2.4 The Functional Mock-Up Interface (FMI)

The Functional Mock-Up Interface (FMI) [6, 16] is a tool-independent standard to support both model exchange and co-simulation of dynamic models which can be developed with any language or tool. Models that implement the FMI can be exported as a Functional Mock-Up Unit (FMU). Such a FMU consists mainly of two parts: (1) XML file(s) describing the interface and (2) the model functionality in compiled binary or C code form. Other tools or models, which also implement the FMI, can import Functional Mock- Up Units. The initial FMI development was done in the European ITEA2 MODELISAR project [17].

2.5 Simulation overview

Figure 2 gives an overview of the simulation setup. To create the PDE component, we reused an existing HiFlow3 application, which solves the boundary value problem for the heat equation (1). In order to import this code into Modelica, we converted it into a Dynamic Shared Object (DSO), which implements the FMI functions and interface descriptions necessary to build a Functional Mock-Up Unit. We then loaded this FMU via FMI into our Modelica model. The details of the PDE component are described in Section 2.6.

For the PID controller, we used the LimPID component from the Modelica standard library. This was connected to the PDE component in a new Modelica model, which is described in Section 2.7.

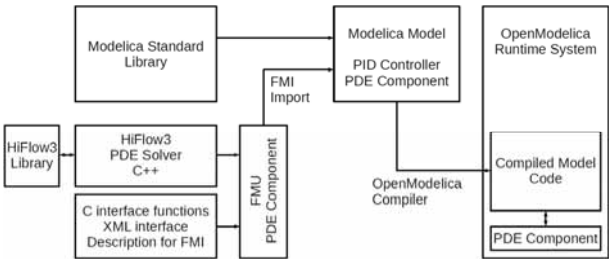


Figure 2: Overview of the creation and coupling of the simulation components.

This model was then compiled with the OpenModelica compiler and executed using the OpenModelica run-time system. By choosing the *Euler* solver, the runtime system provides the time-stepping algorithm according to the implicit Euler scheme, and additionally solves the equations for the PID controller component in each time step. Figure 3 illustrates on a time axis the calls made to the compiled model code, which in turn makes calls to the PDE component.

2.6 HiFlow3-based PDE component

The main sub-part of the PDE component is the `HeatSolver` class, which is a slightly modified version of an existing HiFlow3 application. This class uses data structures and routines from the HiFlow3 library to solve the heat problem (1) numerically. It uses a finite element discretization in space and an implicit Euler scheme in time as described before.

Furthermore, this class provides functions for specifying the current time, the controlled temperature of the heat source, the top and bottom temperatures, and for retrieving the temperature at the measuring point. The top level routine of the `HeatSolver` class is its `run()` function, see Listing 1. This function computes the solution of the heat equation.

We prepared the triangulation Th in a preprocessing step and stored it in a mesh file. When the `run()` function is called for the first time, it reads the mesh file and possibly refines the mesh. It also creates the data structures representing the finite element space V_h from (5), the linear algebra objects representing the system matrix, the right-hand-side vector and the solution vector. Then, the `run()` function assembles the system matrix $M + \delta t_k A$ and the right-hand-side vector $Mw(t_k)$ according to (8). It computes the solution vector $w(t_{k+1})$ for the new time step $t = t_{k+1}$ using the conjugate gradient method [18]. The solution is stored in *VTK* format [21] for visualization.

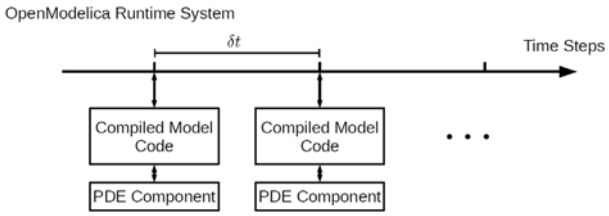


Figure 3: Interaction between the OpenModelica run-time system and the coupled model with the implicit Euler solver.

```
HeatSolver_run() {
  // if this is the first call
  if (first_call) {
  // read mesh file and eventually refine it
   build_initial_mesh();
  //initialize the finite element space and
  //the linear algebra
  //data structures
   prepare_system();
   first_call = false;
  }
  //compute the system matrix and
  //the right-hand-side vector
  assemble_system();
  // solve the linear system
  solve_system();
  // visualize the solution
  visualize();
  //keep solution and time in memory
  CopyFrom(prev_solution,old_solution);
}
```

Listing 1: Run function of the HeatSolver class.

It is important to note that the solution vector and the current time are kept in memory inside the PDE component, since this data is required for computing the solution at the next time step. Although it has been planned for a future version, at present the FMI standard does not directly support arrays, which prevents passing the solution vector back and forth between the PDE component and the Modelica environment as a parameter [16]. Although this use of mutable internal state in the PDE component might be preferable from a performance point of view, it has the drawback of making the function calls referentially opaque: two calls with the same parameters can yield different results, depending on the current internal state. This imposes a strong restriction on the solver used, which must assure that the sequence of time values for which the function `run()` is called is non-decreasing.

For this reason, we only tested the method with the simple implicit Euler solver, and verified that the calls were indeed performed in this way. For more complicated solvers, such as DASSL, this requirement is no longer satisfied.

The entry point of the PDE component is the `PDE_component()` function, see Listing 2. This function is called within the Modelica simulation loop. When it is called for the first time, it creates a `HeatSolver` object. It sets the input values for the heat source, the temperatures at the top and bottom boundary, and the current time. Then the `run()` function of the `HeatSolver` is called to compute the solution of the heat equation. Finally, the run counter is incremented and the measurement value is returned.

2.7 Modelica model

Our Modelica model is shown schematically in Figure 4. It contains the PDE component, the PID controller, and four source components. Two of the sources represent the environmental influences, which are given by sinusoidal functions.

They are connected to the PDE component to give the top and left boundary temperatures u_{top} and u_{left} in Equations (1d) and (1e), respectively. One source is connected to the PID controller and gives the reference value u_{ref} for the desired temperature.

```
PDE_component(
double in_Controlled_Temp,
double in_Top_Bdy_Temp,
double in_Bottom_Bdy_Temp,
double in_Time)
{
  // create HeatSolver object if this is the
  //first call
  if (run_counter == 0)
   heat_solver = new HeatSolver();
  // set input values
  heat_solver->set_time(in_Time);
  heat_solver->set_g(in_Controlled_Temp);
  heat_solver->set_top_temp(in_Top_Bdy_Temp);
  heat_solver-
        >set_bottom_temp(in_Bottom_Bdy_Temp);
  // run the HeatSolver
  heat_solver->run();
  // increment the run counter
   run_counter++;
  // return the measurement value
return heat_solver->get_u(); }
```

Listing 2: Main simulation routine of the PDE component.

The fourth source is connected to the dummy state variable of the PDE component. The dummy state variable and its derivative are in the model due to the fact that the OpenModelica implementation of FMI 1.0 import does not allow for an empty state variable vector. There is however nothing in the FMI 1.0 model exchange specification that disallows this. Additionally, the measurement value of the PDE component is connected to the input of the PID controller, and the output signal of the PID controller is connected to the heat source input of the PDE component.

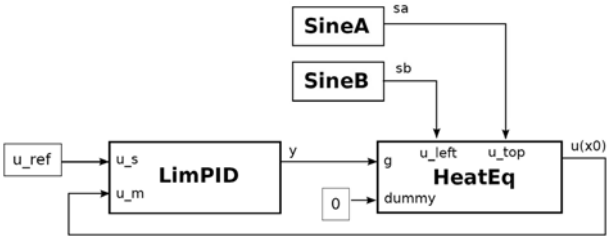


Figure 4: Schematic view of the coupled Modelica model used in the simulation.

The internal constant parameters of the components are summarized in Table 1.

Component Parameter	Value
LimPID	
proportional gain w_p	0.05
integral gain w_I	0.2
derivative gain w_D	0.0
HeatEquationFMU	
thermal diffusivity α	$1.11 \cdot 10^{-4} m^2 s^{-1}$
SineA	
amplitude	0.5°C
vertical offset	3.5°C
start time	$150.0s$
frequency	$0.001s^{-1}$
SineB	
amplitude	6.0°C
vertical offset	3.0°C
start time	$350.0s$
frequency	$0.002s^{-1}$

Table 1: Internal parameters of the components in the simulation model.

3 Results

We carried out a numerical experiment with the following setting. We took a simulation time of $T = 1500$ seconds and a time step of $\partial_t = 1$. We set the initial temperature $u_0 = 0$ everywhere in the computational domain, and we specified the desired temperature as $u_{ref} = 3$. On the upper part of the boundary Γ_{top} and on the left part of the boundary Γ_{left} we modeled environmental influences by the functions

$$u_{top}(t) = \begin{cases} 3.5 & \text{if } t < 150 \\ 3.5 + 0.5 \sin \dfrac{(t-150)\pi}{500} & \text{if } t \geq 150 \end{cases}$$

and

$$u_{left}(t) = \begin{cases} 3 & \text{if } t < 350 \\ 3 + 6 \sin \dfrac{(t-350)\pi}{250} & \text{if } t \geq 350 \end{cases}$$

which are shown in Figure 5.

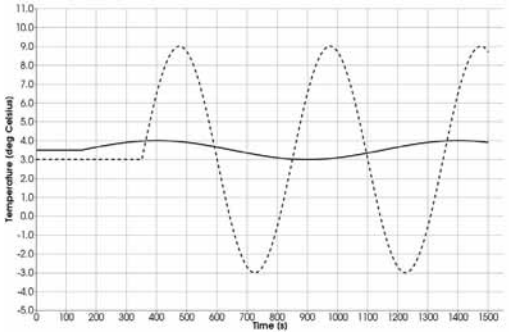


Figure 5: Environmental temperature prescribed on the boundary parts Γ_{left}left and Γ_{top}. Dashed: $u_{left}(t)$, solid: $u_{top}(t)$.

For comparison, we first performed a simulation run with a constant, uncontrolled heat source $g \equiv 3$ on the lower boundary Γ_{src}. Figure 6 shows that the temperature at the point of measurement deviates from the desired temperature $u_{ref} = 3$ due to the environmental influences.

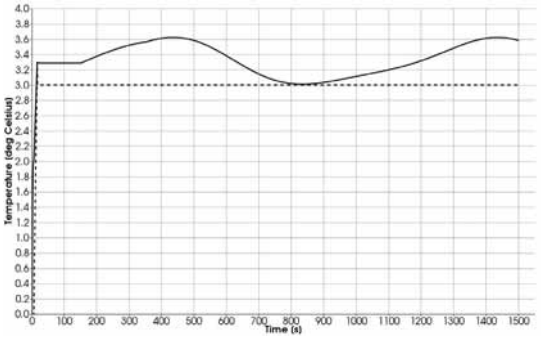


Figure 6: Simulation run with constant heat source $g \equiv 3$. The temperature $u(x_0, t)$ at the point of measurement deviates from the desired value. Dashed: g, solid: $u(x_0, t)$.

The results of our simulation run with a controlled heat source $g = g(t)$ are shown in Figure 7. At the beginning, we fixed the heat source at $g = 2.5$ to let the temperature distribution in the copper bar evolve from the initial state to an equilibrium, at which the measured temperature is slightly higher than desired. The PID controller was switched on at $t = 50$ to take control over the heat source. The curves show that the controller first cools the bottom to bring the temperature at the point of measurement down to the target value. During the rest of the simulation, the PID controller reacts to the environmental influences and adjusts the heat source dynamically over time, so that the temperature accurately follows the desired state. Figures 8-10 illustrate the temperature distribution in the copper bar.

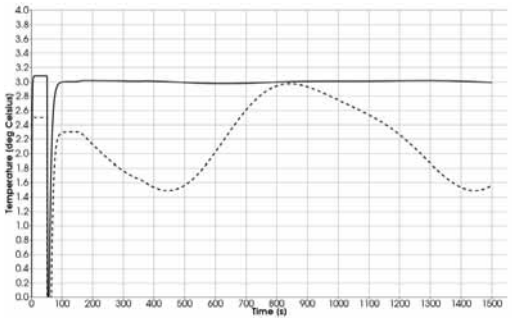


Figure 7: Simulation run with controlled heat source $g = g(t)$. The temperature $u(x_0, t)$ at the point of measurement accurately follows the desired value $u_{ref} = 3$. Dashed: $g(t)$, solid: $u(x_0, t)$.

4 Discussion

The numerical results for the example presented in the previous section show that our method of integrating the PDE solver from HiFlow3 into a Modelica simulation functions correctly. The realization of this particular scenario serves as an illustration of how one can integrate other, more complicated, PDE models into the complex dynamical simulations for which Modelica is especially suited.

The coding and maintenance effort for importing an existing PDE model implemented in HiFlow3 with the method presented here is minimal: in essence only a set of wrapper functions dealing with input and ouput of parameters and state variables is all that is required. The fact that HiFlow3 is free and open source software simplifies the process greatly, since it makes it possible to adapt and recompile the code. This is significant, since the FMI model import requires the component to be available either as C source code or as a dynamic shared object, which is loaded at run-time.

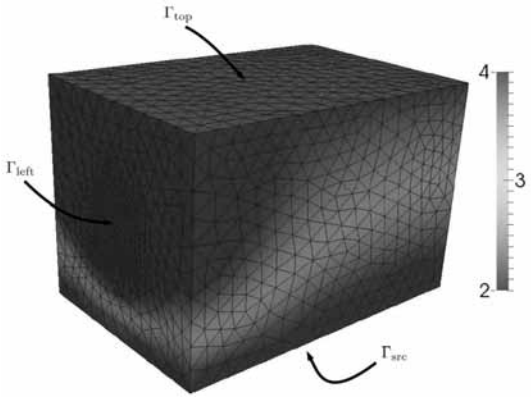


Figure 8: Computational domain of the copper bar with triangulation. The colors indicate the temperature distribution on the surface at time $t = 440$.

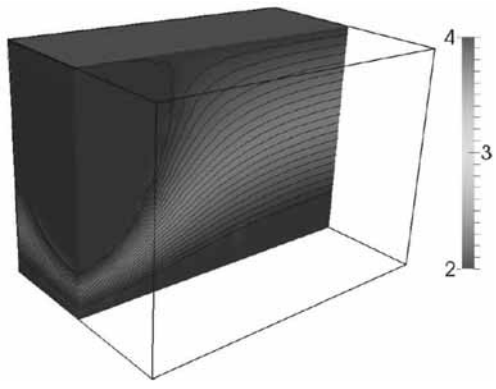


Figure 9: Sectional view with isothermal lines at time $t = 440$.

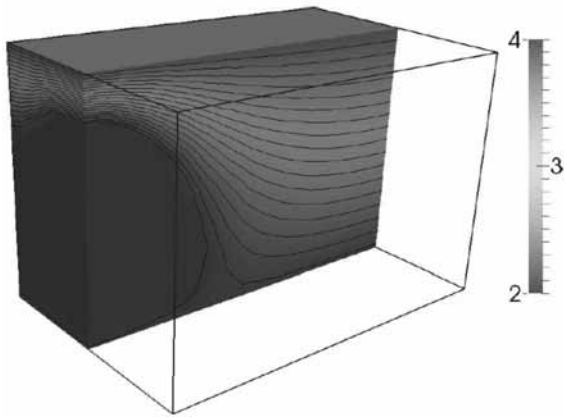


Figure 10: Sectional view with isothermal lines at time $t = 1250$.

Compared to the efforts aiming at extending the Modelica language with support for PDEs, we are working at a different level of abstraction, namely that of software components. The advantage of this is the ability to make use of the large wealth of existing implementations of solvers for various models, in the present case the multipurpose HiFlow3 library. Extending the Modelica language would also make it considerably more complex, since problems for PDEs generally require descriptions of the geometry and 7 the conditions applied to the different parts of the boundary. Furthermore, using this information to automatically generate a discretization and a solution algorithm would require a sophisticated classification of the problem, since different types of PDEs often require different numerical methods. A drawback of working at the software level is that the mathematical description of the problem is not directly visible, as it would be if it was part of the language.

In contrast to the use of 'co-simulation', in this work we have chosen to import the PDE component into the OpenModelica environment, and to make use of one of its solvers. The main benefit is again simplicity: very few changes were required to the PDE component itself, and it was possible to maximize the reuse of existing software. On the other hand, co-simulation, where each submodel has its own independent solver, which is executed independently of the others, also has its advantages. In particular, specialized, highly efficient solution algorithms can be applied to each part of the model, and it is possible to execute the various components in parallel. We are considering to extend the present work to make it usable in a co-simulation setting. Furthermore, we want to investigate the parallelization of the simulation both within and between components.

5 Conclusion

In this paper we have investigated a method of incorporating PDEs in the context of a Modelica model, by using FMI to import a PDE solver from the finite element library HiFlow3. Numerical results obtained using a simple coupled model controlling the heat equation using a PID controller demonstrate that this method works in practice. The main advantages of this type of coupling include its simplicity and the possibility to reuse existing efficient and already validated software. This approach allows to make use of more complex PDE models including high-performance, parallel computations. It has the potential of greatly simplifying the development of large-scale coupled simulations. In this case, however, an extension of the method presented here to support co-simulation might be necessary.

Acknowledgments

Work in this publication was funded by the ITEA2 European Union MODRIO Project, by SSF in the EDOp project, and by the Swedish National Graduate School of Computer Science (CUGS).

This contribution is a post-conference publication from SIMS 2013 Conference (54th SIMS Conference, Bergen University College, Norway, October 16-18, 2013). The contribution is a modified publication from the paper published in the Proceedings of SIMS 2013, to be found at http://www.scansims.org/sims2013/SIMS2013.pdf.

References

[1] Åström KJ, Hägglund T. *PID Controllers: Theory, Design, and Tuning.* Durham: The Instrumentation, Systems, and Automation Society, 1995. 343 p.

[2] Braess, D. *Finite Elemente*. Heidelberg: Springer, 2007. 376 p.

[3] Dshabarow, F. *Support for Dymola in the Modeling and Simulation of Physical Systems with Distributed Parameters* [master thesis]. [Department of Computer Science, Institute of Computational Science, (CH)] ETH Zürich; 2007.

[4] *The Open-Source OpenModelica Development Environment*. http://www.openmodelica.org.

[5] Ern A, Guermond JL. *Theory and Practice of Finite Elements*. New York: Springer; 2010. 526 p.

[6] *The Functional Mockup Interface* (FMI). https://www.fmi-standard.org/.

[7] Fritzson, P. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Press, 2004. 944 p.

[8] Hairer E, Norsett SP, and Wanner G. *Solving Ordinary Differential Equations*. Springer, 2008. 528 p.

[9] Heuser, H. *Lehrbuch der Analysis 2*. Stuttgart: Teubner, 2002. 737 p.

[10] Khan, S. *Discretizing pdes for maplesim*. adept scientific plc. 2012.

[11] Li Z, Zheng L, and Zhang H. Solving pde models in modelica.In *ISISE*. Proceedings of a meeting; 2008, Dec; Shanghai. IEEE Computer Society; 53–57.

[12] Ljubijankic M, Nytsch-Geusen C. 3D/1D Co-Simulation von Raumlufströmungen und einer Luftheizung am Beispiel eines thermischen Modellhauses. *Fourth German-Austrian IBPSA Conference*, *BauSIM* 2012;167-175.

[13] Ljubijankic M, Nytsch-Geusen, Rädler J, Löffler M. Numerical coupling of Modelica and CFD for building energy supply systems. In: Clauß C, editor. *8th International Modelica Conference*; 2011 Mar; Dresden. Linköping: The Modelica Association and Linköping University Electronic Press. 286-294. doi: 10.3384/ecp11063

[14] Maple and MapleSim by MapleSoft. http://www.maplesoft.com/products.

[15] Modelica and the Modelica Association. http://www.modelica.org.

[16] Modelica Association. *Functional Mock-up Interface for Model Exchange and Co-Simulation*, v. 2.0 beta 4 edition, 2012. https://www.fmi-standard.org/downloads [Accessed 2013-08-06].

[17] *The ITEA2 MODELISAR Project*. http://www.itea2.org/project/index/view/?project=217.

[18] Saad, Y. *Iterative Methods for Sparse Linear Systems*. 2000.

[19] Saldamli L. *PDEModelica - A High-Level Language for Modeling with Partial Differential Equations* [dissertation]. [Department of Computer and Information Science]. Linköping University, 2006. http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-7281.

[20] Saldamli L, Bachmann B, Wiesmann HJ, Fritzson P. A framework for describing and solving PDE models in modelica. In: Schmitz G, editor. Proceedings of the *4th International Modelica Conference*; 2005 Mar; Hamburg, Germany. Hamburg: The Modelica Association and Hamburg University of Technology. 113-122.

[21] Schroeder W, Martin K, and Lorensen B. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. 4th ed. Kiteware, Inc., 2006. 528 p.

[22] *The HiFlow3 Multi-Purpose Finite Element Software*. http://www.hiflow3.org/.

Evaluation of Compartment Models for Simulation of Infliximab Depletion

Andreas Bauer[1,*], Christoph Urach[2], Felix Breitenecker[1]

[1]Department of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *andreas.e101.bauer@tuwien.ac.at*

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

Simulation Notes Europe SNE 24(1), 2014, 21 - 26
DOI: 10.11128/sne.24.tn.10225
Received: Dec. 10, 2013 (Selected ERK 2013 Postconf. Publ.);
Accepted: February 15, 2014;

Abstract. Infliximab is an antibody that is approved for treatment of the inflammatory bowel diseases Crohn's disease and ulcerative colitis. It is important to predict the course of the depletion of Infliximab in the body to time the regular infusions that patients get. The base model is a two-compartment-model and three parameter identification approaches are compared: identification of each infusion period separately, identification of each patient separately and identification of all measurement points at once. The best results provides the approach to identify the parameters for each infusion period separately. Wanting to improve the quality of the results, an extended model with continuous parameter antibodies to Infliximab (ATI) is considered, but the mean error is higher than for the base model. We conclude that the presence of antibodies to Infliximab in the body carries more weight than the actual number of antibodies. Finally, a model with an additional parameter, number of previous infusions with Infliximab (PRIORIFX), is created. This model could not improve the results of the base model. This indicates that the number of previous infusions with Infliximab is not significant for the model.

Introduction

Infliximab is an antibody that is approved mainly for treatment of the inflammatory bowel diseases Crohn's disease and ulcerative colitis.

The patients receive infusions with Infliximab on a regular basis, because the concentration of Infliximab within the body must not fall below a critical level. It is important to predict the point of time, when the concentration reaches that critical level to plan the date of the next infusion more accurately. This can help to reduce the number of visits from the patients to the hospital or the doctor and to reduce the number of infusions since Infliximab is very expensive.

The goal of this paper is to model the time dependent course of the concentration of Infliximab in the body. The base model is taken from [1]. In order to improve the prediction quality of the model various approaches for parameter identification and various extensions of input parameters are considered.

1 Structure of the Models

The presented models are compartment-models. A compartment-model consists of separated compartments and flows between them. The interaction with the environment is realised through sinks and sources.

In this particular case, we consider two-compartment-models. The first compartment is the central compartment V_1 and the second compartment is the peripheral compartment V_2. The central compartment represents the well perfused, central part of the body and the peripheral compartment represents the less perfused, peripheral part of the body. There are flows from V_1 to V_2 and back. From a source there is an inflow into V_1 which represents the injection of Infliximab into the body and an outflow from V_1 into a sink which represents the clearance (Figure 1).

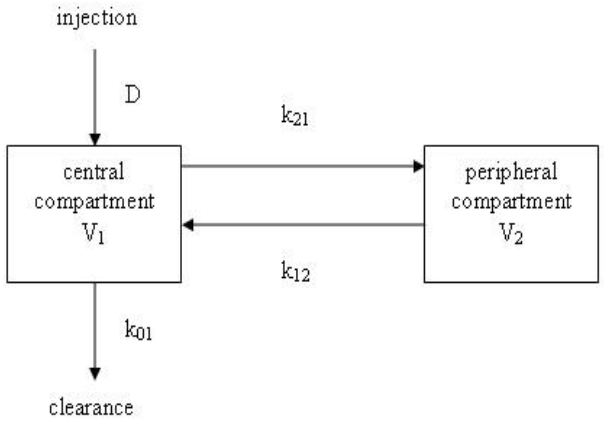


Figure 1: Illustration of the structure of the model.

1.1 Base model (M1)

The structure of the base model is taken from [1]. An overview of the five parameters of the base model and their attributes is given in Table 1.

Name	Abbreviation	Unit	Values
Dose	DOS	mg	300-700
Sex	SEX	none	0 = male, 1 = female
Antibodies to Infliximab status	ATI	none	0 = tested negative on antibodies, 1 = tested positive on antibodies
Weight	WGT	kg	40-130
Albumin Level	ALB	g/dL	3-5

Table 1: Overview of the parameters of the base model

The size of the compartments and the flows depend on the five parameters. The size of the first compartment V_1 is depending on the weight and the sex as seen in equation (1). The size of the second compartment V_2 and the intercompartmental flow Q are constant. The clearance depends on the parameters ALB, ATI and SEX. (2) There are eleven constants (v_{ij}, CL_k) whose values are identified from the data. These constants describe the functional dependencies between the parameters and V_1 and CL:

$$V_1 = v_{11} \cdot \left(\frac{WGT}{v_{12}}\right)^{v_{13}} \cdot (v_{14})^{SEX} \quad (1)$$

$$CL = CL_1 \cdot \left(\frac{ALB}{CL_2}\right)^{CL_3} \cdot (CL_4)^{ATI} \cdot (CL_5)^{SEX} \quad (2)$$

The actual flows are calculated by dividing the quantities Q and CL from above by the size of the compartments [2]:

$$k_{01} = \frac{CL}{V_1} \quad (3)$$

$$k_{12} = \frac{Q}{V_1} \quad (4)$$

$$k_{21} = \frac{Q}{V_2} \quad (5)$$

The compartment model is described by two differential equations. The variables x_1 and x_2 describe the amount of Infliximab in compartment V_1 and V_2. x_1 decreases by the outflows k_{01} and k_{21} from V_1 and increases by k_{12} which goes into V_1 (6). In compartment V_2, there is a flow to V_1 and an inflow from V_1 [3].

A run always simulates one infusion period. The initial value for x_1 is the dose, which is injected all at once, if it is the very first infusion of Infliximab for the particular patient. Otherwise the amount of Infliximab which is left in the body at that time has to be added to the dose to get the initial value. The initial value for x_2 is zero. To get the concentration of Infliximab, the variables x_1 and x_2 have to be divided by the size of the corresponding compartment.

$$\dot{x}_1(t) = -(k_{01} + k_{21}) \cdot x_1 + k_{12} \cdot x_2 \quad (6)$$

$$\dot{x}_2(t) = k_{21} \cdot x_1 - k_{12} \cdot x_2 \quad (7)$$

1.2 Model with continuous ATI (M2)

The model M2 is similar to the base model M1. The only difference is that the parameter ATI is continuous in contrast to the model M1 where it is discrete with possible values zero and one. In the base model, the parameter ATI is simply set to one, if the value in the data is greater than zero.

The reason to incorporate the continuous ATI is that within the data there are already continuous values given for patients with antibodies and to find out the effects of different values of antibodies on the results. The unit of the continuous ATI is Unit per millilitre (U/mL). Four different techniques are used to transform the values from the data which range between 0.5 and 20 onto a certain interval:

1. Normalising the values onto the interval [0,1]
2. Linear transformation onto the interval [0.5,1]
3. Logarithmic transformation onto the interval [0,1]
4. Exponential transformation onto the interval [0,1]

1.3 Model with additional parameter (M3)

The model M3 is also similar to the base model M1. The difference is that the model M3 has an additional parameter PRIORIFX. The value of the parameter indicates, how many previous injections of Infliximab a patient has had. The size of the compartments will not change by additional injections. Thus, the formulas of the size of the compartments are still equal to those of the base model M1.

Two different approaches will be tested if a big number of injections causes an effect of habituation: the first (M3a) is that the new parameter only affects the clearance CL (10) and the second (M3b) is that it affects both the clearance CL and the intercompartmental flow Q (10, 11). In both cases, the factor PRIORIFX to the power of a constant is added to the existing formula.

$$CL = CL_1 \cdot \left(\frac{ALB}{CL_2}\right)^{CL_3} \cdot (CL_4)^{ATI} \cdot (CL_5)^{SEX} \cdot \cdot PRIORIFX^{CL_6} \quad (8)$$

$$Q = Q_1 \cdot PRIORIFX^{Q_2} \quad (9)$$

2 Parametrisation

2.1 Data

The data origins from the Vienna General Hospital and was generously placed at our disposal for this project. 662 records have been collected, but only 96 of them can be used, because only these contain all the required parameters needed for the model. The categories that have been collected besides from those that are used in the models are age, smoking habits, height of the patient and some other categories. However, our models stick with the parameters found to be significant in [4].

The diagnoses of the patients were also recorded. 81 patients were diagnosed with Morbus Crohn (MC) and the other 15 with Colitis Ulcerosa (CU). Since it is unsure, if a single model can be used for patients with both diagnoses and the size of the sample of CU patients is too small, only MC patients are considered.

2.2 Identification of the constants

The constants in the equations (1) and (4) respectively in (10) and (11) have to be identified to make the models fit the given data. For this problem, the method of least squares is used which minimises the squares of the residuals.

The residuals are the differences of the amount of the concentration from the data and the value of the function x_1 of the model at given time. The implementation is done in MATLAB with the pre-implemented function lsqnonlin. The algorithm which is used is the Levenberg-Marquardt-algorithm, an extended Newton-algorithm. Three different approaches to identify the constants are carried out.

The patient history is the complete course of concentration of Infliximab. It can be divided into several infusion periods. Within every infusion period the concentration is measured with a few samples. In Figure 2, a simulated patient history with discrete measurement points is shown.

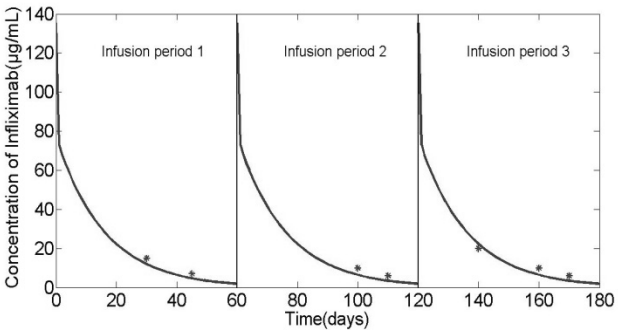


Figure 2: Simulated history of a patient split in its infusion periods and measurement points.

Identification of infusion periods (P1).

The first approach is to identify the constants for every single period between two infusions separately. That means, all data points between the date of an infusion and the following infusion are taken and the constants are identified that the function x_1, which describes the amount of Infliximab in compartment V_1, fits these data points best. This is done for all periods. This approach produces many different sets of identified constants. The mean values for each constant are calculated and used in the general model.

In most of the cases, there is only one data point in an infusion period, so the constants are identified to fit that single point. At most there are three points, thus in almost every case a near perfect fit can be achieved, because in M1 and M2, there are 11 and, in M3, up to 14 constants to identify. In this scenario, problems with overfitting can occur, because the number of parameters is much greater than the number of data points.

Identification of records of patients (P2).
In this approach, the constants are identified separately for each patient. That means, all data points of a patient are taken and the constants are identified. Again, the mean values of all the sets of constants are calculated and used in the general model.

This approach is chosen, because maybe the aggregation of several infusion periods of a patient can provide additional information about the elimination process which cannot be drawn from single infusion periods.

Identification of all records at once (P3).
In the third approach, all given data points are taken without attribution to a specific infusion period or patient and the constants are identified in one run.

3 Results

For the results, only patients are considered that have multiple data points within an infusion period. Thereafter, the data consists of 41 records and is split in a training set of 18 records and a validation set of 23 records. The training set contains data from 8 different patients with 12 different infusion periods. The mean errors in this chapter are always relating to the validation set.

The simulation results show that the concentration of Infliximab falls exponentially, almost linear, in the beginning. Then the graph almost has a kink and after that the concentration decreases exponentially, but much slower, towards zero (Figure 3). When the concentration comes close to zero after about 60 days the next infusion would be administered.

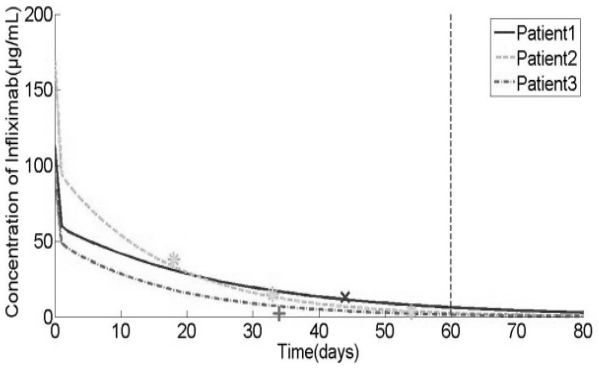


Figure 3: Simulation results of the base model with optimised parameters for three different patients and corresponding data points.

3.1 Comparison of the different parameter identification approaches

The evaluation of the mean errors of the parameter identification approaches shows that the identification per infusion period (P1) has the lowest mean error (Table 2).

Infusion periods (P1)	Patients (P2)	All Records at once (P3)
7,16	7,52	8,18

Table 2: Mean errors of the three parameter identification approaches.

These results indicate that no additional information can be extracted from putting the several infusion periods of a patient in the identification process together. A possible explanation is that single data points which lie quite aside from the others disturb the outcome of the identification process. Since there are only 8 patients, but 12 infusion periods, and the parameters are finally averaged over the number of patients respectively infusion periods, this data point has more impact on the results of the identification per patient.

From now on, we use the parameters from identification process (P2), which provided the best results, for all models.

3.2 Model with continuous ATI (M2)

The results of the model with continuous ATI with the four different transformations in comparison with the base model show that all the models with continuous ATI have higher mean errors (Table 3). That indicates that using the ATI as a continuous parameter does not improve the model.

binary	linear on [0,1]	linear on [0.5,1]	logarithmic	exponential
7,16	7,86	7,69	7,82	7,92

Table 3: Mean errors of the models with continuous ATI in comparison with the model with binary ATI.

The linear transformation onto the interval [0.5, 1] has the least mean error of the considered transformations. This leads to the conclusion that even small numbers of antibodies to Infliximab have a noticeable impact on the depletion of Infliximab in the body, because said transformation puts the small values to at least one half whereas the others put them close to zero.

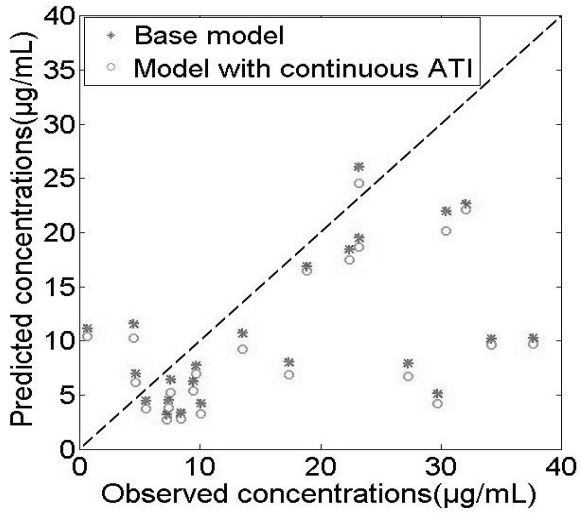


Figure 4: Comparison of the errors of the base model and model M2 with linear transformation onto [0.5, 1]

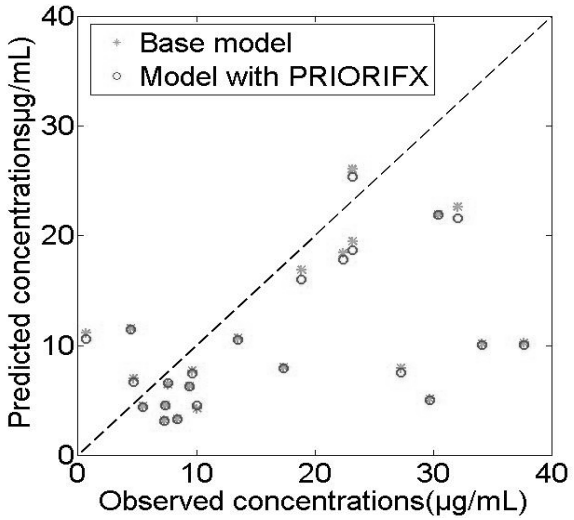


Figure 5: Comparison of the errors of the base model and model M3b

The concentration values of the model with continuous ATI are generally lower than those of the base model (Figure 4). Since the base model is already too low in most cases, this causes the bigger mean error of the given model.

3.3 Model with additional parameter (M3)

The models with the additional parameter PRIORIFX have higher mean errors (Table 4). Hence, the parameter PRIORIFX has no significant impact on the depletion of Infliximab in the body.

Base model	M3a	M3b
7,16	7,36	7,27

Table 4: Mean errors of the models with PRIORIFX in comparison with the base model.

The model M3b, whose clearance and intercompartmental flow depend on the parameter PRIORIFX, shows a slightly less mean error than the model M3a with PRIORIFX-depending clearance and constant flow Q.

In half of the cases, the model with PRIORIFX provides nearly the same results as the base model. In the other cases, the predicted values of the given model are lower than those of the base model (Figure 5).

4 Conclusion

In this paper, different models for the depletion of Infliximab in the body have been presented. The models were all two-compartment-models.

Firstly, three parameter identification approaches have been compared: identifying each infusion period separately, identifying each patient separately and identifying all data points at once. The identification of each infusion period separately has provided the best results. Hence, no additional information could be drawn from putting the several infusion periods of a patient together. Since the data set is not only small, including 8 different patients and 12 infusion periods, but also prone to measurement errors the results concerning prediction quality could be distorted, because if a data point lies quite aside from the others then it influences the optimised parameters of one of 8 patients, but only one of 12 infusion periods.

When the parameters are finally averaged, it has more impact on the patient-wise identification. To eliminate the possibility of this issue, the availability of a bigger data sample would be necessary.

In the base model, the parameter ATI was only binary. Since the data provides continuous values for that parameter, a model with continuous ATI was set up. Although different transformations of data have been tried out, the incorporation of the continuous ATI has not improved the results. This indicates that the presence of antibodies to Infliximab in the body carries more weight than the actual number of antibodies.

The number of previous infusions of Infliximab of a patient has been collected in the data. In order to examine a possible habituation effect, the parameter PRIORIFX was added to the set of parameters. However, the incorporation of the parameter PRIORIFX has not improved the results. So, the parameter PRIORIFX has no significance for the model. This leads to the conclusion that the presumption of a habituation effect is not supported.

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, [5].

References

[1] Estelberger W, Popper N. C15 – Clearance Identification – Definition. *SNE*; 2002.12 (35-36): 65-66.

[2] Macheras P, Iliadis A. *Modeling in Biopharmaceutics, Pharmacokinetics and Pharmacodynamics: Homogeneous and Heterogeneous Approaches*. First Edition. New York: Springer; 2006.

[3] Richter O. *Mathematische Modelle für die klinische Forschung: enzymatische und pharmakokinetische Prozesse*. First Edition. Berlin: Springer; 1982.

[4] Fasanmade AA, Adedokun OJ, Ford J, Hernandez D, Johanns J, Hu C, Davis HM, Zhou H. Population pharmacokinetic analysis of infliximab in patients with ulcerative colitis. *Eur J Clin Pharmacol*. 2009; 65 (12): 1211–1228. doi: 10.1007/s00228-009-0718-4.

[5] Bauer A, Urach C, Breitenecker F. Evaluation of Compartment Models for Simulation of Infliximab Depletion. In: Zajc B, Trost A, editors. ERK 2013. *Proceedings of the twenty-second International Electrotechnical and Computer Sience Conference*; 2013 Sept; Portorož. Ljubljana: Slovenia Section IEEE; P.141-144.

Impact of Counterbalance Mass on Torsional Vibrations of Crankshaft

Asllan Hajderi*, Drakuli Lumi

Department of Mechanic and Transport , 'Aleksander Moisiu' University, Durres,
District Nr. 1, Street : 'Rruga e Currilave' Durres, Albania; *ashajderi@yahoo.com

Simulation Notes Europe SNE 24(1), 2014, 27 - 32
DOI: 10.11128/sne.24.tn.10227
Received: Nov. 20, 2013 (Selected ICBTI 2013 Postconf. Publ.);
Accepted: February 1, 2014;

Abstract. In the present study it is shown how to change the main indicators of vibrations of crankshaft, in case of changing its masses. In the study it was taken the crankshaft of diesel engine with 4 cylinders in line with five main journals. To study the torsional vibration of the crankshaft system, the first it is carried the construction of equivalent reduced scheme with five discs connected to four flexible shafts. Results show that the reduced inertia moments of disks for the case with counterbalances grow to 24%. Free frequency and vibration forms, is calculated by using Holxer-Tole method. Results show that the increase of counterbalance mass leads in the reduction of vibration frequency. For the crankshaft with counterbalance the vibration form does not change, but the forced vibration amplitudes increase over 11%. Most charge part in torsion, remains the fifth shaft equally as the crankshaft without counterbalance.

Introduction

Torsional vibrations of the crankshaft are connected with the lateral system in the first set and at the end with the transmission elements of the power system. From studies it results that for the crankshaft system of tractor engines, the most dangerous form of vibration is vibration forms with four nodes, with the first node near flywheel, which is equal to the free vibration of the crankshaft with open clutch with a node near flywheel [9].

The study was obtained crankshaft of tractor diesel engine with four cylinders in line produced cast iron (Figure 1), which consists from five main journals and eight pages, forming a crankshaft with cranks in a plan.

In this crankshaft due to high loads on the main journals, it occurs a large and irregular consumption of main bearings and journals, which causes a large increase of flexible moments on pages four and five.

For a space 0,15 mm of the first main bearing and journal, which is almost the same as the consumption of middle main bearings and journals, flexible moments in fourth and fifth pages increase over 5 times until they cause his breaking in these pages, which is confirmed by the practice of using engines 75D [4]. To eliminate this phenomenon, on the basis of relevant dynamic study, it has resulted that the introduction of four counterbalances (each with 3.3 kg), its main journals gain a greater reduction of acting forces on this. In the middle main bearings and journals we have a reduction 40% of average radial load, or 13540 N [4].

By placing a counterbalances it is achieved a monotonous consumption of main journals and bearings crankshaft, which leads to reduction of flexible moments and in increasing of crankshaft life.

Figure 1: The crankshaft of tractor 75 D.

Setting of counterbalances brings a change of rotational mass, which affects in the loading state of the crankshaft from torsional vibration [1].

Vibration study of crankshaft system is associated with the movement of the piston and connecting rod group. For these complex systems, it is formed an equivalent system, which has the potential and kinetic energy equal to the real system and then are calculated the frequencies, critical speeds and resonance vibration amplitudes.

1 Construction of the Equivalent System

Equivalent system for calculating crankshaft vibrations, or reduced scheme consists of five discs with mass, connected to four elastic shafts without mass (Figure 3). Construction of equivalent system consists of [3],[7]:

- Determination of reduced inertia moments of disks
- Determination of reduced rigidities of connecting elastic shaft.

For the crankshaft without counterbalances, from [5] reduced inertia moments values of discs will be:

$$I_1 = 0.101,\ I_2 = 0.101,\ I_3 = 0.101,\ I_4 = 0.101,\ I_5 = 2.83\ [kg\ m^2]$$

While for the crankshaft with counterbalances (located on the opposite side of pages), inertia moment of counterbalance for given construction in Figure 2 will calculate:

$$J_{cp} = \rho b 2\alpha \int_{r_1}^{r_2} r^3\, dr = \rho b 2\alpha \frac{r_2^4 - r_1^4}{4} \qquad (1)$$

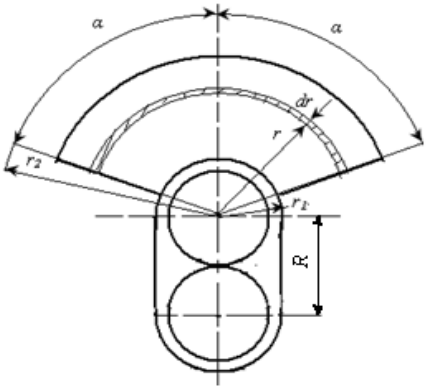


Figure 2: Schema of counterbalance.

or

$$J_{cp} = m_{cp} \frac{r_2^2 + r_1^2}{2} \qquad (2)$$

where :

$$m_{cp} = \rho\ b\ \alpha\ (r_2^2 - r_1^2) \qquad (3)$$

There mass of counterbalance, located in every crank is different and creates the add of inertia moment, $\Delta J = 10\%, 20\%, 24\%$, etc.

For counterbalance dimensions, width $b = 30\ mm$, $r_1 = 62\ mm$, $r_2 = 130\ mm$, $\alpha = 65°$, we take:

$$m_{cp} = 3.3\ kg, J_{cp} = 0.024\ kgm^2\ (\Delta J = 24\ \%),$$

So, for the crankshaft with counterbalances, reduced inertia moments values of discs will be:

$$I_1 = 0.125, I_2 = 0.125, I_3 = 0.125, I_4 = 0.125, I_5 = 2.83\ [\ kg\ m^2]$$

The reduced rigidity of elastic shafts from [5], for crankshaft by cast iron is $C = 1592356\ [\ Nm/rad]$
In this case reduced scheme to calculate the vibration of crankshaft is shown in Figure 3 (Values in parentheses are inertia moments for the crankshaft with counterbalance mass 3.3 kg).

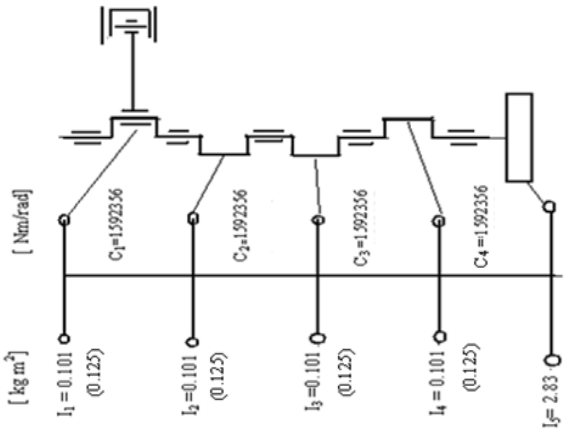


Figure 3: Reduced scheme to calculate crankshaft vibrations of counterbalance.

2 Frequencies and Free Vibration Forms of Crankshaft

To determine the frequency and free vibration forms, it is used the method without forming of differential equations of vibration. We have chosen the method Tole-Holxer [3][7], which is based on the principle of D'Alambert, that during the free vibration of system, the sum of moments of elasticity forces and inertia forces must be equal to zero. In this case the equations system of free vibration becomes a system of algebraic equations, which given [7]:

$$M(i) = M(i-1) + J(i)a(i)\omega^2$$
$$a(i) = a(i) - \frac{M(i)}{C(i)} \qquad (4)$$

where:

$M\ (i)$ - is the moment of the elasticity forces

$a\ (i)$ - is the relative amplitude of free vibrations of the disc,

ω – is the frequency of free vibrations

Under this method it is formed function :

$$R = M(n-1) + J(n)a(n)\omega^2 \qquad (5)$$

Values of ω, for which function R become to zero, are free frequencies. For this it is used the iterative method, where the cord function interrupts the abscissa axis.
This method provides a clear statement on the substance of the made calculations, directly gives the frequency and relative amplitudes of vibration and the algorithm (program) is simple.

So, the frequencies and free vibration forms were calculated according to Holxer-Tole method. Below we have analyzed only the two first frequencies, because they can operate in the area of engine speeds and the results of the calculations for various cases of disk inertia moments are shown in Table 1.

Inertia moment [kg m^2]	The first free frequency, ω_1	Relative amplitudes	The second free frequency, ω_2
J=0.091 (-10%)	1542	1; 0.865; 0.613; 0.274; -0.096	4227
J= 0.101	1471.3	1, 0.863; 0.607; 0.268; -0.098	4013,5
J=0.11 (10%)	1410	1; 0.861; 0.602; 0.260; -0.1	3829
J= 0.121 (20%)	1357	1; 0.859; 0.598; 0.253; -0.115	3653
J=0.125 (24 %)	1338.4	1; 0.859; 0.597; 0.251; -0.12	3609.5
J=0.131 (30%)	1311	1; 0858; 0.594; 0246; -0.125	3564
J=0.139 (38%)	1278	1; 0857; 0591; 0241; -0.131	3526

Table 1: Two first frequencies and relative amplitudes.

The reduction of inertia moment can be achieved by reducing the mass of the piston and connecting rod group. The change of frequency depending on the change of inertia moment is shown in figure 4.

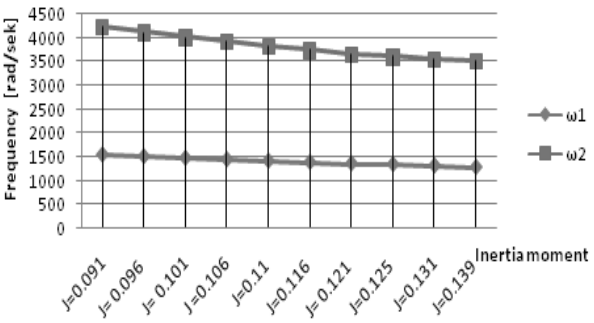


Figure 4: Change of the vibration frequency by increase of inertia moment.

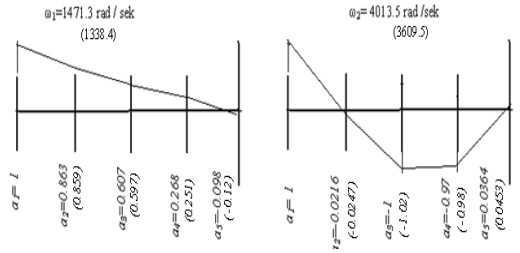


Figure 5: Two forms of free vibration of the crankshaft system.

The vibration forms for the first frequencies for crankshaft without and with counterbalance are given in Figure 5.

3 Critical Speeds of Engine

The development intensity of the torsional vibration system is determined by the value and character of the excitation moment. In this case it consists of the moment of inertia forces and gas pressure forces.
Harmonic analysis of inertia force by the approximation of Den Hartog [2] can be expressed in four harmonics:

$$T_j = m_j R\omega^2(\lambda\sin(\omega t)/4 - \sin(2\omega t)/2 - \\ -3\,\lambda\sin(3\omega t)/4 - \lambda 2\sin(4\omega t)/4\,) \qquad (6)$$

While the inertia force moment is calculated:

$$M_j = T_j R \qquad (7)$$

Harmonic analysis of the gas pressure forces moment is performed based on the principle of linear superposition, which means that vibrations caused by non harmonic moment can be seen as the sum of harmonic vibration caused by specific harmonic components of the excitation moment [7].

$$M_g(t) = M_0 + \sum_{k=1}^{n} M_{gk}\,sin(k\omega t + \alpha_k\,) \qquad (8)$$

There:

M_0 - average torque
M_{gk} - amplitude of order harmonic k
ω - angular speed of the crankshaft
$k\omega$ – frequency of order harmonic k
α_k – initial phase of order harmonic k

The amplitude of excitation moment depend on the indicatorial pressure of the engine. The average effective pressure for a speed regime of engine n_x given [9]:

$$p_{ex} = p\,(\,C_1 + C_2\frac{nx}{n} - \left(\frac{n_x}{n}\right)^2\,) \qquad (9)$$

where

$C_1 = 0.6,\ C_2 = 1.4$ (for diesel engine antechamber).

Recognizing that friction losses are constant, the pressure losses from friction are calculated:

$$P_{fx} = p_i(1-\eta) \quad (10)$$

Where η = 0.8 is the mechanical efficiency of the engine (for diesel engine without distention η = 0.7- 0.82).

Thus the ratio of the change of harmonic amplitudes of gases force, for different speed regimes will be given:

$$\frac{P_{ix}}{p_i} = 1 + \left(C_1 + C_2\frac{n_x}{n} - \left(\frac{n_x}{n}\right)^2 - 1\right) \quad (11)$$

The dangerous rate of a resonance regime depends on the work carried out by the respective harmonics of the excitation moment. Performed work by harmonic order k, for a vibration it is calculated [10]:

$$W = \pi M_k A_{k1} \sum_1^n \vec{a}\, ki \quad (12)$$

where A_{k1} is the vibration amplitude of the first disc under the action of harmonic k and $\sum_1^n \vec{a}_{ki}$ - the vector sum of the relative amplitude of free vibration, is depending on the position. To assess the most dangerous harmonics in real work, it is used relative work

$$w_r = \pi M_k \sum_1^n \vec{a}_{ki} \quad (13)$$

The resonance phenomenon occurs when one of the frequencies of the excitation harmonics becomes equal with the frequency of free vibration. Thus resonance speeds caused by i-the frequency will be determined [7]

$$n_{ik} = \frac{n_f}{k} \quad (14)$$

where $n_f = \frac{30\,\omega_i}{\pi}$, $k = 0.5, 1, 1.5, 2, ...$- excitation harmonic order for engine with four time.

K	9	9.5	10	10.5	11	11.5	12
W_r	23.99	32.66	154.73	30.97	25.43	28.67	140.37

Table 2: Relative works of harmonics.

In working speeds $n < 1500\,rpm$, free vibration of crankshaft on first frequency $\omega_1 = 1471.3\,rad/sec$ (1338.4), are excited by the harmonic of order over 9. Second frequency ω_2 is above from work speeds. Excitation power falls by increasing of the order of the harmonics, so it is used up to 24 harmonics.

Most dangerous critical speeds are determined by the relative works of these harmonics which are given in Table 2.

From Table 2 it shows that the most dangerous critical speeds are harmonics of order 10 and 12. So most dangerous critical speeds n_{10}, n_{12} depending on the change of inertia moments are given in Table 3

ΔJ	-10%	0	+10%	20%	+24%	30%
n_{10} (rpm)	1473	1405	1347	1303	1278	1253
n_{12} (rpm)	1227	1170	1127	1087	1065	1044

Table 3: Most dangerous critical speeds n_{10}, n_{12}.

Vibration amplitudes

Loading condition in torsion of the crankshaft is determined by the level of forced vibration amplitudes. The chosen method for determining of the amplitude is energy method, which is used for regimes within and outside the resonance zone [10]

According to the energy method the forced vibration amplitudes in the first disc, excited by harmonic of order k, is defined:

$$A_{k1} = A_{k0}\beta \quad (15)$$

there A_{k0} is the balance amplitude proposed by Ker Willson [10], which is calculated by:

$$A_{k0} = \frac{M_k \sum_1^n \vec{a}_{ki}}{\omega_i^2 \sum_1^n J_i a_i^2}, \quad (16)$$

and β is the dynamic factor, which is calculated by:

$$\beta = \frac{1}{1-\left(\frac{\omega_k}{\omega_i}\right)}, \quad (17)$$

ω_k - frequency of exciting harmonic of order k ($\omega_k = k\omega$).

For resonance areas ($0.9 < k\ \omega/\omega_i < 1.1$), forced vibration form is very near with free vibration. Discrepancies grow by increasing the resistance forces [10]. The resistance coefficient is taken the same for all cylinders and is calculated:

$$\zeta = \zeta' Fp\, R^2 \quad (18)$$

There ζ' - specific resistance coefficient (for diesel engine given 0.04-0.05 dN sec/cm^3), Fp – the surface of piston

Finally the real vibration amplitude of the first disc caused by excitative harmonic of order k is calculated:

$$A^r_{k1} = A_{k1}/\zeta, \quad (19)$$

and other amplitudes

$$A^r_{ki} = A^r_{k1}\, a_i \quad (20)$$

The values of the resonance vibration amplitudes of the first disc calculated for the critical speeds caused by harmonics of order 10 and 12, by changing the inertia moments of discs are given in Table 4.

Change ΔJ	-10%	0	+10%	+20%	+24%	+30%
$A_{10,1}x10^{-5}$ (rad)	369	390	405	425	434	444
$A_{12,1}x10^{-5}$ (rad)	314	337	354	374	384	395

Table 4: Each Resonance vibration amplitudes of the first crank.

While the impact of the inertia moment on critical speeds and resonance vibration amplitudes in the first disc are shown in Figure 6.

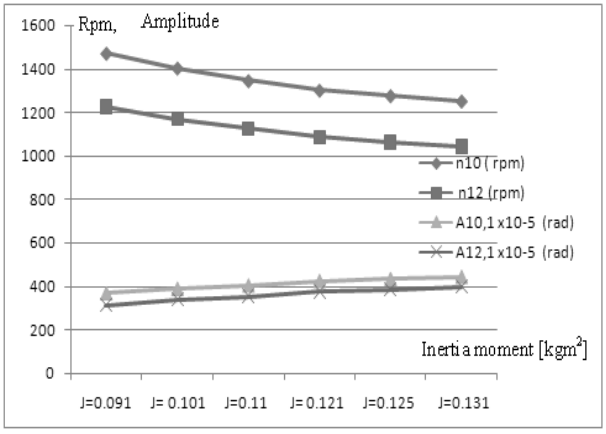


Figure 6: Critical speeds and resonance vibration amplitudes.

The vibration amplitudes of the first crank of crankshaft without and with counterbalance (ΔJ = 24%) calculated in the resonance area and outside its are given in Table 5.

4 Discussion

The construction of the equivalent system for study of crankshaft vibration has accepted a approximation of the value of the inertia moment, therefore the values of frequencies and vibration amplitudes calculated will be approximate. Holxer-Tole method is a simple method that gives accurate results in determining the frequency and relative amplitude of crankshaft vibration.

The obtained results from the calculations given in Figure 5 shows that the free vibration frequencies of crankshaft, by increasing of the inertia moments are reduced almost by linear law. While the form and relative amplitudes of vibration have very small changes.

For the crankshaft with four counterweights, the frequencies are about 10% smaller than those for the crankshaft without counterweights.

Crankshaft	without counterbalance			with counterbalance		
n (rpm)	1450	1405	1300	1300	1278	1200
$A_{10,1}x10^{-5}$ (rad)	0.248	390	0.108	0.39	434	0.13
n (rpm)	1200	1170	1100	1100	1065	1000
$A_{12,1}x10^{-5}$ (rad)	0.164	337	0.078	0.33	384	0.12

Table 5: Vibration amplitudes of the first crank of crankshaft.

The change of vibration relative amplitude of discs for the first frequency is small and the more charged sector remains the node of fifth shaft. By placing of counterweights, the change is not sensitive. The change of the relative amplitude for the second frequency is greater and the more charged sectors remain too nodes of shafts 1 and 4, but these are irrelevant, because the vibration with second frequency are only free.

Comparing the results shows that the variant with the counterweights (3.3 kg), will have a increase of inertia moments of discs up to 24%, which decreases the first frequency up to 10%.

The results given in Table 2 show that the most dangerous critical speeds are those caused by the excitations of order harmonics 10 and 12. The values of these speeds decreased with increasing of inertia moments. For the crankshaft with counterweights ($\Delta J = 24\%$) critical speeds are reduced by 10%. Increasing the inertia moment over 30% creates the risk from the impact of the order harmonic 8 of excitation moment.

The results given in Table 4 and Figure 6 shows that the resonance vibration amplitudes increase by linear law, with the increase of inertia moments. The resonance vibration amplitudes of order 12 are reduced more, than those of order 10. This shows that the dangerous torsional vibration in critical regimes decreases with increasing the order of excitation harmonics and this confirms what it is given in [10], that for vibration calculation it is enough up to 24 harmonics. Results show that for the crankshaft with counterweights, the resonance vibration amplitudes of the first crank increase 14% compared with the crankshaft without counterweights.

The results given in Table 5 shows that the vibration amplitudes of the crankshaft system away resonance regime are negligible compared with the resonance regimes, where the vibration amplitudes increase over 2000 times for order harmonic 10 and over 1500 times for order harmonic 12. This shows that during study of torsional vibration of crankshaft should be calculated only vibration amplitudes in the critical speeds, corresponding to the resonance regimes. For the crankshaft with the counterweights the vibration amplitudes away resonance regime excited by order harmonic 10, increase 2 time, and excited by order harmonic 12, increase 1.4 times. While vibration amplitudes in critical speed $n_{10} = 1278\,rpm$ created by order harmonic 10 increase 11%, and in critical speed $n_{12} = 1065\,rpm$ created by order harmonic 12, increase 14%. So, vibration amplitude in critical speed regimes are important for solidity of the crankshafts and should be performed calculation of crankshaft vibration, if their mass change [8].

The increase of the counterweight mass, that creates the add of inertia moment over 30% creates the possibility of large increase of vibration amplitude, due to the introduction of smaller excitation harmonics, which are dangerous. So, increasing of the mass over 3.3 kg leads to increasing the loading state of the crankshaft and this makes, that its solidity results uncertain.

An effective intervention that reduces the dynamic tension of torsional vibration is the introduction of flexible joints in the crankshaft system, which creates the change of frequencies and free vibration forms.

5 Conclusions

Increasing of the counterweight mass placed in crankshaft, leads in the reduction of free vibration frequencies of crankshaft, while the vibration form and the relative amplitudes remain the same level. Increase of inertia moment 24% reduce the first frequency 10%. Part more charged remains fifth shaft.

Increasing of the counterweight mass leads in reducing of engine critical speed values and can create a excitation from lower harmonics, which are more dangerous. For crankshaft with counterweight critical speeds reduce with the same rate as the first frequency.

Forced vibration amplitudes away critical speeds are not sensitive, compared with those in critical speed regimes.Vibration amplitude in critical speed regimes are important for solidity of the crankshafts.

They grow with increase of inertia moments and if their mass change it should be performed calculation of crankshaft vibration. For the crankshaft with counterweight vibration amplitudes excited by the order harmonic 10, increase 11% and those by order harmonic 12, increase 14%.

Acknowledgement

This contribution is a post-conference publication from ICBTS 2013 Conference (Durres, Albania 1-2 November 2013), co-organized by KSIM, the Kosovo Simulation society (EUROSIM Member). The contribution is a modified publication from the paper published in the Proceedings of ICBTS 2013, UBT Publications, E. Hajrizi (ed.), ISBN: 978-9951-437-24-0, at website http://conferences.ubt-uni.net/2013/digital-proceedings/icbti2013-cse-mek-proceedings.pdf.

References

[1] Baljuk G. Vibrations solidity of the engine Moscow. (Izdatelstvo Nauka). 1983; ch. 3, sec 13, p. 21-27. (in Russian)

[2] Den Hartog IP. Mechanical vibrations. Paris. 1960; ch. 5, sec. 43, p. 348-361(in French)

[3] Hafner KE. (1985) Torsional oscillation in the engine, Wien, New York (in German.)

[4] Hajderi A. (1991) Influence of Rotational Mass and Deviation from Support Axe in the Dynamic of Crankshafts. Tirane. (Thesis of PhD , Politechnic University) P.96-98.(in Albanian)

[5] Hajderi A. Impact of Material on Crankshaft Torsional Vibrations. *Journal of Mechanics Engineering and Automation*. 2014; 4(10): 799-803

[6] Hillier VAW, & Peter Coombes (2005), Engine Hillier's Fundamentals of Motor Vehicle Technology 5 th edition, Book 1. New York, ch. 2, sec 38, p. 233.

[7] Masllov GS. Axes Vibrations. Axes vibrations calculations, Manual date,Vehicle construction, Moscow (Mashinostrojenie). 1998; ch. 6, p.94-113 (in Russian)

[8] Nazarov A. Balancing of crankshaft during engine repair. Vehicles transport nr. 9. Moscow (in Russian); 1981

[9] Popik KG. Torsional vibration of crankshaft system. The dynamics of vehicles and tractors engine, Moscow, (Izdatelstvo Vijshaja scola). 1970; ch. 5, sec 43, p. 255-270 (in Russian)

[10] Willson WK. Amplitude calculation. Practikal solution of torsional vibration problem, Volume two. Chapman and Hall.LTD. London.1964; ch. 6, sec.32, p. 381-388.

The Impact of Technology on Orientation Aid for the Visually Impaired

Ines Bula*, Edin Bula, Muzafer Shala

Faculty of Mechatronics, University for Business and Technology, Lagjia Kalabria p. n., 10000 Prishtinë, Kosovo;
*bulaines@hotmail.com

Simulation Notes Europe SNE 24(1), 2014, 33 - 38
DOI: 10.11128/sne.24.tn.10229
Received: Nov. 20, 2013 (Selected ICBTI 2013 Postconf. Publ.);
Accepted: January 15, 2014;

Abstract. Statistic states that 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. About 90% of the world's visually impaired people live in developing countries.

Taking in consideration that Mechatronics is a methodology used for the optimal design of electromechanical products, and by combining technologies that are available to us we can develop a very useful tool that blind people and people with sight problems can change their lives.

Combining smart phones and digital camera there are possibilities to build smart glasses which will give information to blind people.

In this paper definitely a new ap-proach for making peoples life easy is proposed. Initially the re-sults are reached from simulation using Matlab/SIMULINK package which will lead this research to real time experimental results.

Introduction

By taking into consideration that Mechatronics is a branch of engineering that combines different disciplines of engineering from Computer Science, Mechanical engineering, Electrical engineering, Electronics, to natural sciences such as Physics and Applied Mathematics, in order to solve a particular problem at hand. This proposed product is no exception for we took pre existing technologies and modified it to our benefit. Worldwide statistic states that 285 million people are estimated to be visually impaired worldwide: 39 million are blind and 246 have low vision. About 90% of the worlds visually impaired live in developing countries.

We saw a problem and searched for possible solutions and we came up that by combining smart phones, digital cameras, GPS, 3G or 4G telephone network (that supports internet) and of course some cutting edge programming the goal of developing a tool that is able to make life's of millions a little bit better is very much achievable.

1 Reserch with Autonomus Robots (Robo Earth)

If we take in consideration that The majority of the world's 8 million service robots are toys or drive in preprogrammed patterns to clean floors or mow lawns, while most of the 1 million industrial robots repetitively perform preprogrammed behaviors to weld cars, spray paint parts, and pack cartons [2]. To date, the vast majority of academic and industrial efforts have tackled these challenges by focusing on increasing the performance and functionality of isolated robot systems. However, in a trend mirroring the developments of the personal computing (PC) industry [3], recent years have seen first successful examples of augmenting the computational power of individual robot systems with the shared memory of multiple robots. In an industrial context, Kiva Systems successfully uses systematic knowledge sharing among 1,000 individual robots to create a shared world model that allows autonomous navigation and rapid deployment in semi structured environments with high reliability despite economic constraints [4], [5]. Other examples for shared world models include research on multi agent systems, such as RoboCup [6], where sharing sensor information has been shown to increase the success rate of tracking dynamic objects [7], collective mapping of autonomous

vehicles [8], [9], or distributed sensing using heterogeneous robots [10].

However, in most cases, robots rely on data collected once in a first, separate step. Such pooled data have allowed the development of efficient algorithms for robots, which can then be used offline without access to the original data. Today's most advanced personal assistant robots rely on such algorithms for object recognition and pose estimation [11], [12]. Similarly, large training data sets for images and object models have been crucial for algorithmic advances in object recognition [13]–[14].

The architecture and implementation of RoboEarth is guided by a number of design principles, centered on the idea of allowing robots to reuse and expand each other's knowledge. To facilitate reuse of data, RoboEarth supports and leverages existing standards. The database is made available via standard Internet protocols and is based on open source cloud architecture to allow others to set up their own instance of RoboEarth, resulting in a truly distributed network. The code generated by the RoboEarth Consortium will be released under an open-source license, and will provide well documented, standardized interfaces. Finally, RoboEarth stores semantic information encoded in the World Wide Web Consortium (W3C) - standardized Web Ontology Language (OWL [17]) using typed links and uniform resource identifiers (URIs) based on the principles of linked data [15].

2 Architecture of Robo Earth

RoboEarth is implemented based on a three-layered architecture (Figure 1). The core of this architecture is a server layer that holds the RoboEarth database [Figure 1(a), the "Architecture: Database" section]. It stores a global world model, including reusable information on objects (e.g., images, point clouds, and models), environments (e.g., maps and object locations), and actions (e.g., action recipes and skills) linked to semantic information (e.g., properties and classes), and provides basic reasoning Web services. The database and database services are accessible via common Web interfaces.

As part of its proof of concept, the RoboEarth Consortium [16] is also implementing a generic, hardware-independent middle layer [Figure 1(b)] that provides various functionalities and communicates with robot-specific skills [Figure 1(c)]. The second layer implements generic components. These components are part of a robot's local control software. Their main purpose is to allow a robot to interpret RoboEarth's action recipes. Additional components enhance and extend the robot's sensing, reasoning, modeling and learning capabilities and contribute to a full proof of concept that closes the loop from robot to the World Wide Web database to robot.

The third layer implements skills and provides a generic Interface to a robot's specific, hardware-dependent functionalities via a skill abstraction layer.

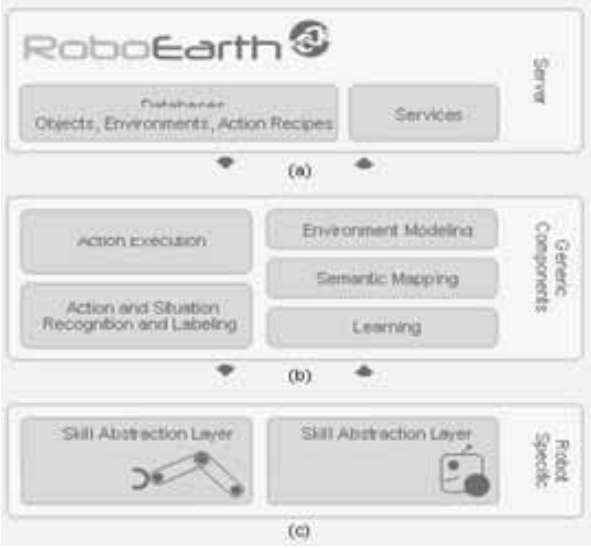


Figure1: Robo Earth's three layered architecture. [1]

3 Data Base of Robo Earth

RoboEarth stores CAD models, point clouds, and image data for objects. Maps are saved as compressed archives, containing map images and additional context information such as coordinate systems. Robot task descriptions are stored as human readable action recipes using a high level language to allow sharing and reuse across different hardware platforms. Such action recipes are composed of semantic representations of skills that describe the specific functionalities needed to execute them. For a particular robot to be able to use an action recipe, the contained skills need to have a hardware-specific implementation on the robot. To reduce redundancy, action recipes are arranged in a hierarchy, so that a task described by one recipe can be part of another more complex recipe. In addition, database services provide basic learning and reasoning capabilities, such as helping robots to map the high-level descriptions of

action recipes to their skills or determine what data can be safely reused on what type of robot.

The RoboEarth database has three main components (Figure 2).

First, a distributed database contains all data organized in hierarchical tables [Figure 2(a)]. Complex semantic relations between data are stored in a separate graph database [Figure 2(b)]. Incoming syntactic queries are directly passed to the distributed database for processing. Semantic queries are first processed by a reasoning server. Data are stored in a distributed database based on Apache Hadoop [20], which organizes data in hierarchical tables and allows efficient, scalable, and reliable handling of large amounts of data.

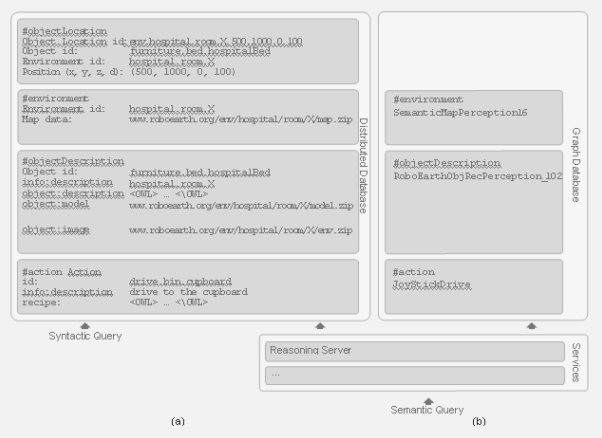


Figure2: The three main components of the RoboEarth database. [1]

Second, a centralized graph database holds semantic information encoded in the W3C-standardized OWL [17]. It stores the following data and their relations.

3.1 Objects

The database stores information on object types, dimensions, states, and other properties as well as locations of specific objects a robot has detected and object models that can be used for recognition (Figure 3). Figure 3(a) describes a recognition model for a certain kind of object (defined by the property providesModelFor), giving additional information about the kind of model and the algorithm used. The actual model is linked as a binary file in the format preferred by the respective algorithm (defined by the property `linkToRecognitionModel`). Figure 3(b) describes the recognition of a specific object. An instance of a `RoboEarthObjRec`- Perception is created, which describes that the object `Bottle2342` (linked through the property `objectActedOn`) was detected at a certain position (linked through the property `eventOccursAt`) at a given point in time using that recognition model (defined by the property `recognizedUsingModel`).

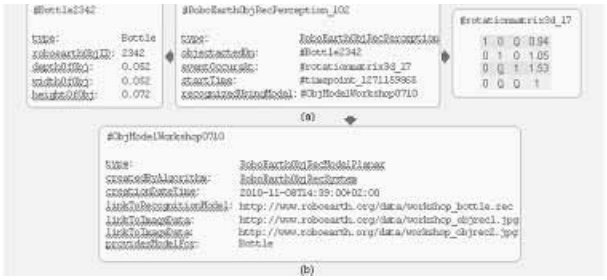


Figure3: The object description, recognition model. [1]

3.2 Environments

The database stores maps for self-localization as well as poses of objects such as pieces of furniture (Figure 4).

The semantic map combines a binary map that is linked using the linkToMapFile property with an object that was recognized in the respective environment. The representation of the object is identical to the one in Figure 3. This example shows that both binary (e.g., occupancy grids) and semantic maps consisting of a set of objects can be exchanged and even combined. The given perception instance not only defines the pose of the object but also gives a time stamp when the object was seen last. This can serve as a base for calculating the position uncertainty, which increases over time.

Figure 4: The environment map used in the second demonstrator. [1]

3.3 Action recipes

The stored information includes the list of subaction recipes, skills, and their ordering constraints required for executing an action recipe as well as action parameters, such as objects, locations, and grasp types (Figure 5). Action classes are visualized as blocks, proper ties of these classes are listed inside of the block, and ordering constraints are depicted by arrows between the

blocks. The recipe is modeled as a sequence of actions, which can be action recipes by themselves, e.g., the GraspBottle recipe. Each recipe is a parameterized type specific subclass of an action such as Translation. Atomic actions, i.e., actions that are not composed from sub actions, represent skills that translate these commands into motions.

A first type of service is illustrated by RoboEarth's reasoning server. It is based on KnowRob [18] and uses semantic information stored in the database to perform logical inference. Services may also solely operate on the database.

RoboEarth's learning and reasoning service uses reasoning techniques [19], [18] to analyze the knowledge saved in the RoboEarth database and automatically generates new action recipes and updates prior information. For example, given multiple task executions, the database can compute probabilities for finding a bottle on top of the cupboard or on the patient's nightstand. Using the additional information that cups are likely to be found next to bottles, the service can automatically create a hypothesis for the probability of finding cups on top of the cupboard.

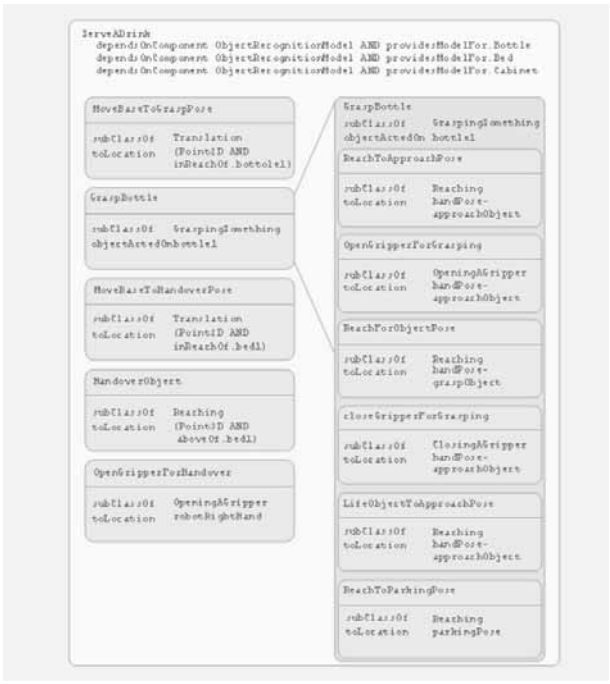


Figure 5: The action recipe used for the second demonstrator. [1]

Such cross correlations between objects can provide powerful priors for object recognition and help to guide a robot's actions. Additionally, if there are two action recipes that reach the same goal in different ways, the learning and reasoning service can detect this, fuse the recipes, and explicitly represent both alternatives.

For example, if robot A was equipped with a dexterous manipulator but robot B only with a tray, the component could create a single action recipe 'serve drink to patient' with two branches depending on the robot's abilities, which would have different requirements: the first branch would require a graspable bottle, whereas the second branch would require the availability of human or robotic help to place the bottle on the tray.

4 Proposal and Conclusion

By utilizing the avaible technologies and modifying them to our needs is mechtronics in action. It may interes you why we focust so much on robo earth.

The answer is that this kind of technology is at the very core of our product for it is easy to connect a digital camera to a smart phone and to use GPS and the internet. But it is al in vain if you don't have an image processor to analyse it and a data base to store and catologe it. All of that robo earth offers us.

The ability of the smart robots to recognize opsticles an independently avoid them in adition to their ability to move without outside aid is something we need for the realization of this project.

From the three layers only two may be needed because hardwere control layer may not be nessesary.

So if we take a smart phone that in it self has a GPS tracking system, significant processing power, storage and internet support. So if we connect a miniature digital camera with the smart phone. And that camera sends us life video from the environment to the phone. Because smart phones are like a small scale PC pre processing and cataloging can be done and then send via internet to the created cloud where the data is analysed, categorized and made available to all the users in this cloud.

So for example a visually impared person walks down the streat and and an unknown object is detected.

By means of a digital camera which records objects that are before us at a certain distance. The picture is then send sent to an application installed on the smart phone, this new application which is connected to a central data- base which one containing the list of possible objects that could be faced along the way, but with enlargement opportunities, that means if the user saw an unknown object that is not registered in the database, the object will be recorded saved, analyzed and registered in the database, and when we encounter it on the road again, it will be registered in the database and it will be available to users.

All objects that are registered in the database must be encoded in advance and each of them is given a code to identify them, therefore all new objects which we encounter and are not encoded, are recorded by the digital camera which then sends the picture to the application which then notifies the database for an object unknown, the user of the tool is then automatically notified even though we don't know what kind of object are we talking about of an potential risk in certain distance and size of that object.

At first only object of a certain size, velocity and distance will be reported to the user in order to not confuse the user. But possible extentions of servise like face regognition, regognition of everyday items, reading text and so on.

This idea of combining technologies is perfect example of mechatronics in use. Even though the idea is very ambitious for a university level it is very much feasible but huge components especially the software and database is in copyright of few companies and the only alternative is a from the scratch approach who is quite difficult if we take into consideration the resources needed to build it. But of coarse if we use different simulation software's a simulation of the product is very much possible. Initially the results are reached from simulation using Matlab/SIMULINK package which will lead this research to real time experimental results.

Next steps might be the development of similar laboratory equipments and development of software and database some of which may be accessible via an open source which can be modified to our needs and the rest of the software will be necessary to be programmed by ourselves.

Acknowledgement

This contribution is a post-conference publication from ICBTS 2013 Conference (Durres, Albania 1-2 November 2013), co-organized by KSIM, the Kosovo Simulation society (EUROSIM Member). The contribution is a modified publication from the paper published in the Proceedings of ICBTS 2013, UBT Publications, E. Hajrizi (ed.), ISBN: 978-9951-437-24-0, at website http://conferences.ubt-uni.net/2013/digital-proceedings /icbti2013-cse-mek-proceedings.pdf.

References

[1] http://www.roboearth.org/wp-content /uploads/2011/06/Waibel2011RoboEarth.pdf

[2] "World Robotics 2010," Int. Federation Robot. Statist. Dept., 2010

[3] Gates B. A robot in every home. *Scientific Amer. Mag.* 2007; 296(1): 58–65

[4] Wurman P, D'Andrea R, Mountz M. Coordinating hundredsof cooperative, autonomous vehicles in warehouses. *AI Mag*. 2008; 29(1)

[5] Guizzo E. Three engineers, hundreds of robots, one warehouse. *IEEE Spectr*. 2008; 45(7): 26–34

[6] Buchheim T, Kindermann G, Lafrenz R. A dynamic environmentmodelling framework for selective attention. In: Proc. IJCAI Workshop: Issues in Designing Physical Agents for Dynamic Real-Time Environments: World Modeling, Planning, Learning, and Communicating. 2003; P.91–98.

[7] Roth M, Vail D, Veloso M. A real-time world model for multirobot teams with high-latency communication. In: Proc. IEEE/RSJ Int.Conf. Intelligent Robots and Systems 2003(IROS'03), vol. 3; P.2494–2499.

[8] Fenwick J, Newman P, Leonard J. Cooperative concurrent mapping and localization. In: *ICRA'02*. Proc. IEEE Int. Conf. Robotics and Automation 2002 . vol. 2; P. 1810–1817

[9] Chang H, Lee C, Hu Y, Lu Y, Multi-robot SLAM with topological/metric maps. In: *IROS'07*. Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems 2007. P. 1467–1472

[10] Parker L, Fregene K, Guo Y, Madhavan R, Distributed heterogeneous sensing for outdoor multirobot localization, mapping, and path planning. In: Proc. 2002 NRL Workshop Multi-Robot Systems: From Swarms to Intelligent Automata, Washington, DC (Multi-Robot Systems: From Swarms to Intelligent Automata, A. C. Schultz and L. E. Parker, Eds.) Norwell, MA: Kluwer, Mar. 2002, P. 21.

[11] Grundmann T,Eidenberger R, Schneider M, Fiegert M, v. Wichert G. Robust high precision 6D pose determination in complex environments for robotic manipulation. In: Proc. Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation at the Int. Conf. Robotics and Automation, 2010, P. 1–6.

[12] Rusu RB, Bradski G, Thibaux R, Hsu J. Fast 3D recognition and pose using the viewpoint feature histogram. In: Proc. Int. Conf. Intelligent Robots and Systems, 2010, P. 2155–2162

[13] Everingham M, Zisserman A, Williams CKI, Van Gool L, et al. The 2005 PASCAL visual object classes challenge.In: Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment (LNAI 3944), J. QuinoneroCandela, I. Dagan, B. Magnini, and F. d'Alche-Buc, eds. Berlin: SpringerVerlag, 2006, pp. 117–176.

[14] Merler M, Galleguillos C, Belongie S, Recognizing groceries in situ using in vitro training data. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[15] Bizer C, Heath T, Berners-Lee T. Linked data—the story so far. *Int. J. Semantic Web Inform. Syst*. 2009; 5(3): 1–22

[16] v. d. Molengraft R, et al. (2010). *The RoboEarth Consortium*[Online]. Available: http://www.roboearth.org

[17] W3C OWL Working Group. (2009 Oct.). *OWL 2 web ontology Language document overview, W3C Recommendation* [Online]. Available: http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

[18] Tenorth M, Beetz M. KnowRob—Knowledge processing for autonomous personal robots. In: *IROS 2009*. Proc. IEEE/RSJ Int. Conf. IntelligentRobots and Systems 2009: P. 4261–4266

[19] Geusebroek J, Burghouts G, Smeulders A. The Amsterdam libraryof object images. *Int. J. Comput. Vis.*. 2005; 61(1): 103–112

[20] Murthy AC, et al. Apache Hadoop, a framework for reliable, scalable and distributed computing [Online]. Available: http:// hadoop.apache.org

[21] Daniels P. Analog and Digital Writing. In: *The World's Writing Systems*. 1996. 886p.

Calibration: A Usecase on the Influenza Season 2006/07 in Austria

Philipp Pichler[1,2*], Florian Miksch[2,3]

[1] Department of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *philipp.pichler@dwh.at*

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

[3] University of the Philippines Cebu, Gorordo Avenue, Cebu City 6000, Philippines

Simulation Notes Europe SNE 24(1), 2014, 39 - 46
DOI: 10.11128/sne.24.tn.10231
Received: October 3, 2013; Revised January 10, 2014;
Accepted: March 11, 2014

Abstract. Calibration deals with finding of unknown parameter values. In this paper a possible calibration approach for agent-based models is defined. After a general explanation the approach is used to calibrate an agent-based model that was developed for the Influenza Season 2006/07 in Austria. This can not only help to fit the simulation to given data, but also to increase model credibility.

Introduction

A crucial task in the process of modelling and simulation is called parametrization[1]. This is the finding of parameter values to feed the model. Usually parametrization goes along with the development of the model/simulation. First of all the system is analysed, usually this results in a huge collection of every data set that is known about the system under study, including publications, studies, other models, surveys, and others. Generally data can be split into two different groups: (1) Input Data: Everything that the model needs to be executed and (2) Output Data: Representation of the real world system after simulation runtime.

The further the modelling process (see Balci [2] or [3]) progresses the more precise the parameters have to be defined. This task fails if there is a lack of data or if values can not be measured in realtity. Calibration helps to fill this gap. What happens is that the model output is fit to real world data.

So basically data processing has to be done in two ways (1) the parametrization: Which deals with the finding and formatting of data to use it in a simulation – the data is directly transformed into parameter values - and (2) the calibration: which tries to find parameter values for the model assisted by known input and output data.

This paper focuses on the 2nd part where parameter values are not known. Basically calibration means that the value is changed repeatatly. Often the output is then compared to the real world output subjectively by an expert. In this paper the authors try to describe how a evaluation of a simulation run can be done in a more efficient way by a function that evaluates how well a simulation fits to the real world data compared to a predefined 'comparison function'. This makes things easier for the subject matter expert.

The following procedure was initially designed for epidemic models but can be adapted to fit to any model that has special conditions.

1 Theory

The data and the simulation output have to be represented in a way to be compareable to each other.

In this section, the key facts that the simulation under study should fulfil are presented. Generally there has to be a defined start and end point of the system under study. Basically the output deals with the number of entities in the system with certain attributes. An aggregated group of entities with this attribute is observed. Beyond that this section describes a function that can support the decision whether two curves coincide satisfyingly well.

1.1 Which entities are observed?

This is the aggregated group of entities that get a certain attribute A per time step. It should be defined exactly and clearly which people are represented in the given data and simulation output. If the identification between data and simulation output is not performed correctly, calibration cannot be done successfully. The entries of the vector $\vec{v}$ may represent the number of entities that have (or gain) the attribute A at time step t. The approach in this paper is defined for one attribute, but could also be extended for several attributes $A_1, .. A_n$.

1.2 Characterization of the system under study

It is not possible to give one definition that fits for all types of simulations, because there are too many factors that have to be taken into account. So basically this task has to be done dependent on every simulation. The aim of this section is to give a procedure for formulation of these characteristics.

First of all a vector $\vec{v}$ is given. This vector has entries $\overrightarrow{v_t},\ t = 1..N$ that represent for each time step the number of entities that have (or get) the attribute A.

N is the total number of time steps. (1) The start point t_{start} that has to be determined. (2) Then, a time period of the length l is defined. This is the time period that will be analyzed. (3) In this time period there can be different characteristics given as the properties P_i. These properties could be the minimum, maximum, or other functions. (4) The end of the analyzed time period is defined as $t_{end} = t_{start} + l$.

1.3 Extraction of the data/simulation vector

The time steps from t_{start} to t_{end} are extracted into a new vector with length l.

$$\vec{\hat{v}} := \left(v_{t_{start}}, v_{t_{start}+1}, \dots, v_{t_{end}-1}, v_{t_{end}}\right). \qquad (1)$$

The actual performance of the extraction must be defined individually and in respect to the properties that are obtained in 1.2 and the given vector $\vec{v}$.

1.4 Distance Function

A distance function compares two vectors $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ to each other and gives a value how good they coincide.

The time steps of these two vectors need to be given in the same step size (hours, days, weeks, …). If they have different step size, they have to be converted to the same step size. It is also very important, that these two vectors are of the same length l (in respect to the same step size). A simple approach is for example the use of a weighted square distance function between $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$.

$$\mathrm{d}(\overrightarrow{v_1}, \overrightarrow{v_2}) = \sqrt{\sum_{i=0}^{l} \omega_i * (v_{1_i} - v_{2_i})^2}. \qquad (2)$$

The weights are very helpful, if some time steps seem more 'important' than others. The weights ω_i represent the weights for time step i and have to be set manually. Generally it is advised that time steps with lower confidence get lower weights and higher confidence means higher weight.

The distance function can be chosen individually and must be adapted to the given data points. Other significant data values could also be taken into account when given.

2 The Approach

The aim of the simulation is to reproduce given data. The simulation is fed with input parameter values and the simulation produces an output. Some of the input parameters can be found others have to be calibrated. Hence, the first task is to determine the parameters which have to be calibrated. This could be either one or more parameters. Only unknown or unreliable parameters have to be calibrated. Before calibration it is very useful to do sensitivity analysis to get to know how the produced output depends on given input variables.

Upon the theory presented in 1.1 and 1.2 the output vector is extracted from the data (1.3) and stored in the vector $\vec{d}$. Then, K simulation runs are executed. K is not specified and can be chosen as required. These simulation runs are started with different values for the parameters that have to be calibrated and give several output vectors. These output vectors may be identified by $\overrightarrow{s_i}, \mathrm{i} = 1 \dots \mathrm{K}$. In every simulation run the simulation vector is extracted as described in section 1.3. The output of this process is stored in $\overrightarrow{\hat{s}_i}, i = 1..K$.

Each extracted simulation vector $\vec{s}_i$ is then compared to the data vector $\vec{d}$ using the distance function $d(\vec{s}_i, \vec{d})$ that was presented in 1.4. Then, the simulation vector $\overrightarrow{s_{best}}$ with the minimal distance function is chosen:

$$\overrightarrow{s_{best}}, \qquad \text{with } d(\overrightarrow{s_{best}}, \vec{d}) = \min_{i=1..K} d(\vec{s}_i, \vec{d}) \tag{3}$$

Calibration is an iterative task. If the simulation run $\overrightarrow{s_{best}}$ fits to the data good enough according to the distance function, calibration stops. If the distance is still too high, new simulation runs have to be started and the whole process starts all over again.

Finally, the parameter value of the simulation run $\overrightarrow{s_{best}}$ is used and calibration is finished.

Here is a short overview of this procedure.

1. Definition of the system under study.
2. Extract data vector from data upon definition
3. Calibration
   - Locate the parameters for calibration
   - Run simulations with a small amount of start infections with different parameter values.
   - Extract the simulation vector from the simulations upon definition.
   - Use the distance function to compare the extracted simulation vector to the extracted data vector. Take the parameter value of the simulation run with minimal distance function.

After calibration, a plausibility check - also called face validation - should be performed to test whether the calibrated parameter values are reasonable. This is not part of this paper and should be evaluated by a subject matter expert. For detailed information see [4], [5].

3 Calibration of the Agent-based INFLUENZA Model

Each model and each epidemic has its own characteristics. Here the calibration approach is given for an agent-based model for epidemic spreading of the influenza virus. The main characteristic of agent-based models is that complex behavior in the system arises from easy rules for each individual.

The model is built on discrete time steps. Each time step represents one day. People are represented individually as so called agents. These agents have several attributes like gender, age, infection attributes (infected, vaccinated, mild symptoms, severe symptoms…), etc. At simulation start, each agent gets initialized being either infected with or without symptoms, susceptible, or vaccinated. In each time step agents have contact with other agents. If an agent has contact with an infected person, an infection happens with a certain infection probability. After some time steps people recover. People that are recovered, vaccinated or already infectious cannot be infected again.

There is also another attribute called naturally immune that controls whether an agent can get infected. This attribute is set for persons, which cannot get infected due to an infection in a past season or due to a good immune system. The number of people that get this attribute is defined via a parameter and can be set only at simulation start.

High model credibility is very important to perform a successful calibration. That is why supportive tasks called validation and verification [3], [5] have to be carried out. A special validation strategy that is used for agent based models can be found in [4].

Since the model is built upon an object oriented approach with different modules, both tasks are quite time consuming. Definition of the influenza epidemic

If vector $\vec{v}$ contains the number of people that evolve (severe) symptoms due to an infection with the influenza virus, then each entry v_t represents the number of persons that evolve symptoms at time step t. The most important facts are that a constant c defines the official start and end of an epidemic season. The start point t_{start} is the first time step where $v_t > c$. The last time step where $v_t > c$ is called the end of the epidemic (t_{end}). We assume that $v_t \gg c$ for all $t \in [t_{start}, t_{end}]$. The constant c is important to define when an epidemic starts and ends according to the data.

The length of the epidemic is identified as $l = t_{end} - t_{start}$. One of the properties that can be found in the influenza season is that the epidemic peak is somewhere in the interval $[t_{start}, t_{end}]$. This is the maximum number of people that develop symptoms. The maximum and the length of the epidemic are important for the extraction of the epidemic in a simulation run.

Figure 1 shows the weekly number of people that consulted a physician due to influenza. Under an additional assumption we assume that this is the number of infected people that evolve severe symptoms per week. The data for the 8th week is doubted, because no explanation could be found for the decreased number of cases. It is assumed, that this is an error in the data. In this example the task to find the start and the end of the influenza season does not have to be done, because the definition was made upon the given data and data was preprocessed in a way that start and end is already given.

The main information that this figure gives, are: the influenza starts in the 3rd week of the year ($t_{start} = 3$) and reaches its maximum between the 7th and 9th week. The actual assumption is, that the maximum is exactly in week 8. The end is in the 13th week ($t_{end} = 13$). That's a duration of $l = 11$ weeks (77 days).

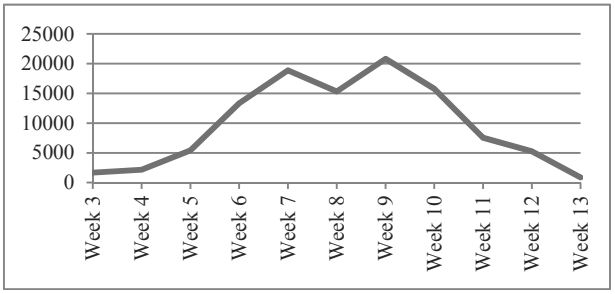


Figure 1: Number of people that evolve severe symptoms per calendar week in influenza season 2006/07 in Austria.

3.1 Extraction from the data

The time steps from t_{start} to t_{end} are extracted into a new vector with the length l.

$$\vec{d} := \left(v_{t_{start}}, v_{t_{start}+1}, \dots, v_{t_{end}-1}, v_{t_{end}}\right). \quad (4)$$

Here, no extraction is necessary because the data is already given in the correct format hence, $\vec{d} = \vec{v}$.

3.2 Calibration procedure

The aim of this section is to show how the calibration task can be done in an efficient way, but not to deliver the perfect calibration utility for this model.

In literature many strategies for model calibration can be found that may be applied [6-8]. Calibration always depends on how much information is available.

Locate the calibration parameter

Several epidemiological studies allow parameterization of the model except for the infection probability, which cannot be measured, hence it needs to be calibrated. The calibration results are shown in section 3.3.

Some parameters like population data or disease progression are highly reliable while others like the percentage of naturally immune people might be scrutinized. The calibration of the parameters infection probability and naturally immune is shown in 3.4.

Run simulations

In reality, spreading of the influenza virus starts with a small amount of infected people until the epidemic officially begins. This is why the simulation runs are initialized with a small number of infectious people.

To make it more reliable, the simulation time should be longer than the actual epidemic. It should cover at least as many time steps so an extraction of the epidemics upon the definition in 0 is possible.

The vector $\vec{s}$ represents the simulation output. Each entry s_t represents the number of people that get severe symptoms at time step t according to simulation.

Extraction

For the use of the distance function, which is applied in the next section, it is important that the finally extracted epidemics in the data and simulation output

1. are of the same length and
2. the time steps represent the same interval (daily, weekly, monthly).

The extraction procedure presented here takes care of these two points. The extraction of the simulation runs and the extraction of the data are two separate procedures. In this section the length of the epidemic and the entry of the simulation run with the highest number of people that newly develop severe symptoms is used for detection.

The duration of the epidemic is important for the detection of the epidemic in the simulation. The simulation has daily-sized time steps. This is why the detection of the epidemic is performed on days. According to the definition of the epidemic the duration of the influenza season as given in the data (Figure 1) is about 77 days (11 weeks), that is why 77 time steps are picked in the simulation run.

Then, the sum of 7 time steps represents a week to be comparable to the original data.

That and the fact, that the simulation is started with a lower number of infected people inquires to take a longer simulation period for the detection of an epidemic. The detection of the epidemic has to be done for each simulation run that was started in 0.

Example for the extraction

To show how the extraction is performed a simulation run is executed, where $\vec{s}$ represents the simulation output and $N = 170$ is the simulation runtime (daily step size). The result of this run is shown in Figure 2.

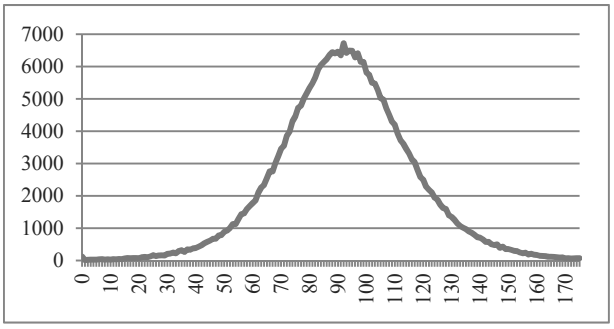


Figure 2: Simulation run with 150 time steps (daily). Occurrence of severe symptoms per day.

First of all, the maximum amount of severe symptoms per time step has to be detected. It is possible to use the maximum function for this detection. If we zoom in (Figure 3) it is obvious, that the maximum time step is at 92.

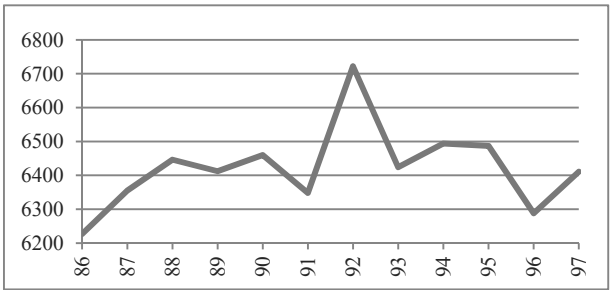


Figure 3: Zoomed in simulation run (daily).

Agent based models underlie some variations, hence it makes sense to smoothen the results. Here, the smooth vector $\vec{\bar{s}}$ calculates by the mean value of three time steps (Figure 4).

$$\bar{s}_t = \begin{cases} \frac{s_t + s_t + s_{t+1}}{3}, & \text{if} \quad t = 0 \\ \frac{s_{t-1} + s_t + s_{t+1}}{3}, & \text{if} \quad t = 1..N-1 \\ \frac{s_{t-1} + s_t + s_{t+1}}{3}, & \text{if} \quad t = N \end{cases} \quad (5)$$

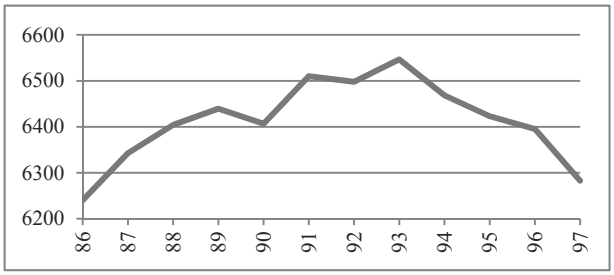


Figure 4: Smoothed simulation run (zoomed in).

Then the maximum of the vector $\vec{\bar{s}}$ is detected. In the example this is marked with the red line and is at time step $t_{max} = 93$. It could be possible, that the maximum is very close to the beginning or the end of the simulation time. This could happen in three cases:

1. There is no significant uprising of the number of people that evolves severe symptoms time step. No maximum can be found.
2. The simulation run time is too short. Then the maximum is at the end. Simulation has to be restarted with a bigger N and re extracted.
3. The percentage of start infections too high. Simulation has to be restarted with a lower percentage of start infections and re extracted.

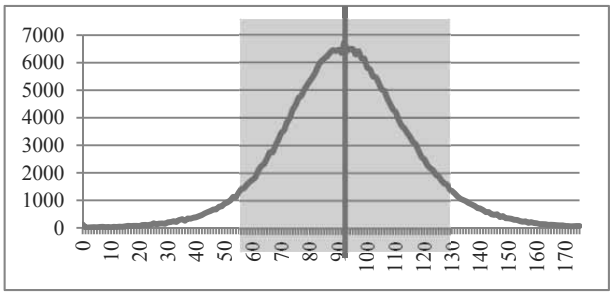


Figure 5: Detected epidemics (daily).

After the time step of $\mathrm{t_{max}} \coloneqq \mathrm{t_{I_{max}}}$ is detected, all $\mathrm{s_t}$ with $t \epsilon \left[t_{max} - \frac{1}{2}, t_{max} + \frac{1}{2}\right]$ are extracted into a new vector.

This vector is represented as

$$\vec{\hat{s}} := \left(s_{t_{max}-\frac{1}{2}}, s_{t_{max}-\frac{1}{2}+1}, \dots, s_{t_{max}+\frac{1}{2}-1}, s_{t_{max}+\frac{1}{2}}\right) \quad (6)$$

All s_t with $t\epsilon\left[92-\frac{77}{2}, 92+\frac{77}{2}\right]$ are stored in the vector $\vec{\hat{s}}$. These are the red marked time steps shown in Figure 5.

The extracted epidemic is shown in Figure 6. This is a vector of the length l.

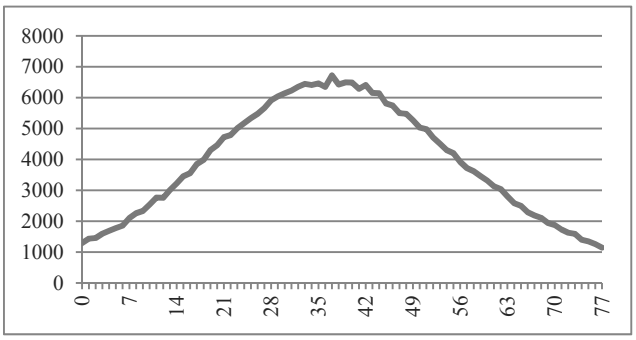


Figure 6: Extracted epidemics (daily).

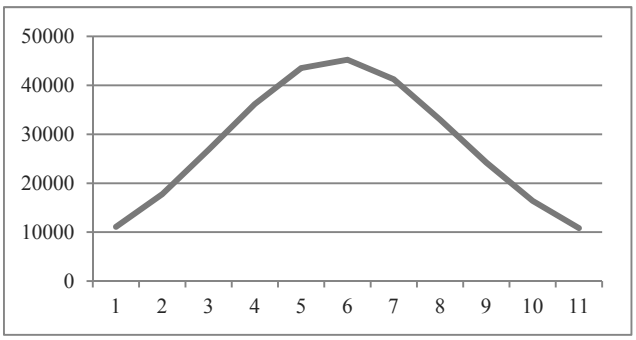


Figure 7: Extracted epidemics per week.

Now each 7 time steps are summed up to get an output in the same step size as given in the data (Figure 7).

Applying the distance function

Use the distance function to compare the extracted epidemic simulation results to the extracted data. Take the parameter value of the simulation run with minimal distance function.

3.3 Results of the calibration of one parameter

For a correct calibration a wealth of simulation runs has to be executed. The data that is shown in Figure 1 refers to the population of 2007, these were about 8.300.000 people. To run an agent-based model with this number of agents takes quite long, that is why for calibration the number of agents is reduced to 830.000 and the data is scaled to this amount of people. This has no impact on the calibration process, because the number of agents is still high enough to produce reliable results to work with.

As already mentioned the infection probability can not be measured so this is the parameter that is varied in the calibration process.

A series of simulation runs $\vec{s_i}, i = 1 \dots K$ is started. All simulations are executed with a low amount of initial infections and different values for the infection probability. In each run the epidemic is detected and stored in $\vec{\hat{s}_i}, \mathrm{i} = 1..K$. The simulation runs are then compared to the original data and the distance function is evaluated.

Some expressive simulation runs are shown in Figure 8. The given data is the red line. The other simulation runs are the detected epidemics for each parameter value. Of course not all simulation runs can be shown here, so this is only a sample set of all runs.

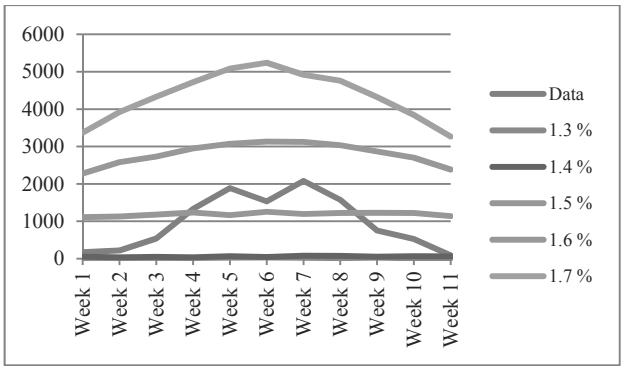


Figure 8: Calibration of infection probability.

The weighted distance function as given in section 40 is used. It is supposed, that the data point of the 6th week is wrong or insufficient. That is why these weeks get a lower weight.

week	1	2	3	4	5	6	7	8	9	10	11
weight	1	4	8	16	42	4	42	16	8	4	1

Table 1: Weights per time step.

Then the distance function is applied. Each simulation run is executed and the distance to the data is given in the following table.

infection probability	distance
1.3 %	3 905.98
1.4 %	3 813.02
1.5 %	2 333.31
1.6 %	6 273.50
1.7 %	11 248.15

Table 2: Distance to given data.

Now the simulation run with the minimal distance is chosen. This is the one with an infection probability of 1.5 % and is stored in $\overrightarrow{s_{best}}$.

Still, the calibration results are not satisfying. The main problem is that far too many people evolve severe symptoms at the beginning and at the end in every simulation. Another point of view is that the model produces too long epidemics using the fixed parameters. Variation of the infection probability does not help to overcome this issue.

3.4 Calibration of two parameters

Now, the same procedure is performed by varying two parameters, the infection probability and the number of naturally immune people.

The simulation runs in Figure 8 show that a higher value for the infection probability leads into an increase of people with severe symptoms at all and a higher value of the maximum of people that evolve severe symptoms.

The number of naturally immune people controls what percentage of the population gets the attribute to be naturally immune at initialization. These people cannot get infected at all.

Sensitivity analysis of this parameter shows that a higher amount of naturally immune people in the beginning leads to less infections, less people that evolve severe symptoms and a shorter duration of the epidemic in the simulation. The results of the sensitivity analysis are not presented here.

Another series of totally 10 000 simulation runs $\overrightarrow{s_1}, i = 1 \ldots K$ is executed, and the epidemics are detected and stored in $\overrightarrow{\hat{s}_1}$.

The infection probability is varied between 0.6 % and 8.80 % and the percentage of people that are naturally immune is varied between 50% and 90%. Due to lack of space not all results can be shown here. In Figure **9** an extract of simulation runs is shown to provide a little insight how close the results of simulation runs with different parameter values are.

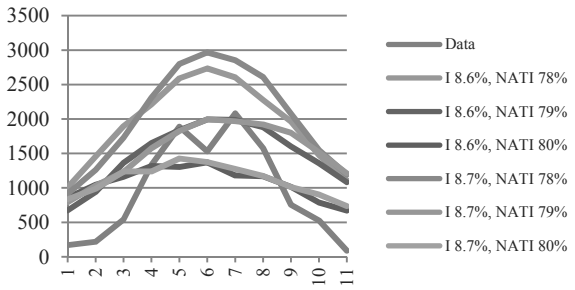


Figure 9: Variation of infection probability (I) and percentage of people with natural immunity (NATI).

In the Table 3 the distance of the extracted simulation runs to the data is shown. The distance function (section 40) uses the same weights as given in Table 1.

infection probability	percentage of natural immune people	distance
8.6 %	78 %	9.800.55
8.6 %	79 %	4 659.97
8.6 %	80 %	7 661.01
8.7 %	78 %	11 658.65
8.7 %	79 %	5 022.45
8.7 %	80 %	6 877.16

Table 3: Distance to given data.

The best simulation $\overrightarrow{s_{best}}$ has an infection probability of 8.6 % and a percentage of start infections of 79 %.

It would be very difficult to choose one of these runs manually because of the large number of runs and a small variation of parameter values results in very similar output as shown in Figure 9. Of course it is not possible to say objectively, that this simulation is really the best representation of the real data, but it helps to decide whether parameter values can be found, that represent the data in a good way or not.

Based on the results, experts have to assess the found parameter values for a final decision of a reliable simulation which represents the data satisfyingly well.

4 Conclusion

Calibration is a crucial task when building a model. It helps to determine whether a model is able to represent the original in a reliable way. The calibration method and especially the examples of the calibration process that are presented here can help to reconsider assumptions that were made in the model, or to start investigations concerning the correctness of the data. If calibration of a parameter can be done with a subjectively good result it will result in even more confidence for the model. It is important to mention, that calibration can not be finished without a plausibility check from a subject matter expert, that gives input, if the found parameters are feasible.

References

[1] Popper N, Pichler P. *Agent-based modeling and simulation in archaeology: Reproducibility*. Springer, 2013. 269 p.

[2] Balci, O. A life cycle for modeling and simulation, *SIMULATION* 2012;88(7): 870–883.

[3] Sargent R. Verification and validation of simulation models. In: *Proceedings of the 2010 Winter Simulation Conference*; 2010 Dec; 166–183.

[4] Klügl F. A validation methodology for agent-based simulations. In: *Proceedings of the 2008 ACM symposium on Applied computing*; 2008; New York. 39–43.

[5] Balci O. Validation, verification, and testing techniques throughout the life cycle of a simulation study. 215–220.

[6] Schade W, Krail M. Modeling and Calibration of Large Scale System Dynamics Models: The Case of the ASTRA Model. In: *Proceedings of the 24th International Conference of the System Dynamics Society; 2006*; Nijmegen. 3210–3226.

[7] Bohensky E, Smajgl A, Herr A. Calibrating behavioural variables in agent-based models: Insights from a case study in East Kalimantan, Indonesia. In: *MODSIM 2007 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand*, 2007.

[8] Abbaspour KC. Calibration of Hydrologic Models: When is a Model Calibrated?. In: *MODSIM 2005 International Congress on Modelling and Simulation*, 2005. 2449–2455.

Support of Event-Graph Lectures by the MMT E-learning System

Martin Bicher[1,*], Gasper Music[2], Irene Hafner[3], Felix Breitenecker[1]

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

[2]Faculty of Electrical Engineering, University of Ljubljana, Slovenia

[3]dwh Simulation Services Vienna, Austria

SNE Simulation Notes Europe SNE 24(1), 2014, 47 - 50
DOI: 10.11128/sne.24.en.102 33
Received: Dec. 10, 2013 (Selected ERK 2013 Postconf. Publ.);
Accepted: February 20, 2014;

Abstract. In a lot of technical professions nowadays an advanced education in mathematical modelling and simulation has already become a prerequisite. Therefore universities are required to provide high quality lectures and teaching materials to guarantee the qualification of their graduates within this field of study. For engineering students surely the emphasis has to be laid on discrete-event modelling, as the understanding of event-operation and scheduling provides the base for designing management and control strategies. Often the concept of event-graph modelling is used to teach the basis of discrete-event simulation in a graphical way. The paper shall give an idea on how methods of event-graph modelling can be taught to students using blended learning concepts. In this special case the e-learning platform MMT (Mathematics, Modelling and Tools) is used, a kind of virtual laboratory for modelling and simulation. The basic ideas of event-graph modelling, how event-graph based examples are prepared for the e-learning platform and the basic properties of the MMT-server are furthermore presented in this paper as well as some simple examples are shown in addition.

Introduction

In a lot of technical professions nowadays an advanced education in mathematical modelling and simulation has already become a prerequisite. Therefore universities are required to provide high quality lectures and teaching materials to guarantee the qualification of their graduates within this field of study.

For engineering students surely the emphasis has to be laid on discrete-event modelling, as the understanding of event-operation and scheduling provides the base for designing management and control strategies.

Regarding discrete-event simulation one of the most challenging parts of the modelling process is the formal description of the model, which is on the one hand essential for the implementation and on the other hand necessary for a comprehensible documentation of the modelling process. One of the most figurative description forms for these kind of models is the so called event-graph modelling, formally described in Section1. This kind of representation strategy is extremely suitable for queuing processes.

Although event-graphs are very simple to understand they are yet very flexible and really complex behaviour can be observed. They are suitable for deterministic as well as stochastic models.

In order to support lectures dealing with event-graphs targeted e- or blended-learning should be used. Common e-learning platforms usually provide the opportunity to offer teaching material and sometimes create tests with multiple-choice or numerical questions, where students can proof their knowledge. However, regarding the pictorial representation and the complex and maybe stochastic behaviour of event-graph models, these opportunities might not be sufficient. Therefore the blended-learning system MMT (*Mathematics Modelling and Tools*) system, in detail explained in Section2, was developed by dwh GmbH Vienna. The platform is based on the idea, that students can additionally experiment on pre-implemented, accurately tested and high quality simulation examples, by changing certain parameters in a browser window. The model is calculated on a fast server in behind and the results are presented either in textual, image or video form directly within the browser window.

In order to integrate user friendly event-graph examples to the MMT server, a new event-graph library was developed in MATLAB.

1 Bases of Event-Graph Modelling

Aforementioned event-graphs are probably the most popular ways to describe a discrete-event model in figurative form. Yet, those models have to fulfil the following conditions in order to be described by an event-graph:

1. Only a finite number of events may occur during simulation-time.
2. All observed state variables remain constant during the time between each two sequential events. These variables change their value only directly at the events.
3. The model is dynamical respectively time-dependent.
4. The time when an event takes place can depend on parameters, random variables and the state variables itself. Therefore the events can be triggered by a so called event-list, which is updated each time the variables change.

The first two conditions are based on fundamental concepts for discrete-event simulation. Usually the set of events can be clustered by condition of occurrence and change of the state variable. Events caused by the same conditions and causing the same change of the state variable are hereinafter called event-type.

Satisfying the aforementioned conditions a discrete-event model is defined by a finite set of possible event-types, linked to conditions when they occur and how they change the state variables, and a finite set of state variables with corresponding initial conditions at $t = t_0$. These ideas inspire defining so called nodes, usually represented by circles - one for each of the event-types. The impact of the event-type on the state variable is quoted usually directly below the node. The causal relationship between the event-types, necessary to consequently trigger an event, is defined by so called edges, represented by connecting arrows. Conditions for the occurrence of an event are quoted directly above the arrow. These might be e.g. time delays and or causal conditions. Before we introduce an example the formal description of an event graph (see also [4]).

1.1 Event-graph formalism

During the following formal description we mainly use the notation of Yucesan and Schruben 1992 [3]. Therefore a quadruple
$G = (V(G), E_s(G), E_c(G), \Psi_G)$ is called event-graph or simulation-graph if...

- $V(G) = \{E_1, \ldots, E_k\}$ is a finite set of event-types (called event-vertices),
- $E_s(G) = \{s_1, \ldots, s_{n_s}\}$ is a set of scheduling edges,
- $E_c(G) = \{c_1, \ldots, c_{n_c}\}$ is a set of cancelling edges (For modelling reasons $E_s(G) \cap E_c(G) = \emptyset$ holds. The set $E(G) := E_s(G) \dot{\cup} E_c(G)$ shall be called the set of all edges of the event graph.),
- $\Psi_G : E(G) \to V(G) \times V(G)$ is a, not necessarily injective, function, called incidence function, assigning an ordered pair of vertices to a given edge.

The edges, each linked to two vertices (or event-types) by the incidence function, are responsible for the causal relationship between those two events. They denote if, how and when the second of the two events is scheduled (or cancelled from the schedule) after the occurrence of the fist one. So far the basic construct of the event-graph is defined which does not contain any definition how state changes take place.

Let $\Omega \in \mathbb{R}$ be the so called state-space, containing all possible states of the model. Furthermore a model M based on an event-graph G is called event-graph model if the following Functions seen in Table 1 are defined and used. Finally the seven-tuple $M = (\mathscr{F}, \mathscr{C}, \mathscr{T}, \Gamma, \mathscr{P}, \mathscr{A}, G)$ is called event-graph model. It is well-defined and can be simulated in a unique way by any event-graph simulation software.

Function	Use
$\mathscr{F} : V(G) \times \Omega \to \Omega$	Specifies the state change for each event-type.
$\mathscr{C} : E(G) \times \Omega \to \{0,1\}$	Specifies the condition of the edge.
$\mathscr{T} : E(G) \times \Omega \to \mathbb{R}^+$	Specifies the delay time of the edge.
$\Gamma : E(G) \times \Omega \to \mathbb{R}^+$	Specifies priorities regarding execution of the edge.
$\mathscr{P} : V(G) \times \Omega \to \mathbb{R}^+$	A function defining event parameters.
$\mathscr{A} : E(G) \times \Omega \to \mathbb{R}^+$	A function defining edge attributes.

Table 1: Functions for an Event-Graph Model.

Surely this completely theoretical is very difficult to understand. Therefore, similar to most simulation techniques, the idea of event-graphs is best introduced studying examples.

1.2 Multiple server queue

Figure1 shows how a part of how the browser screen looks like, when opening the MMT-example 'Multiple Server Queue'. It deals with an implementation of a classical model, perfectly suitable for event-graph novices. The model deals with a queue of objects focussed on the length of the queue. Directly after the queuing process the objects are treated by one of k different servers. There are many ways to interpret this model with real life aspects. One can think of ...

... goods waiting to be processed by one of k-robots.
... data packages waiting for being handled by a k-kernel processor.
... customers waiting in a shopping mall in front of k cash desks.

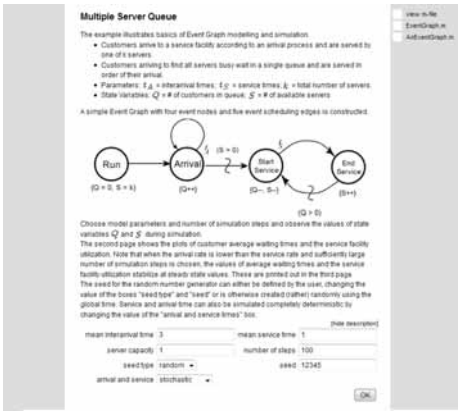


Figure 1: Browser screen of the MMT showing the multiple server queue example.

In all cases the same set of parameters is required:

1. a number of servers k
2. a rate of incoming objects t_A
3. a service time t_S
4. an initial set-up

We furthermore focus on the event-graph illustration in the centre of Figure1 and explain the ideas of this event-graph. First of all the set of all possible event-types is given by

$$V(G) = \{\text{Arrival}, \text{Start service}, \text{End service}\}$$

as between those events the queue is idle. Initially by convention also always a so called 'Run' event is scheduled necessary to start the simulation. In this simple example the positions, orientations and conditions of the scheduling edges are nearly self explanatory if one thinks about the causal relationships between the event-types.

The lower section of the browser window seen in Figure1 is reserved for experimenting with input parameters for the simulation. In this case the modelling type can be changed from deterministic (i.e.deterministic arrival and service time) to stochastic using $t_A \sim Exp(\text{mean}(t_A))$ and $t_S \sim Exp(\text{mean}(t_S))$. The exponentially distributed random numbers are gained by a random number generator and can either be calculated by a 'random' seed (dependent on the global time), or by a user defined seed.

Pressing the 'ok' button, in this case, MATLAB is started in the background calculating the multiple server queue with the chosen input parameters. An output example plot is shown in Figure2. Arrival time (mean 1) and the service time for 8 servers (mean 10) are distributed exponentially as well. Before going into specific code details a closer look at the MMT server is taken.

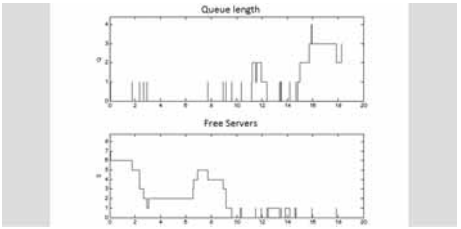


Figure 2: Results of the multiple-server queue. Length of the queue can be seen as well as the occupancy of the servers.

2 MMT Server

The MMT-server was developed and is maintained by dwh-Vienna and is mainly used for lectures at the Vienna University of Technology. Additionally it is used within external courses for modelling and simulation

too. The web-interface provides the basis for high quality e-learning. Students as well as lecturers benefit. The most important properties of the server are presented here (see also [1] and [2]).

2.1 Virtual laboratory

The MMT system provides the opportunity to upload MATLAB, Simulink (Simscape, Simmechanics), Java and AnyLogic models to create a virtual laboratory. The models can be called from the internet-browser, simulation takes place in the background and the results are then sent back to the browser window again in text, picture and/or video form. Specified parameters for the models can be changed directly within the MMT-page by changing the values in provided HTML text-boxes. Thus programming skills are not required for experimenting with high quality modelling and simulation examples. Currently about 500 MATLAB, 50 Simulink, and 30 Java/Anylogic models can be tested and experimented. Examples written within the free-ware programming languages R and Octave are planned to be included too.

2.2 Upload download of files

As other e-learning systems, also MMT provides the opportunity to offer lecture notes, images or pdf explanations for download. Using an unique node-based course-oriented administration system grants, that each participating student gets access only to those files intended for her/him.

2.3 Improve programming skills

As programming skills are becoming more and more important within technical professions the MMT server provides the opportunity downloading each source code, used for the models, and manipulate it at the home-pc. The administrators of the server work hard to guarantee that every code fulfils high quality standards and is fully commented.

3 Summary and Outlook

Regarding examples like the multiple server queue, the event-graph library improves the usage of the MMT server, as discrete event simulation is a very important topic within the area of control design modelling. In collaboration with the University of Ljubljana the Vienna University of Technology several other event-graph examples were implemented during the last years, which were in consequence uploaded to the MMT system increasing the teaching value of the platform. Currently we plan to extend the usage of the MMT system in order to be able to simulate more serious discrete event models e.g. by including libraries for treating DEVS formalism.

Acknowledgment

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, [5].

References

[1] Hafner I, Bicher M, Winkler S, Fitsch U. MMT - An E-Learning System based on Computer Numeric System for teaching Mathematics and Modelling. In Troch I, Breitenecker F, editors. Preprints MATHMOD 2012 Vienna - Full Paper Volume. *MATHMOD 2012* 2012 Feb; Vienna. Vienna: ARGESIM and ASIM.

[2] Hafner I, Bicher M,Peterseil T, Winkler S, Fitsch U, Nagele-Wild N, Wild W, Breitenecker F. Achievements in result visualization with the computer numeric e-learning system mmt. In Troch I, Breitenecker F, editors. Proceedings of the European Modeling and Simulation Symposium, 2012. *THE 24TH EUROPEAN MODELLING & SIMULATION SYMPOSIUM* 2012 Sept; Vienna. Genoa, Italy: ARGESIM and ASIM.

[3] Yücesan E, Schruben L. Structural and behavioral equivalence of simulation models. *ACM Trans. Model. Comput. Simul.*. 1992. doi: 10.1145/132277.132281.

[4] Chick S, Sánchez P, Ferrin D, Morrice D, Schruben L, Roeder T, Chan W, Freimer M. ADVANCED EVENT SCHEDULING METHODOLOGY

[5] Bicher M, Music G, Hafner I, Breitenecker F. Support of Event-Graph Modeling by the MMT E-learning System. In: Zajc B, Trost A, editors. ERK 2013. Proceedings of the 22nd International Electrotechnical and Computer Science Conference. 2013 Sept; Portorož; Ljubljana: Slovenia Section IEEE. P. 161-164.

Comparison of Finite Difference Method and Random Walk Method in ARGESIM Benchmark C19 'Pollution in Groundwater Flow'

Michael Ebner , Stefanie N. Winkler*

Inst. of Analysis and Scientific Computing, Vienna University of Technology,
Wiedner Haupstraße 8-10, 1040 Vienna, Austria; * *stefanie.winkler@tuwien.ac.at*

Simulation Notes Europe SNE 24(1), 2014, 51 - 54
DOI: 10.11128/sne.24.bn19.10235
Received: October 3, 2013; Revised January 10, 2014;
Accepted: March 11, 2014;

Abstract. Groundwater represents one of the most important sources so as to satisfy the steadily increasing demand of pure water in modern times. However, groundwater is very susceptible to many kinds of pollution whose causes can usually be divided into one of two categories: point-source and nonpoint-source pollution.
In this comparison a particular focus was put on the modelling of a 2D-homogeneous groundwater body and the contamination of its groundwater stream caused by a steady point-source pollution in case of a uniform pore-water velocity. Three different tasks were regarded: In task A, the pollution propagation was investigated and compared to an approximated analytical solution in case that no treatment plants are installed. In contrast, task B and C consisted of examining the impact of treatment plants on the actual pollution propagation in case of a permanent activation and when the pollution reduction works according to a set schedule instead. In total, two different computational approaches were chosen and implemented in Matlab whereby one consisted of a finite difference method and the other was based on a random walk ansatz. Similar results were obtained but further parameter studies could be helpful.

1 Modelling & Task Definition

The modelling in this comparison is restricted to the 2D-description of the pollution propagation within a homogeneous groundwater body which suffers a continuous pollution by a point-source. Hence, from another perspective, it is assumed that the actual groundwater body can be approximated by a 2D-domain whereby the concentration in z-direction does not differ. Furthermore, the associated groundwater stream shall be characterized by a uniform velocity. For this setting the propagation of the pollution concentration $c = c(x, y, t)$ with unit g/m^2 can be described by the 2D transport equation

$$\frac{\partial c}{\partial t} = \alpha|\mathbf{u}|\nabla^2 c - \mathbf{u}\cdot\nabla c - \lambda c, \qquad (1)$$

where α denotes the constant dispersivity [m], $\mathbf{u}$ the uniform vector-valued pore velocity [m/s] and $\lambda \geq 0$ the degradation [1/s]. Therefore, the first term on the right-hand side of (1) represents the diffusion and the second the convection term whereas the last term corresponds to the sink within the domain. Moreover, without limiting the generality, the point-source shall be located in the origin $(0,0) \in \mathbb{R}^2$ for further consideration. Similarly, it can be assumed that the pollution only propagates in x-direction, i.e. $\mathbf{u} = (u, 0)$ which can be achieved by affine transformation.

In the course of this comparison the transport equation (1) shall be solved numerically by the application of a finite difference method (FDM) and, a possibly more intuitive approach, the random walk (RW) method.

1.1 Task A: Unaffected pollution spread

The first task consists of the simulation of the pollution spread associated with the assumptions as described above. In this particular case, an analytical approximation can be stated: By assuming a steady source of pollutant on an infinite area where no sink exists, i.e. degradation $\lambda = 0$, the analytical solution of (1) can be approximated through

$$c(x, y, t) = \frac{c_0}{4\sqrt{\pi\alpha r}} e^{(x-r)/(2\alpha)} \operatorname{erfc}\left(\frac{|\mathbf{u}|t}{\sqrt{2\alpha|\mathbf{u}|t}}\right) \qquad (2)$$

with the definitions

$$c_0 := \frac{M}{h\, n_e\, |\mathbf{u}|}, \quad r := \sqrt{x^2 + y^2},$$

and the (complementary) error functions

$$\operatorname{erfc}(\phi) = 1 - \operatorname{erf}(\phi)$$

according to [1].

The associated parameters M, h, n_e represent the input rate of pollutant mass, the thickness of the saturated flow and the effective porous volume respectively.

The computed results of both the FDM and RW based on the parameters of Table 1 shall be compared among themselves and with the approximated solution (2) at $t = 50$ and $t = 150$ days.

Variable	Description	Value
$\boldsymbol{u}$	Pore velocity in x-direction	$10^{-5}\mathrm{m}$
$\boldsymbol{\alpha}$	Dispersivity	$0.05\mathrm{m}$
$\boldsymbol{\lambda}$	Degradation	0
$\boldsymbol{M}$	Input rate of pollutant mass	$0.002\mathrm{g/s}$
$\boldsymbol{h}$	Thickness of saturated flow	$10\mathrm{m}$
$\boldsymbol{n_e}$	Effective porous volume	0.25

Table 1: Notation and description of used parameters.

1.2 Task B: Pollution reduction by facilities

In this task the impact of treatment facilities on the pollution shall be investigated. Therefore, it is assumed that a direct pollution reduction at the pollution source is not possible but treatment facilities can be installed at certain locations in order to reduce the pollution locally. In reality, this might be achieved by wells with chemical substances, pumps blowing in oxygen etc.

In order to reduce the pollution two plants of the treatment facilities are located at $(40\mathrm{m}, \pm 5\mathrm{m})$. The actual reduction of the pollution is modelled with the degradation parameter λ: Within a circle neighbourhood of 5m around each plant the degradation is set to $\lambda = \lambda_{\mathrm{on}} := 10^{-6} \ln(10)\, 1/\mathrm{s}$ which corresponds to a decrease in pollution concentration of 10% each $\delta t = 10^6\mathrm{s}$ based on the assumed exponential pollution sink due to (1), i.e. $e^{-\lambda\, \delta t} = 0.1$.

The algorithms are supposed to be applied to the asymptotic solution obtained by task A. The computed results of both the FDM and RW shall be compared with task A at different locations after $t = 100$ days of active pollution reduction.

1.3 Task C: Controlled pollution reduction

Task C aims to model the scenario in which a reduction of operational costs is sought while maintaining a low pollution level is still desired. The setting for the comparison is equal to task B, but as opposed to $\lambda = \lambda_{\mathrm{on}}$ permanently, the following strategy is applied instead: The degradation λ equals 0 from Monday to Friday between 8am and 8pm. Apart from these periods the degradation λ corresponds to $\lambda_{\mathrm{on}} = 10^{-6} \ln(10)\, 1/\mathrm{s}$ at the respective areas around the facilities. The results of both the FDM and RW shall be discussed at $(x, y) = (50\mathrm{m}, 0)$ for $0 \le t \le 150$ days. In fig. 1 the situation is depicted for all tasks. The next chapters introduce the computational approaches for solving (1): The FDM and RW whereas both are implemented in Matlab.

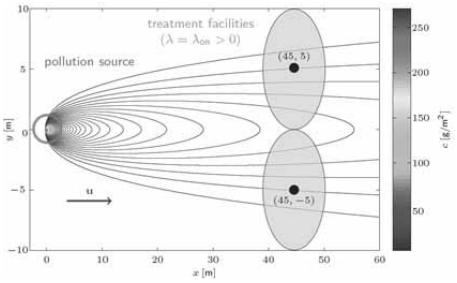


Figure 1: Pollution concentration obtained by (2) after $t = 150$ days with parameters of table 1 (task A). The green regions represent the 5m-circle neighbourhood of the treatment plants at $(40\mathrm{m}, \pm 5\mathrm{m})$ (task B and C).

2 Finite Difference Method

Mesh grid. By assuming equal steps h in x- and y-direction the mesh of the regarded domain shall be given by the nodes $(x_i, y_j) \in \mathbb{R}^2$. Furthermore, the approximation of the pollution concentration $c(x_i, y_j)$ at the nodes $(x_i, y_j) \in \mathbb{R}^2$ shall be denoted by $c_{i,j}$, i.e. $c_{i,j} \approx c(x_i, y_j)$. Thus, the inner nodes are described by (1) whereas for the nodes located at the boundary of the domain one may either set reasonable boundary conditions (b.c.) or assume the domain to be sufficiently large and then impose zero b.c. In this comparison the latter was chosen. Hence, for the comparison, simply the domain of interest is plotted whereas a larger domain was computed. The mesh grid was chosen to be $\{(x_i, y_j) : -10 \le x_i \le 100, -20 \le y_j \le 20\}$ whereby step size $h = 1/4\mathrm{m}$ for the computation of all tasks. In the following, the discretization techniques of the involved terms in (1) are regarded separately.

Diffusion term. Given that both the steps in x and y are equal to h the second order discretization of the Laplace operator ∇^2 by means of a five-point stencil reads

$$\nabla^2 c_{i,j} = (c_{i+1,j} + c_{i-1,j} - 4c_{i,j} + c_{i,j+1} + c_{i,j-1})/h^2$$

for inner domain nodes. By using a lexicographical order for the indices and bearing in mind the zero b.c. the five-point can be represented as block tridiagonal matrices which were implemented as sparse matrices.

Convection term. Since $\mathbf{u} = (u, 0)$ it is reasonable to use a backward difference for the discretization of the gradient operator ∇. Thus, it holds that

$$\mathbf{u} \cdot \nabla c_{i,j} = \left(u \cdot \frac{\partial c_{i,j}}{\partial x} \right) \approx u \cdot \frac{(c_{i,j} - c_{i-1,j})}{h}.$$

Again, the resulting block tridiagonal matrix was stored in sparse format in Matlab.

Time derivative. The remaining time derivative in (1) was implemented by means of an explicit Euler with time step $\Delta t = 2$ hours.

Evolution. The initial concentration was set to zero for all nodes and the concentration at the origin $(0,0)$ got augmented by $M\Delta t/h^2$ each time step Δt.

3 Random Walk

Modelling. The second method, the random walk, may be regarded as a more intuitive approach: Instead of solving the PDE (1) directly, transition probabilities are inferred which model the convection and diffusion behaviour of pollution *particles*. More precisely, the pollutant is modelled by means of a finite number of particles which execute a *deterministic convective* and a *probalistic dissipative* movement: With (p_x^n, p_y^n) denoting the coordinates of one single particle at time $t = n\Delta t, n \in \mathbb{N}_0$, the particle movement is defined as

$$\begin{cases} p_x^{n+1} = p_x^n + \sqrt{2\alpha u \Delta t} Z_x + u\Delta \mathrm{t} \\ p_y^{n+1} = p_y^n + \sqrt{2\alpha u \Delta t} Z_y \end{cases} \quad (4)$$

for one time step Δt whereby Z_x and Z_y represent standard, normally-distributed random variables. For the modelling each particle corresponds to an appropriate amount of pollution mass which depends on the step size and pollution source. Thus, there is no need for neither a mesh grid nor for collision rules. This type of approach has already been modelled in [2]. However, the procedure presented in here follows the explanations in [3] and therefore is defined slightly differently.

Pollution concentration. Even though no mesh grid is necessary for the computation, the mesh associated with the FDM is used in order to compare both methods. Each node is then assigned the pollution concentration given by the mass determined by the amount of particles located within an ε-neighbourhood divided by its area $\varepsilon^2\pi$. For task B and C the pollution reduction is modelled by particle weights ω: If a particle entered the scope of any treatment plant within Δt, it is set $\omega = e^{-\lambda \Delta t}\omega$ whereby $\omega = 1$ at $t = 0$. The pollution reduction then corresponds to a reduced weight of the particle. In further consequence, the pollution concentration at each node can be obtained by considering the weights of each particle.

Evolution. It is assumed that additional 100 particles enter at the origin $(0,0)$ each time step $\Delta t = 2$ hours whereby the pollutant mass per particle is set to $M\Delta t/100$. The radius for the pollution concentration computation corresponds to $\varepsilon = 0.5$m.

4 Results

Task A: Unaffected pollution spread.

In Figure 2 the pollution concentrations according to both numerical approaches and the approximated solution (2) after $t = 50$ days are illustrated.

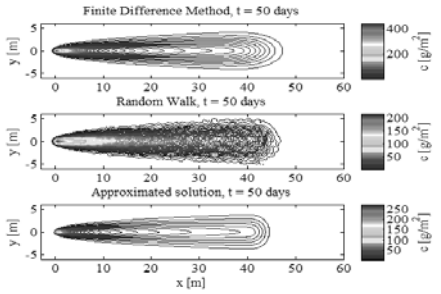


Figure 2: Task A: Pollution after 50 days for all methods

It can be witnessed that all results are qualitatively similar. However, especially at the vicinity of the pollution source the predicted concentrations differ which can also be observed in Figure 3. Moreover, FDM and RW agree quite well beyond approximately $x \geq 5$m. Nevertheless, the computed results of FDM and RW depend crucially on their parametrization, i.e. the choice of input rate ($M\Delta t/h^2$ at FDM, $M\Delta t/100$ at RW) and ε for the ε-neighbourhood (RW). Therefore, other values might yield a better alignment among all methods. For computational time, RW is more advantageous in general.

Task B: Pollution reduction by facilities.

The figures 4 and 5 show the impact of the pollution reduction facilities. At $x = 30$m virtually no pollution reduction is recognized for both methods whereas their effect becomes the more manifest the higher the distances are. At $x = 50$m the concentration has already dropped by more than a half compared to task A.

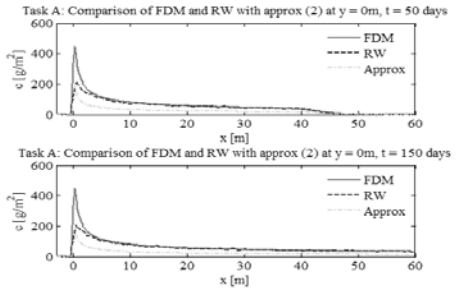


Figure 3: Task A: Pollution after 50 and 150 days at $y = 0$m.

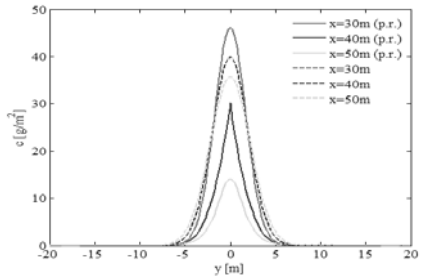


Figure 4: Task B – FDM: Pollution after 100 days at 30m, 40m and 50m with and without pollution reduction facilities.

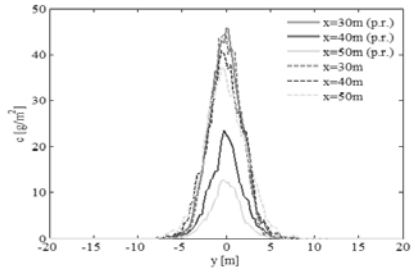


Figure 5: Task B – RW: Pollution after 100 days at 30m, 40m and 50m with and without pollution reduction facilities.

Task C: Controlled pollution reduction.

With respect to the controlled pollution reduction (fig 6 and 7) it can be witnessed that the pollution decreases steadily for both the FDM and RW until it remains small but fluctuates around 20g/m^2. As opposed to the case of using a permanent pollution reduction according to task B (indicated by blue circles at some time points) the pollution concentration is slightly higher. However, a still substantial decrease is achieved and the benefit in cost savings may outweigh the higher pollution level.

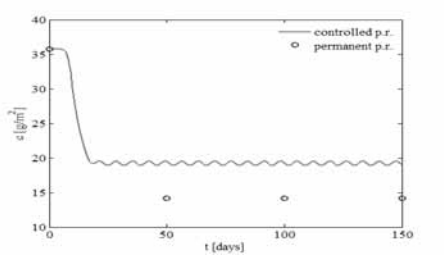


Figure 6: Task C – FDM: Comparison of pollution spread at $(50\text{m}, 0\text{m})$ for permanent and controlled pollution reduction.

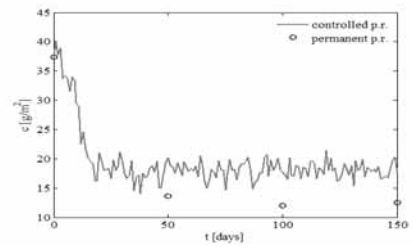


Figure 7: Task C – RW: Comparison of pollution spread at $(50\text{m}, 0\text{m})$ for permanent and controlled pollution reduction.

5 Conclusion

Both the FDM and RW yielded qualitatively comparable results. However, their actual figures depend considerably on the implementation of the input pollution rate at the origin. Hence, parameter studies could be necessary in order to find the right parametrization according to a particular problem in reality.

Model sources

Finite Difference Method and Random Walk Method are directly programmed in MATLAB. All MATLAB m-files and a short file documentation can be downloaded (zip format) by EUROSIM societies' members from SNE website, or are availably from the author.

References

[1] Judex F, Breitenecker F, Höfinger G. 'Pollution in Groundwater Flow' – ARGESIM Benchmark C19R with Spatially Distributed Modelling. *Simulation News Europe*. 2006; 16(3): 57-60.

[2] Höfinger G, Judex F, Breitenecker F. Pollution in Groundwater Flow – Definition of ARGESIM Comparison C19. *Simulation News Europe*. 2005; 44(45): 51-53.

[3] Winkler SN. *Comparative Mathematical Modelling of Groundwater Pollution* [Master thesis]. [Institute for Analysis and Scientific Computing, (AT)], Vienna University of Technology; 2014.

Petri Net Modelling and Simulation in AnyLogic and MATLAB for ARGESIM Benchmark C4 'Dining Philosophers'

Ruth Leskovar[1,*], Johannes Tanzler[1], Martin Bicher[1,2]

[1] Inst. of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; [*] *ruth.leskovar@tuwien.ac.at*

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

Simulation Notes Europe SNE 24(1), 2014, 51 - 54
DOI: 10.11128/sne.24.bn04.10237
Received: June15. 2013; Revised December 2, 2014;
Accepted: January 4, 2014;

Abstract. To analyse the "Dining Philosophers" comparison Petri Net models are used and implemented in two different libraries based on the simulation environments, MATLAB and AnyLogic are used. Different strategies are investigated to solve this logical problem. Therefore the visualisation in the used MATLAB library is very helpful but the easy application of the AnyLogic library allows a fast adaption and testing of different strategies.

Introduction

This benchmark 'Dining Philosophers' [1] describes the situation of five philosophers sitting around a table meditating. Like every human being also philosophers have the need to eat. Following from the definition of this comparison it is not possible that all philosophers eat simultaneously.

A task of this comparison is to investigate if every philosopher gets enough food and if this is not the case to create strategies in which this is ensured. The optimum strategy is found when every philosopher gets about the same amount of food.

According to the comparison description the method of Petri Nets has to be applied for analysing this situation. In this particular project two different libraries for Petri Nets are used, on the one hand a library created by Gaspar Music [3], an associate Professor of the University of Ljubljana, Slovenia, which is based on MATLAB, and on the other hand a library in AnyLogic [4].

1 Model Description

The initial situation is that five philosophers are sitting around a large round table and everyone has a bowl of food in front of him and between each of the bowls a chopstick is placed.

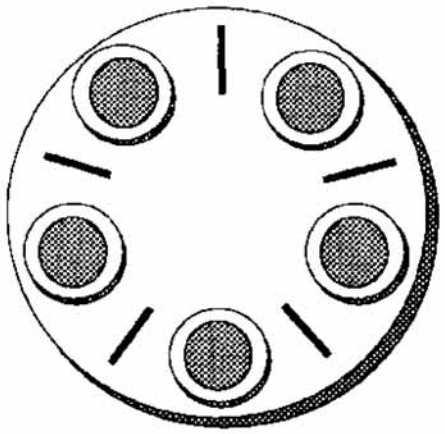


Figure 1: Philosopher's table [1].

All philosophers are either meditating or eating. If a philosopher wants to eat, both chopsticks beside his bowl have to be available. Each philosopher takes the chopstick to his left first and if available afterwards the one on his right. If a philosopher has accomplished getting both chopsticks he starts to eat and then he starts meditating again. Philosophers need lots of food because of their hard mental work and so they are hungry again immediately after they have started meditating.

It is obvious that if all philosophers are meditating at the same time, all of them grab the left chopstick simultaneously in the next step. But the problem is that then there are no chopsticks left and therefore no one has a chopstick available to his right.

For analysing and solving this problem Petri Nets [2] are used to describe this model. Petri Nets are used for a qualitative analysis of a system. Every Petri Net consists of places, transitions, arcs and tokens. Arcs always connect places with transitions. The tokens are located at the places and if predefined logical conditions concerning the transitions are fulfilled, tokens can change the places. Tokens can only change their positions along the direction of the arcs. Concerning the logical conditions a transition can only fire if all places which are connected by arcs with directions to the transition are taken with tokens. Further a transition, connected with places by arcs in direction to the places, adds a token to each of those places.

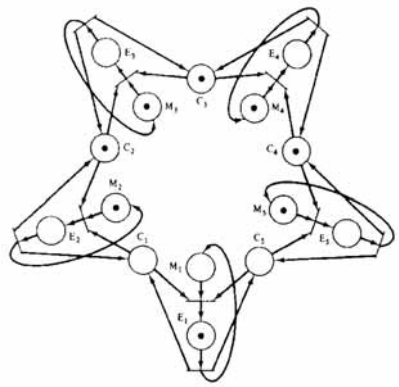


Figure 2: Philosopher's Petri Net [1].

Figure 2 shows a Petri Net of the philosophers sitting around the table. The places E_i represent the eating state and the M_i represent the meditating state of each philosopher. The places C_i stand for the chopsticks. If a token is placed in a place, it means that this state is active respectively that the chopstick is available. Figure 2 does not include the theory that every philosopher first takes the chopstick to his left and in this case just one philosopher is eating and the others are meditating.

Like mentioned above, if the preference of the left hand is considered, it is possible that at some point no one can get a second chopstick and eat. The first strategy to solve the problem is to add a hungry state. This means that after meditating the philosopher gets hungry and then takes the left chopstick and if then the chopstick to his right is available, he eats.

Also in this case a deadlock can occur and so the strategy 'gentleman' is added. This means that one philosopher is very gentle but also very fast in his actions. If this particular philosopher gets hungry, he does not take the left chopstick first.

2 Implementation

2.1 AnyLogic

For the implementation in AnyLogic a programming example of the AnyLogic Help is used. This example provides a possibility to work with Petri Nets. It has places, transitions and arcs which just have to be connected properly and in this case a little bit adapted. The places and transitions are implemented as agents. As an additional opportunity it is possible to use this as well for timed Petri Nets. But in this model the time is set to 1 and so it is equivalent to a not timed Petri Net.

2.2 MATLAB

For the implementation in MATLAB a Petri Net library from Gaspar Music is used. First of all the initial marking of the tokens and the number of places have to be defined. Then the transitions have to be defined. This means how many transitions are needed and in which direction they are connected to the places. Simultaneously in this step the arcs are defined automatically. For a proper graphical output the positions of the arcs and the transitions have to be set. For this model the library has to be adapted slightly.

3 Results

First of all the Petri Net was implemented with an initial marking as shown in Figure 2. Figure 3 shows in which state the philosophers are at the end of the simulation run.

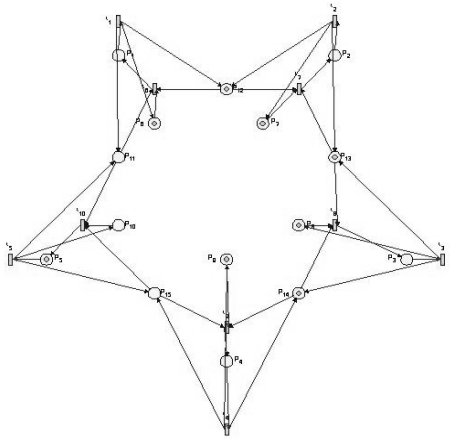


Figure 3: Petri Net – MATLAB.

Obviously one philosopher is eating while the others are meditating. In this scenario, with this initial marking, it is not possible that more philosophers are eating at the

same time because the chopsticks are not available immediately after eating as a result of passing the Transitions. In Figure 4 the places `Eating` and `Meditating` of each philosopher are presented.

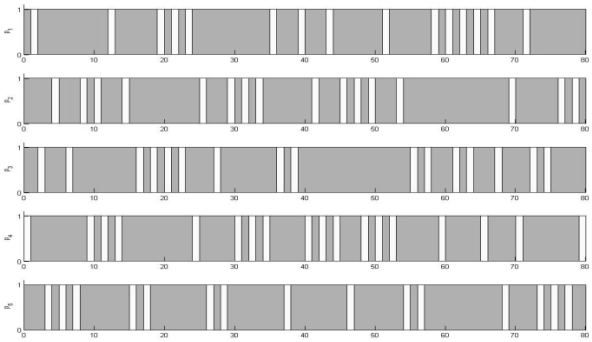


Figure 4: Places Eating and Meditating – one Philosopher eating – MATLAB.

The black bars mark the places `Meditating` and the yellow bars represent the places `Eating`. Clearly each philosopher stays in one of these two places. No deadlock can appear under those circumstances.

Using an initial marking where two philosophers are eating it is easy to see that then alternating two philosophers are eating and none is eating. Nevertheless with both initial markings the philosophers get the same amount of food. In Figure 5 the result of the simulation with two philosophers eating is presented.

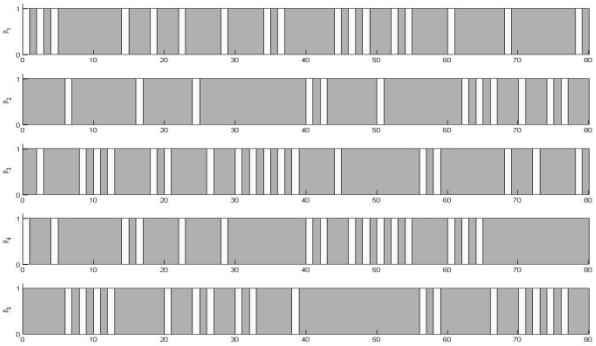


Figure 5: Places Eating and Meditating – two Philosophers eating – MATLAB.

In the definition of the project is mentioned that the Philosopher do not take both chopsticks at once but take first the left. So in a further model an additional place `Left Hand` is implemented for each Philosopher. In this scenario each simulation leads to a deadlock i.e. all philosophers stop in the place `Left Hand`. In Figure 6 for each Philosopher the places `Left Hand` and Eating are plotted, the black bars mark the `Left Hand` and the yellow bars mark the `Eating Places`.

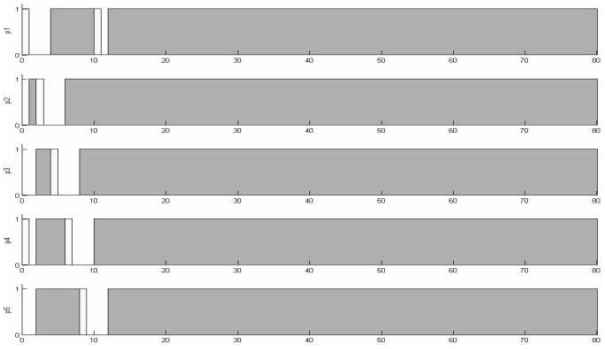


Figure 6: Places Eating and Left Hand – MATLAB.

To handle this problem an extra place `Hungry` is added for each philosopher. This place is scheduled between the places `Meditating` and `Left Hand` and therefore the chopstick can only be taken if `Hungry` is active. This strategy does not prevent the possibility of a deadlock.

On the one hand this depends on the initial marking on the other hand it depends on which transition is randomly chosen. For analysing this scenario an initial marking with two philosophers eating is chosen because all the possible outcomes if just one philosopher is eating are included and investigated. There are three possible outcomes for this model. One possibility is a deadlock this means in this case that all philosophers have their left chopstick in their left hand.

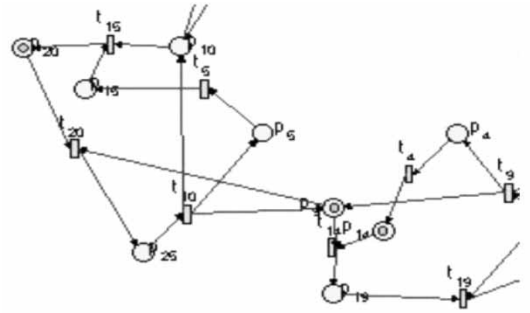


Figure 7: Transition Conflict – MATLAB.

In Figure 7 key marking is shown on which the outcome of the simulation depends. One philosopher holds his left chopstick in his left hand (`Place20`) and his right neighbour is hungry at this moment (`Place14`). Between them one chopstick is available (`Place9`) now it depends if the `Transition14` or the `Transition20` will fire. This situation is called a transition conflict when the firing conditions are fulfilled for two different transitions with the same Token. In MATLAB as well as in AnyLogic it is randomly chosen which Transition will fire.

If the `Transition14` fires the token jumps to the `Place19` which represents the `left hand`. To get from place `Left Hand` in the Eating state the right chopstick is needed. If the right chopstick is not available and no other philosopher is eating a deadlock occurs.

The second possibility occurs if the `Transition20` fires the token then jumps in the `Place25` which means that the philosopher eats and the right neighbour stays hungry. If no other philosopher is eating then the other four philosophers hold their left chopsticks in their hands and so the neighbour to his left can start to eat by taking the right chopstick. This eating loop then goes on counter clockwise.

The third possibility is similar to the second possibility. In this case the situation which is explained above occurs several times at different steps, which leads to an additional loop and so the eating frequency is increased like illustrated in Figure 8.

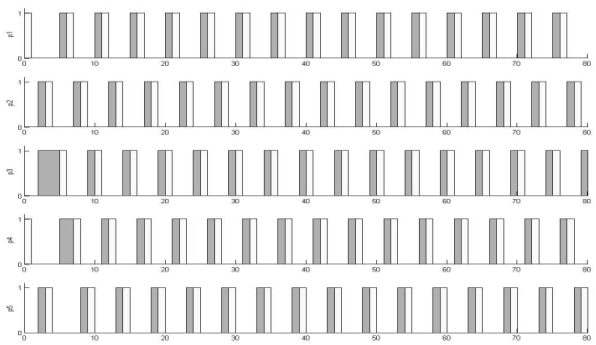


Figure 8: Places Eating and Left Hand – two Philosophers eating – MATLAB.

To prevent the remaining deadlocks another strategy is implemented. In this case one of the philosophers, called the 'gentleman', takes both chopsticks at a time and does not take the left chopstick first. Therefore if not the second philosopher to his left or he himself eats at least the one to his left can eat and so again a counter clockwise loop occurs.

4 Conclusion

To summarise, the advantages and disadvantages of both libraries are described. As usual the drag and drop method of AnyLogic does not require much previous knowledge and is easy to expand. Additionally the visualisation of the simulation is already included, in contrast to the MATLAB library where additional programming is necessary for proper visualisation. But for the better understanding of the Petri Net the MATLAB library is easier to follow because the Tokens are visualised in the places and not in the transitions like in AnyLogic. Additionally it should be mentioned that it is not possible to exclude deadlocks in this Petri Net model by using these implementations but needs further analysing.

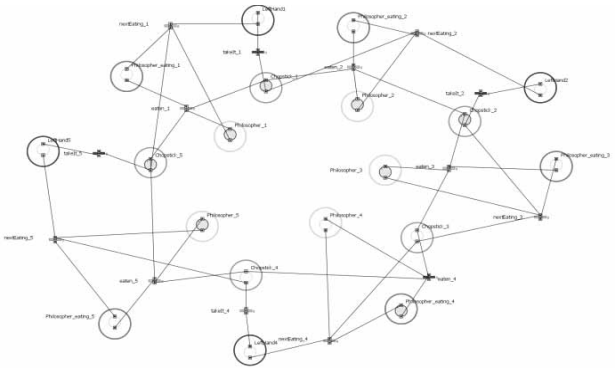


Figure 9: Gentleman Petri Net – AnyLogic.

In Figure 9 the Petri net in AnyLogic for this strategy is shown where the `Philosopher3` is the 'gentleman'.

Model sources

In both simulators AnyLogic and MATLAB special modules for Petri net modelling are used: in AnyLogic a template for Petri nets is used (part of AnyLogic), in MATLAB Music's external toolbox was used. Model parametrization files for both simulation systems, a short file description and a link to the MATLAB Petri net toolbox are can be downloaded (zip format) by EUROSIM societeties' members from SNE website, or are availably from the author.

References

[1] SNE Editors. Comparison 4: Dining Philosophers Problem. EUROSIM - Simulation News Europe. 1991; 3(1), 28 p.

[2] Priese L, Wimmel H. Petri-Netze. 2th ed. Berlin: Springer; 2008. 374 p.

[3] Music G, Hafner I, Winkler S, Skrjanc I. A MATLAB based Petri net Tool for E-learning: Examples for timed simulation and scheduling. In: Breitenecker F, Troch I. Full Paper Preprint Volume of the 7th Vienna International Conference on Mathematical Modelling. 7th Mathmod; 2012 Feb; Vienna. Vienna: ARGESIM and ASIM.

[4] The AnyLogic Company, 2014, AnyLogic Example model. Petri Nets. AnyLogic 7 University 7.0.2.

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

Simulation Notes Europe SNE is the official membership journal of EUROSIM and distributed / available to members of the EUROSIM Societies as part of the membership benefits. **SNE** is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With **Online SNE** the publisher **ARGESIM** follows the **Open Access** strategy for basic **SNE** contributions. Since 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 additional **SNE** contributions are available via membership in a **EUROSIM** society.

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

SNE Editorial Office /ARGESIM

→ *www.sne-journal.org, www.eurosim.info*
office@sne-journal.org (info, news)
eic@sne-journal.orgt 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 *Austria, Germany, Switzerland*
CEA-SMSG	Spanish Modelling and Simulation Group *Spain*
CROSSIM	Croatian Society for Simulation Modeling *Croatia*
CSSS	Czech and Slovak Simulation Society *Czech Republic, Slovak Republic*
DBSS	Dutch Benelux Simulation Society *Belgium, Netherlands*
FRANCOSIM	Société Francophone de Simulation *Belgium, France*
HSS	Hungarian Simulation Society *Hungary*
ISCS	Italian Society for Computer Simulation *Italy*
LIOPHANT	LIOPHANT Simulation Club *Italy & International, Observer Member*
LSS	Latvian Simulation Society *Latvia*
PSCS	Polish Society for Computer Simulation *Poland*
SIMS	Simulation Society of Scandinavia *Denmark, Finland, Norway, Sweden*
SLOSIM	Slovenian Simulation Society *Slovenia*
UKSIM	United Kingdom Simulation Society *UK, Ireland*
KA-SIM	Romanian Society for Modelling and Simulation, *Romania, Observer Member*
ROMSIM	Romanian Society for Modelling and Simulation, *Romania, Observer Member*
RNSS	Russian National Simulation Society *Russian Federation, Observer Member*

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) *esko.juuso@oulu.fi*
Past President	Khalid Al.Begain (UKSIM) *kbegain@glam.ac.uk*
Secretary	Borut Zupančič (SLOSIM) *borut.zupancic@fe.uni-lj.si*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
SNE Repres.	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

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 *felix.breitenecker@tuwien.ac.at*

→ *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 AutomaticControl department,
University of Valladolid,
Real de Burgos s/n., 47011 Valladolid, SPAIN

CAE - SMSG Officers	
President	M. À. Piera Eroles, *MiquelAngel.Piera@uab.es*
Vice president	Emilio Jiminez, *emilio.jiminez@unirioja.es*
Repr. EUROSIM	Emilio Jiminez, *emilio.jiminez@unirioja.es*
Edit. Board SNE	Emilio Jiminez, *emilio.jiminez@unirioja.es*
Web EUROSIM	Mercedes Peres, *mercedes.perez@unirioja.es*

Last data update December2013

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 Bozikov, *jbozikov@snz.hr*

Last data update December2012

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 nam. 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 December2012

DBSS – Dutch Benelux Simulation Society

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

→ www.eurosim.info

a.w.heemink@its.tudelft.nl

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

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

Last data update April 2006

FRANCOSIM – Société Francophone de Simulation

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

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

→ *www.eurosim.info*

y.hamam@esiee.fr

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

FRANCOSIM Officers	
President	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 Jávor,
Budapest Univ. of Technology and Economics,
Sztoczek u. 4, 1111 Budapest, Hungary

HSS Officers	
President	András Jávor, *javor@eik.bme.hu*
Vice president	Gábor Szűcs, *szucs@itm.bme.hu*
Secretary	Ágnes Vigh, *vigh@itm.bme.hu*
Repr. EUROSIM	András Jávor, *javor@eik.bme.hu*
Deputy	Gábor Szűcs, *szucs@itm.bme.hu*
Edit. Board SNE	András Jávor, *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 December2013

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	Zdzislaw 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.ed.pl
Web EUROSIM	Magdalena Topczewska *m.topczewska@pb.edu.pl*

Last data update December2013

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 December2013

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 December2013

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, *rchc@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 December2013

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 Durrhes, Albania, an IFAC Simulation worksh ops 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 December2013

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 Stanciulescu,
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, *mradulescu@ici.ro*
Repr. EUROSIM	Florin Stanciulescu, *sflorin@ici.ro*
Deputy	Marius Radulescu, *mradulescu@ici.ro*
Edit. Board SNE	Florin Stanciulescu, *sflorin@ici.ro*
Web EUROSIM	Zoe Radulescu, *radulescu@ici.ro*

Last data update December2012

RNSS – Russian Simulation Society

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

→ *www.simulation.su*

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

SNE is the official membership journal of **EUROSIM**, the Fede ration of Europe an Si mulation Societies. A News Section in **SNE** provides inf ormation f or **EUROSIM** Simulation Societies and Sim ulation Groups. In 2013, SNE introduced an extended subm ission strategy i) individual s ubmissions of scientific papers, and ii) submissions of selected c ontributions 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) a nd in a n online ve rsion (Online ISSN 2306-0271). W ith **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 Obj ect 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 i nvited to subm it contributions which have n ot been p ublished a nd ha ve not being considere d for pu blication else where t o the SNE Editorial Office. SNE di stinguishes 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

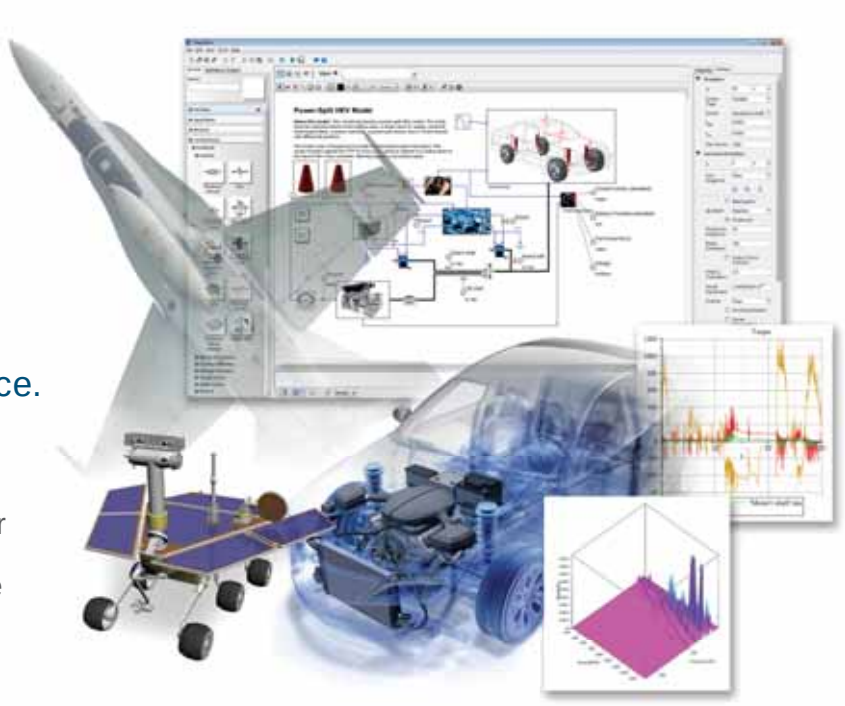
→ *www.sne-journal.org, www.eurosim.info*

office@sne-journal.org (info, news)

eic@sne-journal.org Felix Breitenecker (publications)

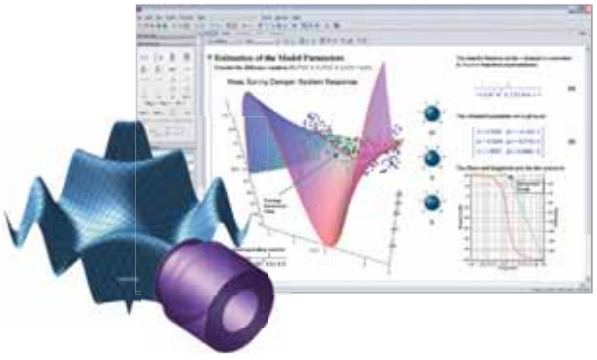
A modern approach to modeling and simulation

MapleSim™



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

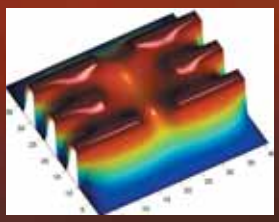
Contact us: +49 (0)241/980919-30

Maplesoft
Mathematics • Modeling • Simulation
A Cybernet Group Company

www.maplesoft.com | germany@maplesoft.com

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

Image: Kim Young-Sang, Jung HeeJun, Quantum Device Lab, Hanyang Univ.

MATLAB®

The language of technical computing