

SNE SIMULATION NOTES EUROPE



Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 33 No.1, March 2023

ISSN Online 2306-0271

DOI 10.11128/sne.33.1.1063

ISSN Print 2305-9974

ARGESIM

SNE - Aims and Scope

Simulation Notes Europe (SNE) provides an international, high-quality forum for presentation of new ideas and approaches in simulation - from modelling to experiment analysis, from implementation to verification, from validation to identification, from numerics to visualisation (www.sne-journal.org).

SNE seeks to serve scientists, researchers, developers and users of the simulation process across a variety of theoretical and applied fields in pursuit of novel ideas in simulation. **SNE** follows the recent developments and trends of modelling and simulation in new and/or joining areas, as complex systems and big data. **SNE** puts special emphasis on the overall view in simulation, and on comparative investigations, as benchmarks and comparisons in methodology and application. For this purpose, **SNE** documents the **ARGESIM Benchmarks** on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions. **SNE** welcomes also contributions in education in/for/with simulation.

SNE is the scientific membership journal of **EUROSIM**, the *Federation of European Simulation Societies and Simulation Groups* (www.eurosim.info), also providing Postconference publication for events of the member societies. **SNE**, primarily an electronic journal **e-SNE** (ISSN 2306-0271), follows an open access strategy, with free download in basic version (B/W, low resolution graphics). Members of most **EUROSIM** societies are entitled to download **e-SNE** in an elaborate full version (colour, high resolution graphics), and to access additional sources of benchmark publications, model sources, etc. (via group login of the society), **print-SNE** (ISSN 2305-9974) is available for specific groups of **EUROSIM** societies.

SNE is published by **ARGESIM** (www.argesim.org) on mandate of **EUROSIM** and **ASIM** (www.asim-gi.org), the German simulation society. **SNE** is DOI indexed with prefix 10.11128.

Author's Info. Individual submissions of scientific papers are welcome, as well as post-conference publications of contributions from conferences of **EUROSIM** societies. **SNE** welcomes special issues, either dedicated to special areas and/or new developments, or on occasion of events as conferences and workshops with special emphasis.

Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the **SNE** Editorial Office.

SNE distinguishes different types of contributions (*Notes*), i.e.

- **TN** Technical Note, 6–10 p.
- **SN** Short Note, max. 5 p.
- **SW** Software Note, 4–6 p.
- **BN** Benchmark Note, 2–10 p.
- **ON** Overview Note – only upon invitation, up to 14 p.
- **EN** Education Note, 6–8 p.
- **PN** Project Note 6–8 p.
- **STN** Student Note, 4-6 p., on supervisor's recommendation
- **EBN** Educational Benchmark Note, 4–10 p.

Further info and templates (doc, tex) at **SNE**'s website, or from the Editor-in-Chief

www.sne-journal.org

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

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 of submission to **SNE** (and extended for special issues and Postconference publication):

Felix Breitenecker, Felix.Breitenecker@tuwien.ac.at
TU Wien, Math. Modelling, Austria, Editor-in-chief
David Al-Dabass, david.al-dabass@ntu.ac.uk,
Nottingham Trent University, UK
Maja Atanasijevic-Kunc, maja.atanasijevic@fe.uni-lj.si
Univ. of Ljubljana, Lab. Modelling & Control, Slovenia
Aleš Belič, ales.belic@sandoz.com
Sandoz / National Inst. f. Chemistry, Slovenia
Peter Breedveld, P.C.Breedveld@el.utwente.nl
University of Twente, Netherlands
Agostino Bruzzone, agostino@itim.unige.it
Universita degli Studi di Genova, Italy
Vlatko Čerić, vceric@efzg.hr, Univ. Zagreb, Croatia
Russell Cheng, rhc@maths.soton.ac.uk
University of Southampton, UK
Roberto Cianci, cianci@dime.unige.it,
Math. Eng. and Simulation, Univ. Genova, Italy
Eric Dahlquist, erik.dahlquist@mdh.se, Mälardalen Univ., Sweden
Umut Durak, umut.durak@dlr.de
German Aerospace Center (DLR) Braunschweig, Germany
Horst Ecker, Horst.Ecker@tuwien.ac.at
TU Wien, Inst. f. Mechanics, Austria
Vadim Engelson, vadime@mathcore.com
MathCore Engineering, Linköping, Sweden
Peter Groumpos, groumpos@ece.upatras.gr
Univ. of Patras, Greece
Edmond Hajrizi, ehajrizi@ubt-uni.net
University for Business and Technology, Pristina, Kosovo
Glenn Jenkins, GLJenkins@cardiffmet.ac.uk
Cardiff Metropolitan Univ., UK
Emilio Jiménez, emilio.jimenez@unirioja.es
University of La Rioja, Spain
Peter Junglas, peter@peter-junglas.de
Univ. PHTW Vechta, Mechatronics, Germany
Esko Juuso, esko.juuso@oulu.fi
Univ. Oulu, Dept. Process/Environmental Eng., Finland
Kaj Juslin, kaj.juslin@enbuscon.com, Enbuscon Ltd, Finland
Andreas Körner, andreas.koerner@tuwien.ac.at
TU Wien, Math. E-Learning Dept., Vienna, Austria
Francesco Longo, f.longo@unical.it
Univ. of Calabria, Mechanical Department, Italy
Yuri Merkurjev, merkur@itl.rtu.lv, Riga Technical Univ.
David Murray-Smith, d.murray-smith@elec.gla.ac.uk
University of Glasgow, Fac. Electrical Engineering, UK
Gasper Music, gasper.music@fe.uni-lj.si
Univ. of Ljubljana, Fac. Electrical Engineering, Slovenia
Thorsten Pawletta, thorsten.pawletta@hs-wismar.de
Univ. Wismar, Dept. Comp. Engineering, Wismar, Germany
Niki Popper, niki.popper@dwh.at, dwh Simulation Services, Austria
Kozeta Sevrani, kozeta.sevrani@unitir.edu.al
Univ. Tirana, Inst.f. Statistics, Albania
Thomas Schriber, schriber@umich.edu
University of Michigan, Business School, USA
Yuri Senichenkov, sneyb@dcn.infos.ru
St. Petersburg Technical University, Russia
Michal Štepanovský, stepami9@fit.cvut.cz
Technical Univ. Prague, Czech Republic
Oliver Ullrich, oliver.ullrich@iais.fraunhofer.de
Fraunhofer IAIS, Germany
Siegfried Wassertheurer, Siegfried.Wassertheurer@ait.ac.at
AIT Austrian Inst. of Technology, Vienna, Austria
Sigrid Wenzel, S.Wenzel@uni-kassel.de
Univ. Kassel, Inst. f. Production Technique, Germany
Grégory Zacharewicz, gregory.zacharewicz@mines-ales.fr
IMT École des Mines d'Alès, France

Editorial

Dear Readers, This first issue of SNE Volume 33, 2023, presents post-conference publications from ASIM's 26th Symposium Simulation (TU Wien, July 2022). A special review board has selected these contributions, and the authors have submitted improved and corrected versions of their contributions, and SNE 33(1) publishes those with emphasis on modelling and simulation in/of digital systems (to be followed by contributions with different emphasis in the next SNE issues). The range of contributions underlines SNE's broad area: coupling of simulation tools, proxel based simulation, system entity structure, Julia library for heat conduction, DES and reinforcement learning, virtual stochastic sensors, and route guidance in a cyber-physical lab. SNE Volume 33 continues another SNE tradition: SNE covers with algorithmic art graphics. Graham Horton, Professor for Simulation and Modelling at University Magdeburg, and ASIM-Member for many years, provides his computer-generated marbled pattern graphics. Graham Horton started modelling and simulating the handcraft of making marbled paper, used for covers in traditional bookbinding. The algorithms in behind manufacture not only approximations of handmade marbled patters, they generate new marbled patterns as type of algorithmic art: Digital Marbling. This March Graham Horton gave a keynote on Digital Marbling at an ASIM workshop in Magdeburg – the abstract of this presentation introduces into this very charming application of modelling and simulation, reprinted at the next page, together with an astonishing relation between SNE and Marbled Paper. I would like to thank all authors for their contributions for this issue, and thanks to members of the special post-conference review board for selecting the contributions; also and many thanks to the SNE Editorial Office for layout, typesetting, preparations for printing, electronic publishing, and much more. And many thanks to Graham Horton for his Digital Marbling Graphics, continuing excellently Vlatco Čerić's, algorithmic artwork on covers of previous SNE volumes. And we are glad that after the series of virtual conferences we can promote again EUROSIM face-to-face conferences: the EUROSIM Congress 2023, July 2023 Amsterdam, the conference ASIM SPL 2023, Ilmenau, in September 2023, I3M September 2023 in Athens, and others (see covers).

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

Contents SNE 33(1)

e-SNE 33(1), DOI 10.11128/sne.33.1.1063

www.sne-journal.org/sne-volumes/volume-33

Basic e-Version with Open Access

Full e-Version for Members of publication-active EUROSIM Societies: ASIM, CSSS, DBSS, KA-SIM, NSSM, SIMS, SLOSIM, UKSIM
print-SNE for members on demand (printer INTU TU Wien)

Coupling of Simulation Tools for Obtaining Local Fatigue in Combination with Experimental Data. <i>N. Adamchuk, B. Rösch, W. Schlüter, F.-C. Reissner, J. Baumgartner</i> ...	1
Adapting to Change of Model Transitions in Proxel Based Simulation of CHnMMs. <i>D. Bodnár, C. Krull</i>	9
An Extension for the Specification and Automated Selection of System Variants Based on the System Entity Structure Using a Problem from Process Industry. <i>H. Folkerts, T. Pawletta, U. Durak</i>	17
Hestia.jl: A Julia Library for Heat Conduction Modeling with Boundary Actuation. <i>S. Scholz, L. Berger</i>	27
Parameter-Free Approximation Method for Controlling Discrete Event Simulation by Reinforcement Learning. <i>D. Pasterk, A. Körner</i>	31
A Hybrid User Model for Virtual Stochastic Sensors. <i>C. Krull</i>	35
Modular Platform for Route Guidance in the Cyber-Physical Laboratory Test Field. <i>M. Göllner, S. Jacobitz, T. Li, X. Liu-Henke</i>	45
EUROSIM Societies & ARGESIM/SNE Short Info . N1 – N5	
Conferences EUROSIM	Back Cover
Cover: Digital Marbling by Graham Horton, type 'Bird Wing' <i>digital-marbling.de</i>	

SNE Contact & Info

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

→ www.sne-journal.org

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

✉ SNE Editorial Office

Johannes Tanzler (Layout, Organisation),
Irmgard Husinsky (Web, Electronic Publishing),
Felix Breiteneker (Organisation, Author Mentoring)
ARGESIM/Math. Modelling & Simulation Group,
Inst. of Analysis and Scientific Computing, TU Wien
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

SNE SIMULATION NOTES EUROPE

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

Scope: Developments and trends in modelling and simulation in various areas and in application and theory; comparative studies and benchmarks (documentation 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 Breiteneker, TU Wien, Math. Modelling Group
✉ Felix.Breiteneker@tuwien.ac.at, ✉ elic@sne-journal.org

Print SNE: INTU (TU Wien), Wiedner Hauptstrasse 8-10,
1040, Vienna, Austria – www.intu.at

ARGESIM Publisher: ARBEITSGEMEINSCHAFT SIMULATION NEWS
c/o Math. Modelling and Simulation Group, TU Wien / 101,
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria;
www.argesim.org, ✉ info@argesim.org
on behalf of ASIM www.asim-gi.org and
EUROSIM → www.eurosim.info

© ARGESIM / EUROSIM / ASIM 2023

Digital Marbling – Simulation of a Traditional Bookbinder's Craft

Graham Horton, Computer Science Department,
Univ. Magdeburg, 39016 Magdeburg, Germany,
graham.horton@ovgu.de

Up until the era of mass production, books were bound manually, often using elaborately decorated paper for the endpapers and covers. In the technique known as marbling, pigments were distributed on the surface of a water bath, and a pattern was induced in them with a needle or a comb. This pattern was then "printed" by carefully laying a sheet of paper onto the surface of the liquid.



Figure 1: Making spirals in the floating pigments using a needle. (Photo: Barbara Kelnhofer)

Using a very simple mathematical model, many traditional marbling patterns can be recreated by simulating the effect of dragging a needle or comb through the water bath. The resulting images can be quite beautiful, especially when printed at high resolution on an A2-sized sheet of paper.

The focus of the investigations are to show various patterns generated by the simulator. These include comparisons with the endpapers of various 19th and 20th century books that show how accurate even the simple mathematical model can be.

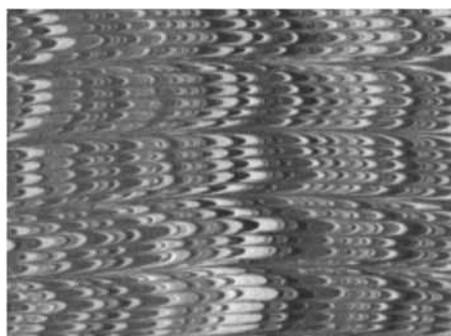


Figure 2: Detail from the endpaper of a 19th-century book. (Photo: Folger Shakespeare Library, used under a CC BY-SA 4.0 license).

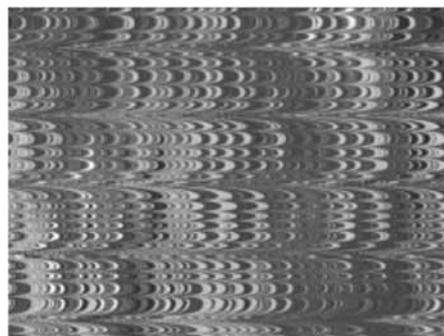


Figure 3: Simulation of the marbled pattern.

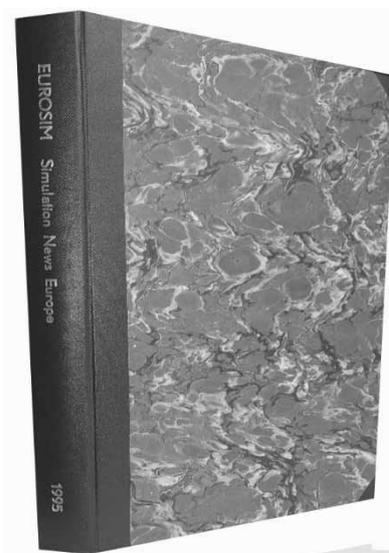
The focus of the investigations are to show various patterns generated by the simulator. These include comparisons with the endpapers of various 19th and 20th century books that show how accurate even the simple mathematical model can be.

Furthermore, original, digitally marbled patterns can be generated, which - although physically feasible - have yet to be created by traditional craftspeople.

Reproduced and modified with the permission of the author from *Digital Marbling: Simulation of a Traditional Bookbinder's Craft*, Graham Horton; in *Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU* (Eds. Claudia Krull; Walter Commerell, Umut Durak, Andreas Körner, Thorsten Pawletta), ISBN ebook: 978-3-903347-62-5, DOI 10.11128/arep.22, ASIM Mitteilung AM 186, ARGESIM Verlag Wien, 2023, p. 5

SNE and Marbled Paper

The SNE Editorial Office found a charming relation between SNE and Marbled Paper. The office is not only taking care of a digital archive for all SNE issues, it also keeps a stock of the printed SNE issues.



And for better preservation, SNE Volumes have been bound as book in traditional style – with marbled paper as cover and with linen spine for the early volumes.

The picture at left shows the marbled paper-bound SNE Volume of 1995 – is it analog or digital marbling ?

Coupling of Simulation Tools for Obtaining Local Fatigue in Combination with Experimental Data

Nazar Adamchuk^{1*}, Bernhard Rösch¹, Wolfgang Schlüter¹, Felix-Christian Reissner²,
Jörg Baumgartner²

¹Competence Center for Industrial Energy Efficiency, Ansbach University of Applied Sciences, Residenzstraße 8, 91522 Ansbach, Germany; *nazar.adamchuk@hs-ansbach.de

²Fraunhofer Institute for Structural Durability and System Reliability LBF, Bartningstrasse 47, 64289 Darmstadt, Germany

SNE 33(1), 2023, 1-8, DOI: 10.11128/sne.33.tn.10631
Selected ASIM SST 2022 Postconf. Publication:2023-02-01;
Received Revised Improved:2023-02-27; Accepted:2023-03-10
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Cast iron components have a good strength to weight ratio. This leads to their frequent use in the wind industry. The design of cast iron components is currently based on the use of individual simulation tools and material data that is common to all components. In order to better exploit the lightweight potential of cast iron components, it is necessary to link the simulation software tools and thus take into account local material properties already in the design phase. This is described in this paper using the example of a large casting for the wind industry.

Introduction

Designers of castings are increasingly using simulation tools. There are typically three steps involved: (1) The designer creates a geometry based on the requirements and calculates the stresses resulting from the applied static operating loads and inertia forces using FEM simulation with structural analysis. In an optimisation loop, the component can be optimised to minimise the stresses under certain constraints. (2) The casting process simulation is performed based on the 3D CAD model. The entire casting process from pouring to solidification of the melt is analysed in the form of a CFD simulation and the casting system (feeder, material allowance, etc.) is defined. (3) Based on the CAD data and, if necessary, the local material properties, the fatigue life is assessed and potential local weak spots in the component are identified. A number of sub-steps are required before a final design is achieved.

The use of simulation programmes for casting pro-

cess, structural and fatigue analysis is currently still carried out independently. If several calculation tools are used, there is no exchange of all relevant data, partly due to the lack of software interfaces. There is no holistic view of the simulation data.

Cast iron components are often subjected to high mechanical loads. To ensure the safe operation of plant and machinery, the fatigue life of the component must be guaranteed under the assumed operating conditions over the planned service life. New or expanded applications and increased safety requirements increase the need for detailed life prediction. From this point of view, the following question arises: How can casting process simulation and structural analysis contribute to the fatigue life calculation?

In addition to simulation-based fatigue analysis, the fatigue life of a component can also be determined experimentally, but these tests are time consuming and expensive. They also require extensive laboratory equipment. In addition, component tests of this magnitude are almost impossible to perform experimentally. Therefore, this route is only taken in a few cases, and experimental validation of simulation results is also too costly in most cases. The present work is the first attempt to take into account the experimental data obtained at great expense in simulation-based fatigue analysis. These data have been obtained from samples of large castings.

1 Simulative Determination of Local Fatigue Limits

In the following chapter, two different approaches are presented from a methodological point of view. The first approach describes the calculation of fatigue limits based on a homogeneous material definition, i.e. the material definition is identical for the whole component.

Locally varying material parameters are not taken into account. The second approach describes the integration of casting process simulation data into the fatigue analysis. Here the local material properties are defined on a common FE mesh. Both approaches have already been mentioned in [1, 2, 3]. This paper describes in detail the technical realisation of the data using an example.

1.1 Procedure with Global Material Data

FEMFAT is the world's leading solver for FE-based fatigue analysis and calculates the structural durability of statically and dynamically loaded components based on the results of FE calculations. As an FE post-processor, FEMFAT requires not only the structural analysis data (FE mesh and loads) but also the material data (strength values). The figure 1 shows the typical procedure for calculating local fatigue limits in FEMFAT.

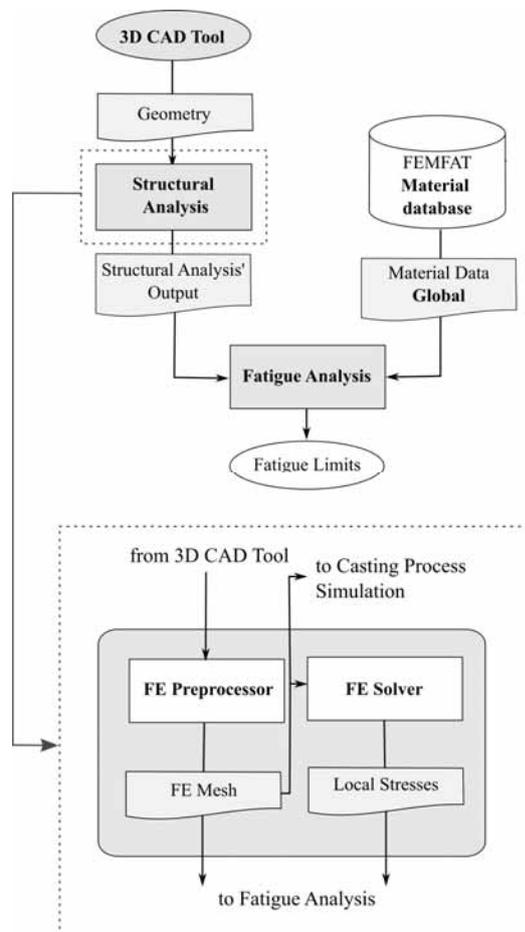


Figure 1: Calculation of fatigue limits with global material data.

The following example of a large wind turbine casting illustrates the process in more detail. The *poll end* or *canister* (the cast iron box on the end of the wind shaft through which the sails stocks pass) of a wind turbine has to withstand high cyclic loads. It is therefore extremely important to ensure fatigue limits in areas of high stress.

The determination of fatigue limits starts with the import of the component geometry into the structural analysis pre-processor to generate an FE mesh. This step has been performed in VisPER, a component of the structural analysis tool PERMAS 18.00.404.

Simulations to calculate the static stresses were carried out using the PERMAS FE solver. For the poll end, the resulting stresses for the different rotor positions were investigated for a complete poll end revolution at a distance of 45°. For the resulting 8 positions, the mechanical stresses were determined with PERMAS (Figure 2).

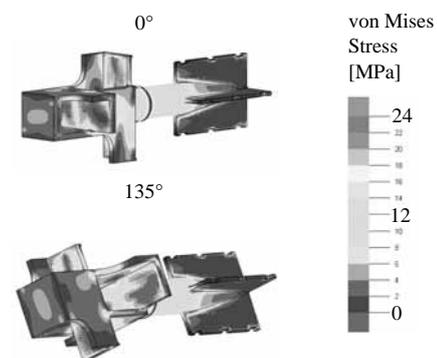


Figure 2: Simulated distribution of mechanical stresses at two different angular positions.

The TransMAX module from FEMFAT was used to calculate the locally endurable stresses. This provides the user with the ability to analyse structural durability based on load-time histories. Prior to this, a material had to be defined for the fatigue analysis model. Typically, the option of a homogeneous material definition from the internal FEMFAT database is used.

A material class is selected from the material database. Based on pre-programmed ratios, the following missing material parameters are added to a predefined tensile strength for the calculation to be automatically generated:

- Young's modulus
- Yield strength
- Elongation at break

At the FE node, local fatigue limits are calculated from material parameters that are additionally influenced by local component properties (e.g. notch effect) and loads. The basic procedure for calculating fatigue limits is based on the influencing factors that increase or decrease the fatigue strength. The FKM (Forschungskuratorium Maschinenbau) standard describes these factors [4].

1.2 Integration of a Casting Process Simulation

The casting process simulation determines the local microstructure formed by the manufacturing process. A microstructure is formed as the metal solidifies and consists of different microstructural phases with different shape, size and distribution (grains, dendrites, lamellae, pores) [5]. Figure 3 illustrates the distribution of a microstructural phase - pearlite - at the poll end.

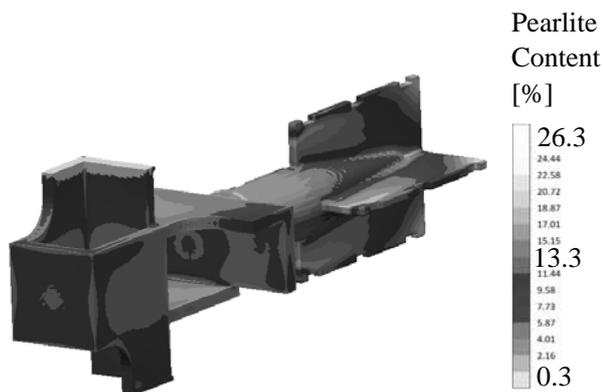


Figure 3: Simulated microstructure fraction in the component at room temperature.

Given the demands on the accuracy of today’s simulations in the foundry industry, local material differences must be taken into account in the fatigue analysis. One way of dealing with this is to use the local material data from the casting process simulation. From the microstructure, the casting process simulation can determine local material values (tensile strength, Young’s modulus, yield strength, elongation at break) in the casting in the next step (Figure 4).

Figure 5 shows the schematic description of the approach to calculate the local fatigue limits with integration of the local material data from the casting process simulation:

For the calculation of the local material data of the poll

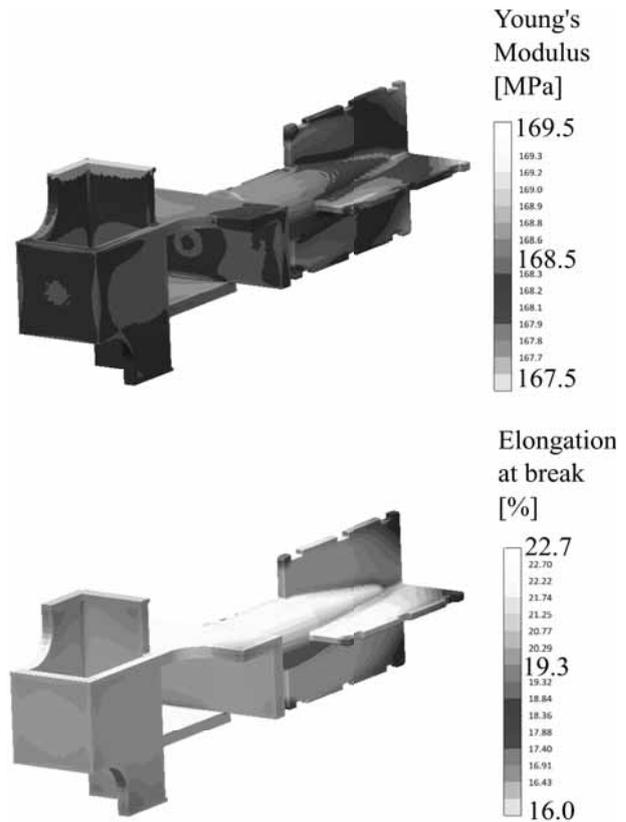


Figure 4: Simulated local material properties in the component at room temperature.

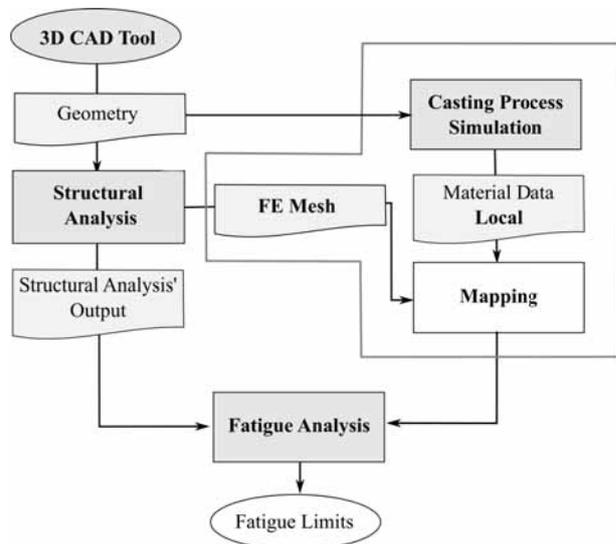


Figure 5: Calculation of fatigue limits with local material data from casting process simulation.

end, the casting process simulation was carried out in the commercial software package MAGMASOFT®5.4.

The discretisation for the numerical solution algorithm in the casting process simulation on the one hand and in the fatigue/structural analysis on the other hand is different: in MAGMASOFT® the discretisation is based on the Finite Volume Method (FVM) and in the fatigue/structural analysis on the Finite Element Method (FEM). There is a need to map the results from the casting process simulation to the FE mesh generated in the structural analysis. The mapping is realised by MAGMALink, a casting process simulation module. This makes the results of the casting process simulation available for further processing in the fatigue analysis.

According to the state of the art, FEMFAT can read in the local material data from the casting process simulation and use the output from the structural analysis (FE mesh and local loads) to calculate the fatigue limits for each FE node. Comparing the two approaches (Figure 6), it can be seen that when the casting process simulation is included, the fatigue limits are on average between 10 % and 20 % higher. The difference at the edge of the poll end is significantly larger, up to 50 %.

2 Consideration of Experimental Data.

The described simulative approaches start from the component geometry and the determination of the fatigue limits is computer-aided. However, the fatigue life analysis tool FEMFAT also offers the possibility of directly importing already existing local fatigue data in order to perform a more specific calculation.

It has already been shown in several publications that the microstructure has an influence on the fatigue life [6, 7, 8, 9]. In particular, the ratio of pearlite to ferrite and nodularity have been shown to be important microstructural parameters. In this context, a high pearlite content and a high nodularity have a positive effect on the endurable stresses and thus on the component fatigue. However, nodularity is not calculated by the casting process simulation and is therefore consequently not considered further.

The casting process simulation does not require the local mechanical properties as in the purely numerical approach, but the local microstructure.

If experimental data on the microstructure and the associated fatigue limit are now available in a material database, the microstructure resulting from the casting

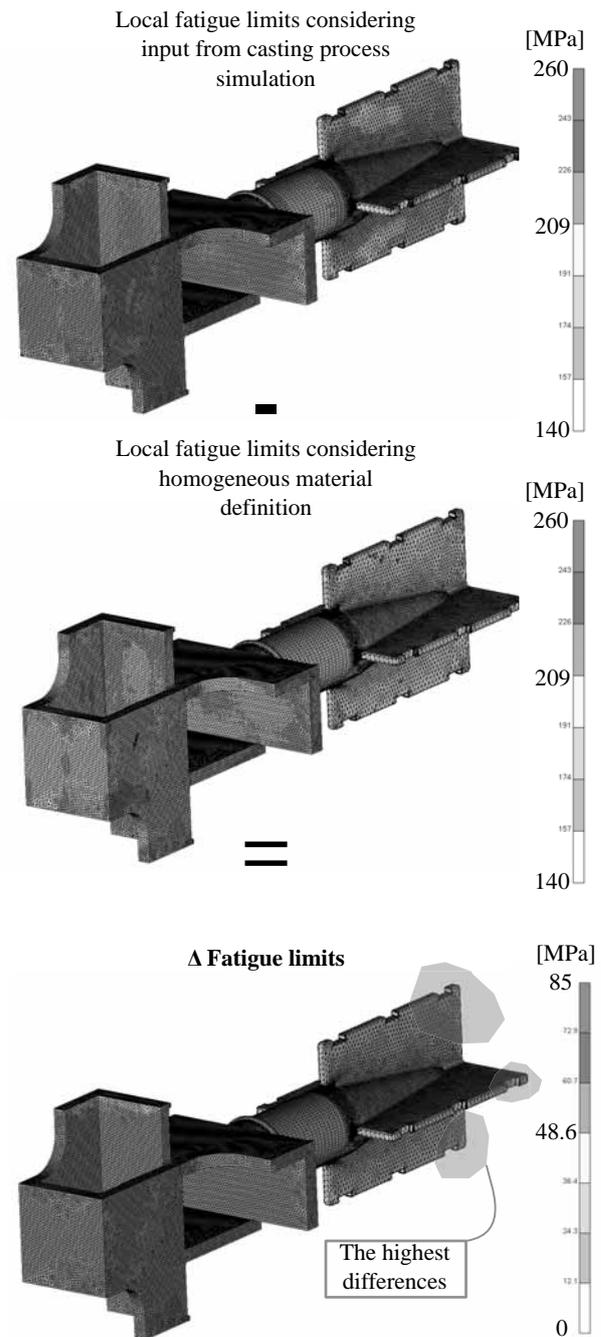


Figure 6: Fatigue limits with homogeneous material data and incorporating casting process simulation data

process simulation can be used to generate local fatigue limits (Figure 7).

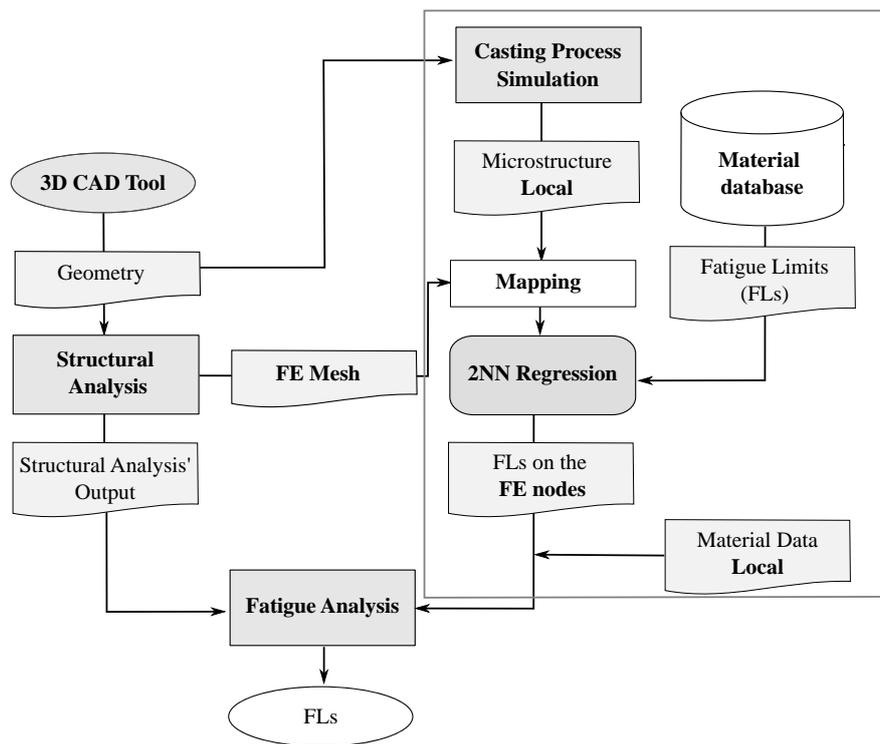


Figure 7: Calculation of local fatigue limits using experimental data.

Since the experimental determination of microstructure/fatigue limits data is very costly, only a few data sets are available. In order to determine the associated fatigue limits for all the microstructure data from the casting process simulation, these must be approximated from the available experimental data.

Using the k-nearest neighbour algorithm (kNN) as a regression method [10], an individual fatigue limit can be generated for each FE node (Figure 8). Here, the value at a FE node is weighted by the distances d of the 2 nearest neighbours proportional to their distance. In the present work, the regression method is implemented in the form of in-house developed MATLAB code.

The fatigue limit $FL(x)$ for a node x is obtained from the experimentally determined fatigue limits of the two nearest neighbours x_1 and x_2 (both calculated node-wise from the casting process simulation) according to the equation 1:

$$FL(x) = \frac{d(x, x_1) \cdot FL(x_1) + d(x, x_2) \cdot FL(x_2)}{d(x, x_1) + d(x, x_2)} \quad (1)$$

In the equation 1 the $d(x, x_1)$ and $d(x, x_2)$ represent the distances from the FE node to the two nearest neighbours.

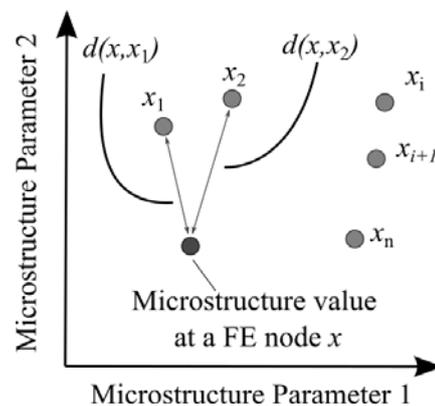


Figure 8: 2NN regression of fatigue limits from microstructure values.

bours (experimental data) in two-dimensional space consisting of two positive real numbers, the values for microstructure parameter 1 (x -axis) and microstructure parameter 2 (y -axis).

Finally, the fatigue limits obtained are modified by FEMFAT's internal algorithm using the stresses from the structural analysis. This allows a comparison of the fatigue limits from the three approaches described.

Compared to the calculation using the local material properties from the casting simulation, the fatigue limits with the regression from the microstructure data increase again by between 10 % and 20 %. Figure 9 shows the areas with the highest differences between the approach based on experimental data and the approach based on the local material properties. The differences are located at the blade bearings and the contact vials to the shaft.

Conclusion and Outlook

In this article, three possibilities for the simulative determination of the fatigue limit have been presented on the basis of a large cast component used in wind turbine technology. The integration of experimental data represents a new way of combining information from simulation and experiment. It is shown that the determined fatigue limits are lower with the purely numerical approaches than with the consideration of experimental data, which supports the application of purely numerically determined fatigue limits in industrial practice.

The complexity of the study for the numerical determination of local fatigue limits is explained by the purposeful coupling of software tools in the foundry industry which have not been connected up to now. In order to combine the different software modules, a number of interfaces are required. Table 1 shows the formats of the respective interfaces used in the three fatigue limit calculation approaches.

The integration of experimental data into the numerical process chain could be realized by a 2NN regression. Since the acquisition of experimental data on fatigue limits is very time-consuming, the algorithm is currently based on very little data. The incorporation of more experimental data will improve the prediction accuracy. These should also cover significantly more microstructure classes with the associated fatigue limits. An evaluation of the method presented here is only possible if the fatigue limit of a component is determined both experimentally and numerically.

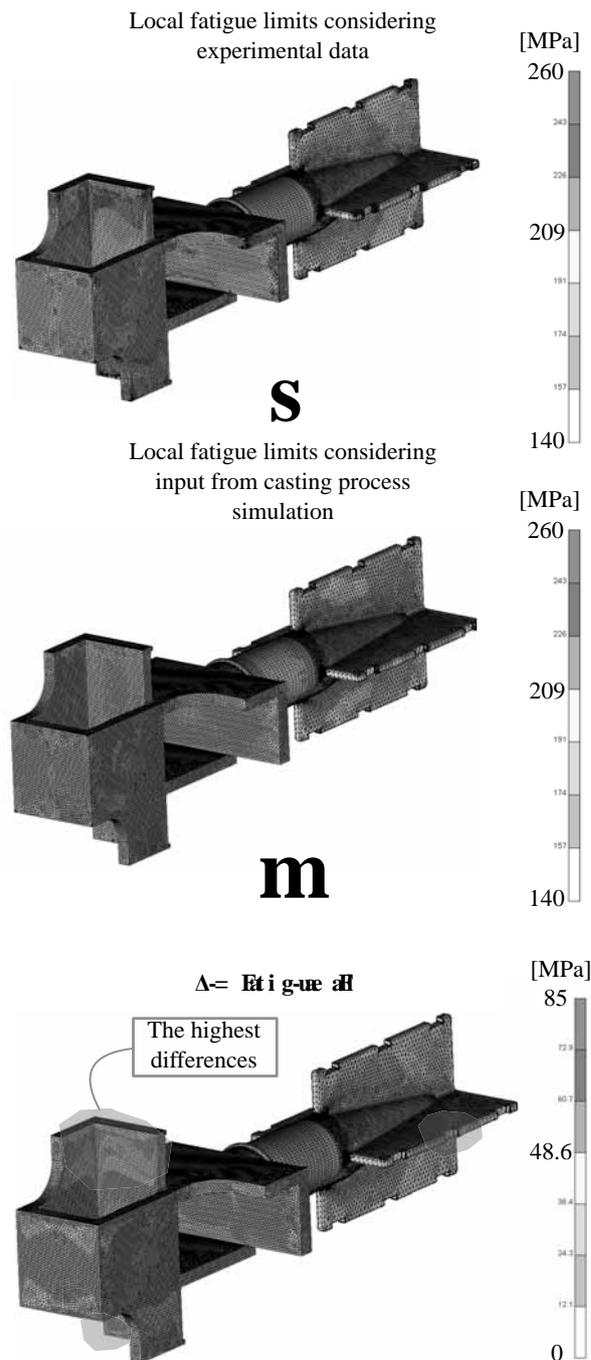


Figure 9: Fatigue limits considering data from casting process simulation and considering experimental data.

#	Interfaces between software tools	Transmitted data	Format (file extension)
1.	3-D CAD tool → Structural analysis	Geometry	- (STL)
2.	Structural analysis → fatigue analysis	FE stresses	PERMAS (POST)
3.	Structural analysis → fatigue analysis	FE mesh	PERMAS (DAT)
4.	Structural analysis → casting process simulation (MAG-MAlink)	FE mesh	PERMAS (DAT)
5.	Casting process simulation → fatigue analysis	Local material data	- (UNV)
6.	Casting process simulation → regression	Local microstructure	PERMAS (DAT)
7.	Material DB → regression	Microstructure-dependent fatigue limits	- (XLSX)
8.	Regression → fatigue analysis	Local fatigue limits and material data	- (UNV)

Table 1: Overview of the used file formats.

Acknowledgement

The authors would like to thank the German Federal Ministry for Economic Affairs and Climate Action (BMWK), which financially supported the research activities within the DNAguss project by a resolution of the German Bundestag (03EE3018B).

References

- [1] Chen X, Engler-Pinto C, King M, Li M, Prabhu E, Su X. Aluminum Cylinder Head High Cycle Fatigue Durability Including the Effects of Manufacturing Processes. 2012; pp. 2012-01-0540. URL <https://www.sae.org/content/2012-01-0540/>
- [2] Gaier C, Eberhard D, Aichberger W, Dannbauer H. Betriebsfestigkeitsanalyse mit Berücksichtigung von Ergebnissen aus der Prozesssimulation. Wiesbaden, Germany; .
- [3] Puchner K, Aigner R, Rinnerschwendner L, Dannbauer H. Consideration of local material characteristics for the fatigue assessment of cast component. In: *Proceedings of the 1st Congress for intelligent Combining of Design, Casting, Computer Simulation, Checking and Cyclic Behaviour for efficient Cast Components*. Darmstadt; pp. 155-164.
- [4] Rennert R. *Rechnerischer Festigkeitsnachweis für Maschinenbauteile aus Stahl, Eisenguss- und Aluminiumwerkstoffen*. FKM-Richtlinie. Frankfurt am Main: VDMA-Verl, 6th ed. 2012.
- [5] Mercier JP, Zambelli G, Kurz W. Microstructures. In: *Introduction to Materials Science*, pp. 239-259. Elsevier. 2002;. URL <https://linkinghub.elsevier.com/retrieve/pii/C20090291483>
- [6] Olofsson J. Local microstructure-based material performance and damage in design and finite element simulations of cast components. *Journal of Computational Design and Engineering*. 2018; 5(4):419-426. URL <https://academic.oup.com/jcde/article/5/4/419/5728982>
- [7] Thomser C, Bodenburg M, Sturm JC. Optimized Durability Prediction of Cast Iron Based on Local Microstructure. *International Journal of Metalcasting*. 2017;11(2):207-215. URL <http://link.springer.com/10.1007/s40962-016-0091-x>
- [8] Systemzuverlässigkeit FIFBU. Maßgeschneiderte Bauteileigenschaften durch Integration von Fertigungs- und Funktionssimulation : BMBF-Verbundprojekt MABIFF ; Abschlussbericht des Verbundprojektes.

Tech. rep., Technische Informationsbibliothek u. Universitätsbibliothek. 2011. Artwork Size: Online Ressource (103 S., 8,63 MB) Medium: application/pdf Version Number: 1.0.

URL

<https://www.tib.eu/suchen/id/TIBKAT:68815087X/>

- [9] Hack M, Jung D, Egner-Walter A. Optimierter Lebensdauer-Berechnungsprozess von Eisengussbauteilen unter Berücksichtigung des lokalen Gefüges*. *Materials Testing*. 2012;54(7-8):497–502.
URL <https://www.degruyter.com/document/doi/10.3139/120.110356/html>
- [10] Zhang Z, Gu GX. Finite-Element-Based Deep-Learning Model for Deformation Behavior of Digital Materials. *Advanced Theory and Simulations*. 2020;3(7):2000031.
URL <https://onlinelibrary.wiley.com/doi/10.1002/adts.202000031>

Adapting to Change of Model Transitions in Proxel Based Simulation of CHnMMs

Dávid Bodnár*, Claudia Krull

Institut für Simulation und Graphik, Otto-von-Guericke-Universität Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany; *david.bodnar.ovgu@gmail.com

SNE 33(1), 2023, 9-16, DOI: 10.11128/sne.33.tn.10632
 Selected ASIM SST 2022 Postconf. Publication: 2023-02-01;
 Received Revised Improved: 2023-02-27; Accepted: 2023-03-10
 ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Virtual Stochastic Sensors (VSSs) [1] aim to provide insight into stochastic processes by producing statistically relevant estimates of non-measurable system properties. During behavior reconstruction of these discrete stochastic systems the internal system state changes are often described as time-homogeneous distribution functions, as in Conversive Hidden non-Markovian Models (CHnMMs). However, the system behavior might change over time or the sample, used for the model creation, might not describe the system accurately. In [2] and [3] we have shown that detecting these changes is possible, yet the resource consumption for the re-estimation of the model was a clear problem. In this paper we present a solution to that problem by replacing the used statistical tests with Kernel Density Estimation (KDE) and by integrating the hidden model description into the proxel-based state space simulation method. By using the Change Adaptation Algorithm (CAA) this paper shows that adapting to runtime changes is possible, while preserving parameters on transitions where no change occurs. The algorithm was tested with 5 different types of Probability Density Functions (PDFs) which showed accurate results. By using the CAA one is able to construct adaptive models for behavior reconstruction without the need to fully parametrize the model. In this way, loss of modeling accuracy in the model construction process can be significantly decreased.

Introduction

VSSs were introduced in [4]. They are tools to reconstruct the behavior of partially observable processes in discrete stochastic systems. Constructing VSSs relies

in practice heavily on manually provided knowledge about the system. But what happens if that information becomes outdated or inaccurate during the model construction or over time? What happens if there are flaws in the model construction of the VSS?

To overcome this limitation this paper introduces the CAA, which describes how a proxel-based analysis of CHnMMs can be extended to tune the stochastic parameters of the system model during runtime. The implementation utilizes KDE to re-estimate the state change distributions in every time step based on historical runtime information. In this way potentially different system models are available in a given time step to provide the most accurate model and trace estimation at the end of the behavior reconstruction.

The paper is a proof of concept to analyze whether change adaptation in this way is possible to provide a more realistic coupling between simulation models and the real world and whether such adaptation can be maintained during the VSS's lifetime.

1 Related Work

Measuring information in complex systems has often physical or financial limitations, which might be resolved using Virtual Sensors (VSs) [5]. For stochastic systems, by combining VSs with stochastic processes, a so-called VSS can be constructed to measure statistically relevant estimates of non-measurable system parameters. One of these possible stochastic processes is called CHnMM [6], which can be analyzed by the proxel-based analysis method.

In this section, a brief overview will be given of the previous work on VSSs and KDE with which we are extending the concept to make change adaptation possible. Additionally, the energy distance will be introduced in a few words, hence this was used during the model evaluation.

Conversive Hidden non-Markovian Model

The Hidden Markov Model (HMM) [7] is a well researched technique to analyze directly not observable models through probabilistic symbol emissions. The HMMs assume memoryless model state changes, which is why the concept of Hidden non-Markovian Model (HnMM) was introduced in [8] to extend the HMMs with state changes governed by arbitrary continuous distribution functions. Using HnMMs one is able to define time dependence between different system states.

[6] introduced CHnMMs as a subclass of HnMMs where all state changes of the hidden process of interest emit a symbol for the observer, making additional performance optimizations possible. In this paper, the implementation of the CAA was restricted to CHnMMs to make the proof of concept analysis easier to interpret and to reduce the number of interference factors.

Both CHnMMs and HnMMs try to solve, similarly to the HMMs, the so-called evaluation (finding the probability that a given trace was generated by the model) and decoding tasks (finding the most likely generator state sequence). However, instead of using the Baum-Welch algorithm [7] or the Viterbi algorithm [7] the so-called proxel-based analysis is used which is briefly introduced in the next paragraph.

Proxels-Based Analysis To reconstruct the state space of stochastic processes it is possible to use the so-called proxel-based analysis [9] introduced in [10]. Using this technique one is able to construct container like „probability elements” (proxels) which store all relevant information of a defined discrete simulation state. These are, as shown in Equation 1, the current system state (m), the transition age vector (τ), the probability of the current state (p) and the current timestamp (t). But of course, it can be further extended, for example, with the generator path.

$$P_x = (m, \tau, p, t) \quad (1)$$

The analysis uses discrete timesteps to track the state changes during the process. The connection between a parent proxel and its children in the next time step is characterized by the Hazard Rate Function (HRF) in Equation 2 which describes the rate of probability that a specific state change will happen in

the next timestep if it has not happened yet.

$$H(\tau) = \frac{f(\tau)}{1 - F(\tau)} \quad (2)$$

The network of parent and child proxels in the simulation domain construct the so-called proxel tree, which tracks all possible system states in discrete time steps. To prevent state-space-explosion and to provide acceptable simulation times, impossible or very unlikely proxels are pruned.

Kernel Density Estimation KDE was introduced in [11] and [12]. The idea behind the KDE is to place small kernels $K()$ on samples X_i in the domain and use their aggregated sum as a PDF, as written here:

$$\hat{f}(x, b) = \frac{1}{nb} \sum_{i=1}^n K\left(\frac{x - X_i}{b}\right) \quad (3)$$

Where n is the number of elements in the sample, and b is the so-called bandwidth parameter, which is a free smoothing parameter to „stretch” the kernels to a possibly optimal PDF. The bandwidth selection is a well researched topic on its own. [13] gives a detailed overview about the different techniques and their limitations.

There are multiple techniques to improve the accuracy of the estimated PDF. One of them is using variable KDE, discussed in [13] and [14], where every single kernel gets its own bandwidth through the weights w_i . In this way, one is able to create „spikes” in the PDF where the underlying samples are more dense and still preserve smooth tails and junctions, which is challenging with real world data and a constant bandwidth.

$$\hat{f}(x, b) = \frac{1}{nb} \sum_{i=1}^n \frac{1}{w_i} K\left(\frac{x - X_i}{bw_i}\right) \quad (4)$$

As the CHnMMs are working in the time domain, boundary correction [15] can be used to restrict the KDE computation to positive numbers.

KDE has the advantage over regular PDF definitions that one is not bounded to a defined class of PDFs, because a wide variety can be approximated with KDE as long as a suitable number of samples (n) is available and one is able to choose an appropriate bandwidth (b).

Energy distance The energy distance [16] (Equation 5) describes a statistical distance based on New-

ton's potential energy between two independent random samples (X and Y) described by the Cumulative Distribution Functions (CDFs) F and G . The resulting distance is 0 if and only if $F = G$.

$$D^2(F, G) = 2\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\| \quad (5)$$

Compared to the popular statistical tests like the Wald-Wolfowitz Runs Test, the Kolmogorov-Smirnov Test [17], etc. [18] the energy distance is superior in quantifying differences [19] and it is also closely related to the Cramér von Mises distance [20], which is often used to quantify the difference between distribution functions.

In order to interpret the energy distance, it needs to be normalized as written in [21] (Equation 6). The expression D_n is bounded between zero and one and it is equal to zero if and only if the samples X and Y have the same distribution.

$$D_n = \frac{D^2(F, G)}{2\mathbb{E}\|X - Y\|} \quad (6)$$

This metric can be used to quantify deviation between two distribution functions so in this paper it was used for describing the deviation of the model transition from the ground truth in the evaluation section.

2 Change Adaptation Algorithm

The CAA utilizes the KDE ideas described in Section 1 to construct the transition distributions in every time step. This extension has the additional positive effect that the state transitions are not directly linked to any elementary distribution function and its preliminary properties.

For the KDE computation a given amount of historic samples are used, which are stored for every transition separately using sliding windows. The basic idea is to couple the proxels using these samples with the model definition, in a way that every single proxel encapsulates its own model representation of the CHnMM. As a kernel the standard normal distribution was selected, which has an infinite support, resulting in theory in KDEs accepting new samples from the whole domain, even outside of the current PDF with a very small but non-zero probability.

By using KDE, the CAA extends the content of a single time step. Before, the time step t_i contained n probable system states from which the most probable

one(s) will survive. With CAA the number of proxels in a given time step represent $m \leq n$ different models with n different system states from which the most probable one(s) will survive.

Upon a model drift on a specific transition, the new samples will be automatically assigned to the history of the most probable transition as a result of Equation 2. By doing so, the new samples push out the old model samples from the history with every new time step and this automatically results in a reconfiguration of the CHnMM model through the KDE. As a result, the model is able to adjust itself to model changes with a given delay defined by the length of the history vector and the mean of the new samples.

Implementation To implement the CAA the transition age vector τ gets replaced by a transition vector \mathbf{T} which holds the age τ and the history vector \mathbf{h}_n for every single transition in the CHnMM. The history vector holds the last n firing time samples for the transition T_i .

$$P_x = (m, \mathbf{T}(\tau, \mathbf{h}_n), p, t) \quad (7)$$

The basic mechanisms of the CAA are shown on a flowchart in Figure 1. The values of the history vectors are used as a basis for the KDE. As a bandwidth selector a univariate direct plug-in selector, described in [13], was used.

In a given timestamp t_i , a KDE is computed for every active transition of every proxel. The state change probabilities are computed based on these KDEs and the algorithm generates all child proxels based on the possible state changes for the time step t_{i+1} . The history vector of all the fired transitions gets altered based on the current timestamp and age criteria. At the end, very unprobable child proxels are pruned from the proxel tree before the next time step begins.

The CAA needs an initial model injected upon start. To do that, every transition of the first proxel at t_0 is preseeded with some kind of history. We used as preseed n quantile samples with a n^{-1} step regular grid from an assumed CDF. The samples were reordered according to the PDF so that the most probable sample gets removed first during the sliding window approach.

KDE is a resource intensive computation. We used a look-up-table to speed up the algorithm with a residual error in the PDF of 10^{-5} . This error does not cause a significant difference in the end results unlike other computations in the background, like the Kingsbury-Rayner formula [7], that have a much higher computa-

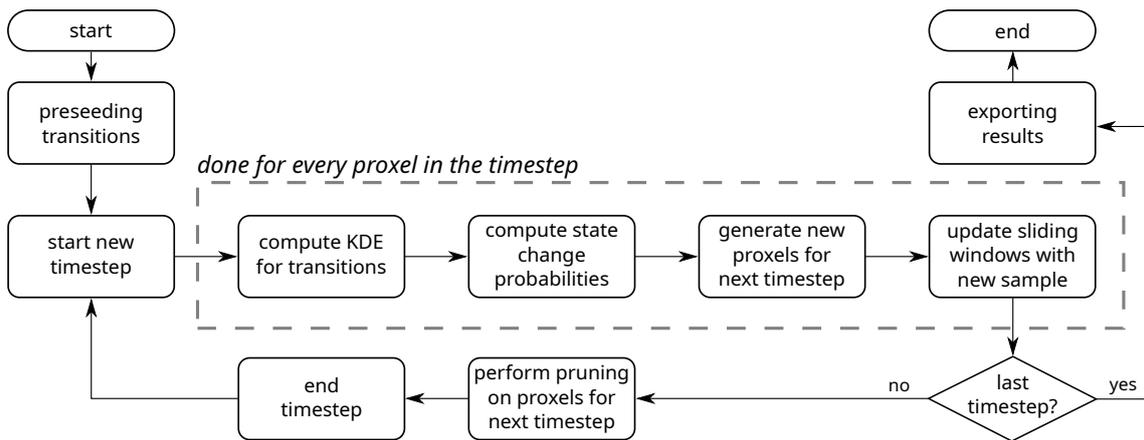


Figure 1: Flowchart of the Change Adaptation Algorithm

tional error. This discretization has the negative drawback that very strong changes in the model result in 0 probability instead of a very small one which can result in a died out proxel tree. This means, that in practice the algorithm will need a roughly usable preseed at the beginning.

Parameters There are two major parameters impacting the results. One of them is the size of the history vector represented by the window size. The other one is the amount of improbable proxels getting removed from the proxel tree in every time step controller by the pruning.

Window size The window size, so the length of the history vector, defines the memory of a given transition. By choosing higher values the algorithm gets more robust against outliers and KDE becomes more accurate. However, this also results in more computational complexity and the adaptation to a new change gets slower as new samples need more time to push out the old ones from the window. Choosing a too small window size can result in a faster computation, but the model will be affected by any interference on the input side of the simulation.

Pruning There are two major strategies for pruning away unlikely states. One of them is keeping only a given number of most-likely proxels, which might result in more inaccurate models. The other one is defining a so-called pruning threshold which will delete proxels if their probability gets below a given value.

In this paper we restricted ourselves to the second approach.

$$p(P_{pruned,t_i}) < r \max(p(P_{x,t_i})) \quad (8)$$

The pruning threshold r , as it can be seen in the Equation 8, is defined by a fixed probability ratio of the most probable proxel in the time step and the proxels which are considered too unlikely. Choosing a high value results in a proxel tree without any real diversity in a single time step. In most of these cases a single model becomes prevalent and if that becomes impossible, the tree dies out. However, if one chooses a too low value then the state space explodes and the algorithm will run too slow (as long as enough RAM is available).

$$r = \begin{cases} r_{min} & \text{if } \#P_{x,t_i} < \#P_{min} \\ r(\#P_{x,t_i}) & \text{if } \#P_{min} \leq \#P_{x,t_i} \leq \#P_{max} \\ r_{max} & \text{if } \#P_{x,t_i} > \#P_{max} \end{cases} \quad (9)$$

The concept of variable pruning threshold was introduced, as fixed pruning thresholds did not fulfill all the requirements of the CAA. The basic idea is (Equation 9) to define a mathematical function between a fixed minimum and maximum pruning threshold based on the number of the proxels in the given time step and use that to define a dynamic transition between those values. This makes it possible to significantly reduce the execution time and to prevent state space explosion while maintaining accurate results and model estimation.

3 Experiments

Experiment setup To validate the CAA, a basic academic example was used, originally introduced in [6]. In this model two production lines get merged before a common quality tester. Both lines produce faulty products with a given probability.

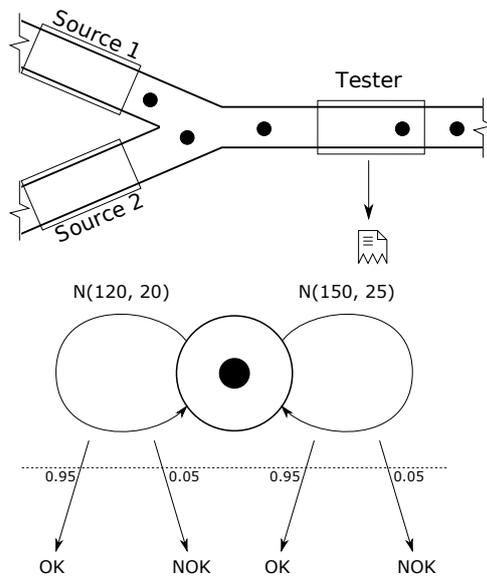


Figure 2: Quality tester example [6] and its ASPN [1]

The system can be described with the ASPN [1] in Figure 2. The symbol emission probabilities were chosen to be equal on both transitions in order to eliminate possible information gain through the asymmetric probabilities.

To validate the CAA, different PDFs were assigned to the transitions describing the production line sources. In a single experiment, Source 1 and Source 2 had always the same distribution type as ground truth to simplify the comparison of the results and to be able to neglect errors introduced by the KDE, as the error appears in both distribution estimations in a similar way. Figure 2 shows the original model parametrization, which was used to preseed the algorithm upon execution start. If not normal distribution was used, the distribution of the specific type under analysis had the same location and scale parameters as mentioned in the figure.

The input data for the experiment was generated in a way that Source 2 received ground truth information, while for Source 1 samples from a changed distribution were generated. For easier comparison of the results,

Scipy's [22] location-scale(-shape) parametrization was used. Please, be aware that this parametrization differs from the standard academic notation. Five distribution function types were analyzed: normal, uniform, exponential, lognormal and Weibull distribution. The change consisted of location parameters on a grid of [60, 150] with a step size of 10, while the scale parameter was tested between [5, 30] with a step size of 5. For PDF types lognormal and Weibull an additional shape parameter was used to test a wider range of possible changes. For every parameter combinations, 20 random experiments were executed.

Parameter selection To analyze the impact of the variable parameters on the experiment results, the parameters window size and pruning threshold were analyzed in a way that the parameter under test was changed on a regular grid while the other parameter was held constant. The results can be seen in Figures 3 and 4.

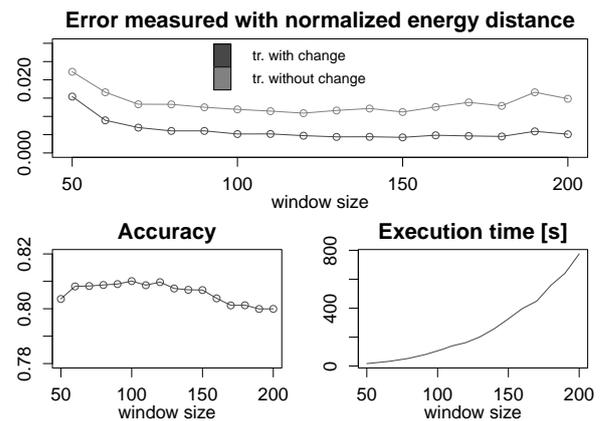


Figure 3: Impact of window size. The error is measured between the ground truth and the model distributions reached at t_{end} . The accuracy plot shows the rate of correctly classified symbols.

In Figure 3 one can see that by increasing window size the execution time increases drastically, as the KDE computation has $O(n^2)$ [13] computational complexity. However, we can also see that choosing a greater window size does not automatically lead to better performance as described in Section 2. Similarly, if we choose the window size too small then the outliers have a negative impact on the results. The optimal window size lies in this case between 120 – 130 so we chose 125 for the experiments.

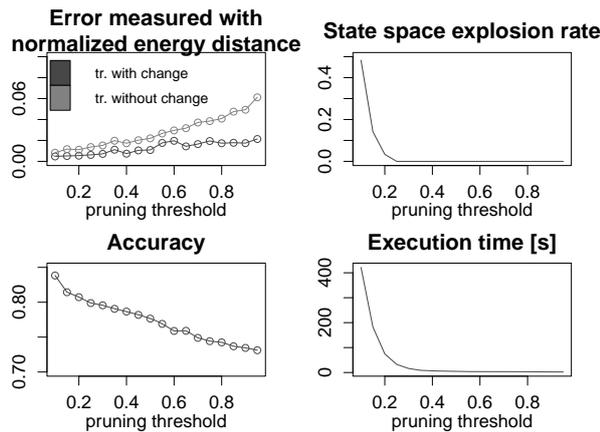


Figure 4: Impact of pruning threshold. The error is measured between the ground truth and the model distributions reached at t_{end} . The plot with the state space explosion shows the rate of failed experiment executions due to out of RAM. The accuracy plot shows the rate of correctly classified symbols.

In Figure 4 one can see that with increasing pruning threshold the model and the symbol classification become more inaccurate. This is a direct impact of reduced diversity in a time step. The optimal pruning threshold can be found around the value 0.2, however, there is a significant risk of experiencing state space explosion.

$$r(\#P_{x,t_i}) = 0.15(\log(\#P_{x,t_i}) - 3)^2 + 0.1 \quad (10)$$

To overcome these limitations variable pruning was introduced between $n = [10^3, 10^5]$ and $r = [0.1, 0.7]$ with Equation 10. This made it possible to reach the same accuracy results as with the static threshold 0.2 and prevent state space explosions accurately on the cost of $\approx 40\%$ increase in mean execution time.

Experiment results Table 1 shows how well the symbols were classified. The precision and recall values were also included as there is a significant difference between the number of symbol emissions on the different transitions. Generally, it can be said, that the classification performs quite well, however, the transition with change has a higher true positive rate ($\geq 80\%$ compared to $\geq 75\%$). This is due to the higher emission rate on the transition 1.

Surprisingly, even in case of the exponential distribution, we get accurate results. Without the CAA, as written in [2] and [3], concurrent exponential transitions were always a problem for the proxel-based simulations as the HRF is in this case constant and all symbol emissions get assigned to the transition with the higher HRF. KDE cannot fully reproduce the exponential distribution in the background, in this way, by violating the theory of the exponential distribution we get practically usable results.

Table 2 shows the model errors at the end of the experiment. There is an expected amount of faulty results in the experiment which was basically predefined by the setup. The defined parameters in Subsection 3 already show that there are cases when the two transitions will (with a very high probability) merge. These expectations were documented in the „Tr. merge” column of the table with ranges in square brackets. One can see that these expectations were always fulfilled, but the values were in the lower range. This means that the CAA was able to distinguish in some cases between distributions which were very close to each other, however, this is most probably pure luck.

Of course, when an algorithm tries to adjust itself to a changing environment, the classification might flip, which in our case means that the transitions switch places. We label a case as a transition flip if the mean of the resulting transitions flip compared to the ground truth. This happened only in less than 5% of the cases, except the lognormal distribution. It means, that the CAA is able to accurately adjust itself to new models while it is able to maintain a constant distributions.

If we take a look at on the bad, or better said not fully accurate, models, the same statement can be made. A model on a transition was considered to be bad, if the normalized energy distance (Equation 6) between the KDE estimate and its target distribution exceeded the value 0.03. Please, note that this is not a p-value. A flipped model should automatically result in a bad model in this table, so that is responsible for a significant amount of the bad models. However, please bear in mind that there are model definitions very close to each other so merged models with very slight differences in the ground truth might not result in a bad model with or without a transition flip. This happened in the case of the normal and the exponential distributions. Generally, it can be said that in $\geq 91\%$ at least one, and in $\geq 84\%$ both transition models were accurate.

Dist. type	Accuracy	Precision 1	Precision 2	Recall 1	Recall 2
Normal	80.60% ± 0.43%	83.52% ± 0.41%	75.50% ± 0.50%	83.17% ± 0.43%	75.95% ± 0.48%
Uniform	89.92% ± 0.45%	91.71% ± 0.36%	86.59% ± 0.66%	91.25% ± 0.38%	86.69% ± 0.66%
Exponential	84.15% ± 0.37%	86.67% ± 0.37%	80.05% ± 0.39%	85.82% ± 0.38%	81.25% ± 0.39%
Lognormal	80.86% ± 0.28%	80.18% ± 0.33%	81.05% ± 0.25%	83.82% ± 0.28%	77.36% ± 0.29%
Weibull	88.07% ± 0.24%	89.27% ± 0.24%	85.96% ± 0.27%	89.30% ± 0.23%	85.93% ± 0.27%

Table 1: Model state classification results. The values with 1 in the header, refer to the transition where a model change was introduced, while the values with 2 in the header refer to a transition where the model was kept constant.

Dist. type	Bad model tr. 1	Bad model tr. 2	Bad model both	Tr. flips	Tr. merge
Normal	1.44%	2.03%	1.11%	1.81%	4.54%, [2.78% – 5.56%]
Uniform	3.29%	3.29%	2.92%	0.74%	1.39%, [1.39% – 2.78%]
Exponential	5.28%	3.52%	1.81%	2.96%	3.94%, [2.78% – 5.56%]
Lognormal	15.15%	12.45%	8.03%	7.39%	5.31%, [2.92% – 5.84%]
Weibull	7.39%	5.82%	4.81%	3.98%	3.41%, [2.38% – 4.76%]

Table 2: Model estimation results. The values with 1 in the header, refer to the transition where a model change was introduced, while the values with 2 in the header refer to a transition where the model was kept constant. The values in [] in the Transition merge column refer to the expected rate of transition merges due to the experiment setup.

One can also see that the lognormal distribution results are by far the worst for the CAA. However, theoretically there should be no significant limitation for this distribution type. A high amount of bad models can be traced back to transition flips. After a deeper analysis, it turned out that 90% of the models classified to be bad can be traced back to heavily tailed lognormal distributions. This is most probably a result of inaccurate KDEs and a too small window size. We assume that by increasing the window size these model problems would disappear.

The CAA is highly parallelized in a time step and it has a low memory footprint (around 1-2 GB-s) compared to the previous algorithm described in [2] and [3]. As a result, the average execution time varied between 1-5 minutes which is superior to the old algorithm. However, this is still around 10 – 20x higher than the same VSS without the CAA.

4 Conclusion

The experiments show that the CAA is efficiently able to track and to adapt to model changes in CHnMMs positively affecting the symbol classification accuracy.

Doing so does not affect the capability of withstanding changes on transitions where no change occurred.

By using the algorithm, the construction of CHnMM based VSSs is possible without deep analysis of the model parameters. It is enough to inject a distribution independent sample upon start.

The CAA also resolves some old practical issues with Proxel-based simulations, like the problem of concurrent exponential distributions, limitation through fixed distribution types and runtime changes of the model, which were not manageable before.

There is no theoretical limitation that would speak against the application of the CAA. It results in a very high probability of better evaluation and decoding results than manually parametrized models. Therefore, its usage is generally encouraged. However, if higher execution times are not acceptable, one might want to compromise and enable the functionality only on selected transitions.

Applications and Future Research The CAA opens up more accurate simulation possibilities for non-stationary models, like production lines and other human-influenced system.

Future research possibilities include the generalization of the CAA for VSS related problems beside CHnMMs. The transition flip problem could be reduced by adding penalty terms for transition movements. Additionally, the variable pruning can be evaluated as general pruning algorithm for VSSs.

Acknowledgement

The authors thank [22], [23] and [24] for making their work available on a free and open-source basis.

References

- [1] Krull C. *Virtual Stochastic Sensors: Formal Background and Example Applications: Reconstructing the Behavior of Partially Observable Discrete and Hybrid Stochastic Systems*. Shaker. 2021.
- [2] Bodnár D. *Change Detection of Model Transitions in Proxel Based Simulation of CHnMMs*. Magdeburg: Otto-von-Guericke-Universität. 2016.
- [3] Bodnár D, Krull C, Horton G. Change Detection of Model Transitions in Proxel Based Simulation of CHnMMs. In: *Analytical and Stochastic Modelling Techniques and Applications*, edited by Thomas N, Forshaw M. Cham: Springer International Publishing. 2017; pp. 32–46.
- [4] Buchholz R, Krull C, Horton G. Virtual Stochastic Sensors: How to gain insight into partially observable discrete stochastic systems. *The 30th IASTED International Conference on Modelling*. 2011;.
- [5] Wilson E. Virtual sensor technology for process optimization. 1997. Presentation at the ISSCAC 1997.
- [6] Buchholz R. *Conversive Hidden non-Markovian Models*. Magdeburg: Otto-von-Guericke-Universität. 2012.
- [7] Fink GA. Markov Models for Pattern Recognition. In: *Advances in Computer Vision and Pattern Recognition*. 2014; .
- [8] Krull C, Horton G. Hidden non-Markovian Models: Formalization and solution approaches. *6th Vienna International Conference on Mathematical Modelling*. 2009;.
- [9] Lazarova-Molnar S. *The Proxel-Based Method: Formalisation, Analysis and Applications*. Magdeburg: Otto-von-Guericke-Universität. 2005.
- [10] Horton G. A new paradigm for the numerical simulation of stochastic Petri nets with general firing times. In: *Proc. European Simulation Symp. ESS'02*. 2002; .
- [11] Rosenblatt M. Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*. 1956;27(3):832 – 837.
- [12] Parzen E. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*. 1962;33(3):1065–1076.
- [13] Gramacki A. *Nonparametric Kernel Density Estimation and Its Computational Aspects*. Cham: Springer International Publishing. 2018.
- [14] Salgado-Ugarte I, Perez-Hernandez M. Exploring the Use of Variable Bandwidth Kernel Density Estimators. *Stata Journal*. 2003;3:133–147.
- [15] Jones M. Simple boundary correction for kernel density estimation. *Statistics and Computing*. 1993;3:135–146.
- [16] Rizzo ML, Székely GJ. Energy distance. *WIREs Computational Statistics*. 2016;8(1):27–38.
- [17] Magel RC, Wibowo SH. Comparing the Powers of the Wald-Wolfowitz and Kolmogorov-Smirnov Tests. *Biometrical Journal*. 1997;39(6):665–675.
- [18] Arnold TB, Emerson JW. Nonparametric goodness-of-fit tests for discrete null distributions. *R Journal*. 2011;3(2).
- [19] Rizzo ML. A test of homogeneity for two multivariate populations. *Proceedings of the American Statistical Association, Physical and Engineering Sciences Section*. 2002;.
- [20] Szekely G. E-statistics: The Energy of Statistical Samples. 2002.
- [21] Szekely G, Rizzo M. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*. 2013;8.
- [22] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat , Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 10 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020; 17:261–272.
- [23] Rizzo M, Szekely G. *energy: E-Statistics: Multivariate Inference via the Energy of Data*. 2022. R package version 1.7-9. URL <https://CRAN.R-project.org/package=energy>
- [24] Carreño CR. *vnmabus/dcor: Version 0.5*. 2020. URL <https://doi.org/10.5281/zenodo.3996697>

An Extension for the Specification and Automated Selection of System Variants Based on the System Entity Structure Using a Problem from Process Industry

Hendrik Folkerts^{1*}, Thorsten Pawletta¹, Umut Durak²

¹Research Group Computational Engineering and Automation, University of Applied Sciences Wismar, Philipp-Müller-Straße 14, 23966 Wismar, Germany; *hendrik.folkerts@hs-wismar.de

²Institute of Informatics, Aeronautical Informatics, Technical University Clausthal, Albrecht-von-Groddeck-Straße 7, 38678 Clausthal-Zellerfeld, Germany

SNE 33(1), 2023, 17-25, DOI: 10.11128/sne.33.tn.10633
Selected ASIM SST 2022 Postconf. Publication: 2023-02-01;
Received Revised Improved: 2023-03-04; Accepted: 2023-03-10
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Modeling and Simulation (M&S) is widely used in various fields of engineering, manufacturing or process industry to investigate different system variants. An increasing problem of M&S is the complexity and variety of system variants. This concerns on the one hand the modeling effort and on the other hand the management of system variants in simulation studies. A general approach to the specification of different system variants is offered by the System Entity Structure (SES). It describes a set of system designs with different system structures and parameter configurations. In combination with a Model Base (MB), an SES can be used to describe different configurations of simulation models. The SES/MB framework and extended software architectures based on it define methods for the automated selection of system/model variants as well as for the generation and execution of simulation models. In this paper, the special descriptive element of the SES, the multi-aspect, is discussed. It is shown, how with hierarchically arranged multi-aspects certain forms of system variants can be specified very efficiently. Furthermore, a method for an automated derivation of system variants is presented in the context of hierarchical multi-aspects. This is important for the automation of simulation studies. For a practical illustration of the general method, it is presented using a problem from the process industry.

Introduction

Today's systems are often characterized by a high degree of variability. Variability modeling means to describe several system configurations. A system configuration represents one variant characterized by a system structure and parameter settings. Zeigler [1] introduced with the System Entity Structure (SES) a general high level approach for variability modeling. To describe and manage different configurations of simulation models, the SES was combined with a Model Base (MB) and extended to an SES/MB framework [2, 3]. The MB is a repository for organizing a set of basic dynamic models. Moreover, the framework specifies two general methods: (i) the pruning method for selecting specific system configurations from an SES and (ii) the build method for generating executable Simulation Models (SMs). The general framework does not define concrete algorithms for these methods.

Since the introduction of the SES/MB framework, it has been continuously developed by different researchers, such as presented in [4, 5, 6, 7, 8, 9, 10]. There are several approaches to the pruning method. Originally, pruning is an interactive process. However, interactive pruning is costly and error prone for a high number of variants coded in an SES [6]. Therefore, automation of the pruning process is crucial.

The automated goal-driven selection of system variants is a prerequisite for automating simulation studies involving different system configurations. Accordingly, Schmidt [9] proposes an extended SES/MB-based software architecture to automate simulation studies in the MATLAB/Simulink environment. Based on [9], a soft-

ware architecture is developed in [10] that supports the generation of executable simulation models for different target simulators. Among other things, this is based on the use of the Functional Mock-up Interface (FMI). In particular, the build method and the MB have been further developed so that, in addition to the SES, the MB is also largely simulator-independent.

A long unsolved problem has been the automated derivation of system configurations from an SES when using multi-aspects hierarchically in an SES [5, 6]. Zeigler and Hammonds [5] propose restructurings of the SES for this purpose. In [11], the authors developed an alternative approach which, in their view, is much easier to implement. This paper focuses on the basic approach in [11]. SES modeling with multi-aspects is introduced step by step and deepened by means of an example from the process industry. Before that, basic aspects of SES' are briefly discussed and an SES/MB-based software architecture is presented, in which the pruning method has been integrated to automate simulation studies.

1 Some Basics of SES

An SES is a tree structure with entity nodes, descriptive nodes, and attributes. While entity nodes describe an object of the real or imaginary world, descriptive nodes specify the relations among at least two entities. The descriptive nodes are divided into aspect, multi-aspect, and specialization node types. Aspect and multi-aspect nodes describe the composition of an entity. Coupling relations between entities can be specified in a special attribute (*couplings*). The multi-aspect node is a special aspect node that specifies a composition of several entities of the same kind. Number of Replications (num-Rep) is an additional attribute, which can be used to define a variable number of entities. Specialization nodes specify the taxonomy of an entity. For automated pruning specialization nodes have to define a selection rule as attribute. Zeigler [1, 5] defined six axioms for the construction and pruning of an SES. Applying the axioms, it follows among other things, that the root node is always an entity, representing several or one system configuration. The leaf nodes represent entities that are not further decomposed. Like descriptive nodes, entity nodes can specify attributes to define characteristic features. For example, as Schmidt [9] shows, it is useful to define the reference to a model in an MB and its parameter settings as attributes of a leaf node.

In terms of variant and variability modeling, multi-aspect and specialization nodes represent variation points. In order to derive one specific system configuration by pruning an SES all variation points have to be resolved by evaluating the node attributes. However, not only attributes at nodes of variation points have to be evaluated. Pruning can also involve adjustments to coupling relationships or attribute changes to entity nodes when resolving specializations. The result of each pruning operation is a Pruned Entity Structure (PES). A PES codes exactly one system configuration. There are several ways to implement a pruning method [4, 6]: (i) interactive pruning, (ii) automated pruning, (iii) enumerative pruning to derive all possible variants, (iv) selective pruning to derive one variant, etc. This paper focuses on the last one, the automated, goal-directed selection of exactly one system variant by pruning. For this purpose, we use the extension of the SES with information about pruning in the node attributes according to [8, 9, 12].

2 An Extended SES/MB-based Software Architecture

For applications in the field of Modeling and Simulation (M&S), the SES/MB framework was introduced [2, 3]. Basic dynamic models, which are referenced from leaf nodes in an SES, are organized in an MB. In addition to the pruning method discussed previously, the framework must provide a build method for generating executable SMs. In the following, we briefly present an extended SES/MB-based software architecture, which supports an automation of simulation studies.

The architecture is shown in Figure 1. The concept of the architecture was introduced in [8]. Besides the two new components, Experiment Control (EC) and Execution Unit (EU), also some new SES features like SES variables (SESvars) and SES functions (SESfcns) have been introduced.

The EC is a higher-level control unit. It defines the experiment goals, steps, and settings. Experiment steps and settings can reactively depend on previous simulation results. The EC activates the other components and evaluates their operations. The EU is an interface to target simulators on which the generated SM is executed. It is a kind of wrapper and uses the Application Programming Interface (API) of a target simulator to execute the simulation and to collect results. The SES/MB framework provides an interface to communicate with

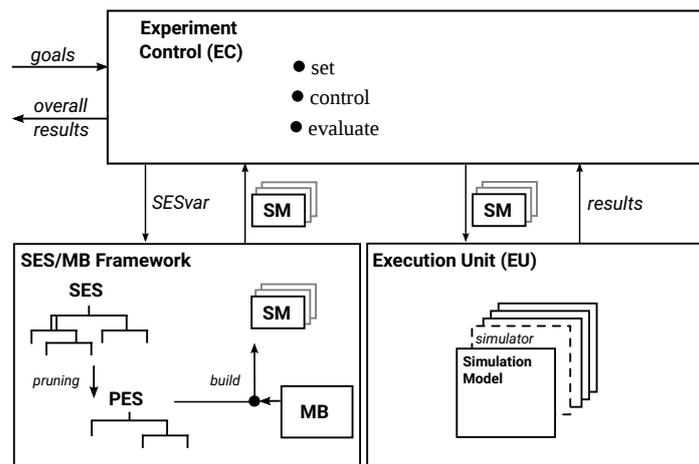


Figure 1: Extended SES/MB-based software architecture.

the EC. The newly defined SESvars are used as input interface. As output, the framework returns the SM derived by pruning and generated by the build method to the EC.

Using SESvars, value assignments to attributes of the SES can be defined variably and depending on settings in the EC. The value of the SESvars is determined before a pruning operation depending on the experiment specification in the EC. With regard to the SES, the SESvars are variables with a global scope. The same applies to the newly introduced SESfcns. SESfcns allow the specification of procedural knowledge and can be called in attributes of the SES. A typical application is the definition of dynamic coupling relations [8].

Currently there are two implementations of the architecture, which are freely reusable [13, 14].

3 Modeling and Pruning of SES with Multi-Aspect Nodes

Multi-aspect nodes are a powerful modeling element. However, pruning without user interaction is challenging for multi-aspects with a succeeding specialization or for several multi-aspect nodes in one path. Unlike the other descriptive nodes, pruning a multi-aspect does not lead to a reduction of the tree, but to an expansion due to the replication of the following entity node.

Restructuring SES' to avoid hierarchies of multi-aspects and specializations as suggested in [5] is challenging to automatize. Additional attributes may be needed for the restructured SES describing the same set

of system configurations and it is doubtful how values can be assigned to these attributes during pruning. In the next subsections it is demonstrated how hierarchies of multi-aspect nodes in combination with specialization nodes can be pruned automatically.

For illustration, we use a problem from the field of process industry, which is stepwise extended. Flexibilization in the process industry through modularization leads to a large number of process alternatives and parameter variants in plant design and operation [15]. On the other hand, modularization increases the reusability of models or model components [16]. The variant management in modular plant design according to [17] could be efficiently handled with the SES/MB approach.

3.1 Single Multi-Aspect

The simplest case is an SES with a single multi-aspect in a path as depicted in Figure 2. Here, the SES specifies a plant system that can consist of any number of identical partial plants.

In the SES tree, the root entity *plant* is followed by the multi-aspect node *plantMASP* with the attributes *numRep* and *couplings* and this is followed by the entity node *partialPlant* with the attribute *_partialPlant*. The *numRep* attribute at node *plantMASP* describes the varying number of *partialPlants* and the *couplings* attribute describes their coupling relations.

Above the SES tree the SESvar *NumPartialPlants* is defined as input interface. The following Semantic Condition describes the permissible value range of the

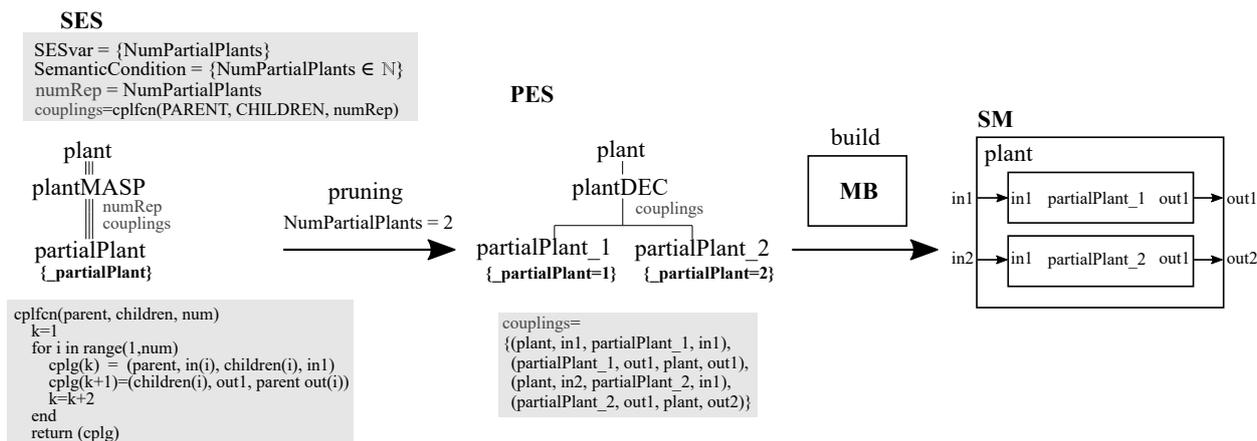


Figure 2: SES with a single multi-aspect, derivation of a possible PES, and the resulting model.

SESvar *NumPartialPlants*. After that the attribute *numRep* is defined using the SESvar *NumPartialPlants* and the *couplings* attribute using an SESfcn. The SESfcn *cplfcn* is called with the implicit variables *PARENT* and *CHILDREN*, and the *numRep* attribute as input parameters. The implicit variables refer to the parent and children nodes of the multi-aspect. The SESfcn specifies a parallel connection of partial plants. The attribute values were not defined directly in the tree only for reasons of clarity.

Before pruning, the SESvar must be assigned a value to initialize the attribute *numRep*. Then, during pruning the entity node *partialPlant* is replicated according to the current value of its attribute *numRep*. Since entities are replicated at a multi-aspect node, the entities following a multi-aspect are called generating entity in [5]. We propose to add at a generating entity node an attribute starting with an underscore followed by the name of the generating entity. The value of this underscore attribute remains undefined in the SES. Pruning implicitly assigns a value that represents a numbering of the generated entities. Thus, the entities are distinguishable based on the attribute value. The reason of this procedure is demonstrated in the next subsection.

Under the SES, Figure 2 shows the derivation of a possible PES by pruning. In the example, the value two was assigned to the SESvar *NumPartialPlants* before pruning. Accordingly, two replications of the entity node *partialPlant* are generated with the names *partialPlant_1* and *partialPlant_2*, and their attribute values are implicitly assigned to them. Also, the

multi-aspect node is converted into an aspect node and the *couplings* attribute is computed using the SESfcn. Thus, the aspect node describes the parallel composition of the two partial plants.

The derived PES describes exactly one system configuration. If one extends the leaf node *partialPlant* by an attribute with a link to a basic model in an MB, an SM could be generated with the build method as shown in the resulting model. To focus on the new extensions, the specification of coupling relations and some other attributes, such as references to the MB, are omitted in the following subsections and the build method is not considered.

3.2 Multi-Aspect with Succeeding Specialization

The example from Section 3.1 is now extended in the form of describing configurations of a plant system consisting of a variable set of subplants with identical input/output interfaces, but which are structured internally differently. The two subplant types of power station and waste treatment are used as an example. The further structural decomposition of the subplants is neglected here, but it is shown in the next subsection in Figure 4.

Figure 3 shows in the left part the specification of the problem with an SES. A specialization node called *partialPlantSPEC* is added to the SES following the generating entity *partialPlant* of the multi-aspect *plantMASP*. The specialization node defines a taxonomy of its parent node *partialPlant*, which can be assigned to either the *power station* or *waste treatment* category.

The child entity nodes describe the categories. How the category is assigned when pruning the SES is defined in the *specrule* attribute of the specialization.

The box above the SES tree in Figure 3 defines the input interface and the two attributes *numRep* and *specrule* of the SES. The input interface has been extended by the SESvar *PartialPlantTypes*. The Semantic Condition specifies that the SESvar *PartialPlantTypes* is a vector whose dimension must correspond to the value of the SESvar *NumPartialPlants* and whose elements can have the values 'ps' or 'wt'.

The Semantic Condition is followed by the definition of an SESfcn and the two SES attributes. The attribute *numRep* is defined analog to the example in Figure 2. The *specrule* attribute assigned to the *partialPlantSPEC* node defines rules for selecting the partial plant category using the SESfcn.

The middle and right tree in Figure 3 show step-by-step results of a pruning operation. In the example shown, the SESvar were previously assigned as follows: *NumPartialPlants* = 2 and *PartialPlantTypes* = ['ps', 'wt'].

In the 1st pruning step the multi-aspect *plantMASP* is resolved and the two entities *partialPlant_1* and *partialPlant_2* are generated. The subsequent subtree starting with the specialization node is appended to each of the generated entities. The result of this step is called *intermediate PES*. In the 2nd step the variation point specified by the specialization *partialPlantSPEC* is resolved. Pruning a specialization results in the union of the parent node with one selected child. Children, which are not selected, are removed like the specialization node itself. The selection is controlled by the rules in the *specrule* attribute. In this example, the SESfcn *ppTypes* is called in the *specrule*. It receives as input arguments the value of the underscore attribute *_partialPlant* of the parent node and the vector defined in the SESvar *PartialPlantTypes*. Depending on the value in *_partialPlant* an element of the vector *PartialPlantTypes* is returned, which describes the category to be selected and thus the selection of a child node. The selected child node and the parent node are merged into one entity node, as indicated by the merged node name in the PES.

The underscore attribute on the generating entity of a multi-aspect and the implicit value assignment during pruning when generating the entities makes branches in the tree distinguishable. The distinguishability enables the subsequent automated assignment to different cate-

gories when resolving the specialization node.

3.3 Several Multi-Aspects and Specializations in a Common Path

The introduced example is now extended to two multi-aspects and two specializations in a common path. A third category of subplant, called *chemicalProduction*, is introduced. A plant may comprise several subplants of this type. A subplant *chemicalProduction* may in turn consist of a varying number of further subplants, called *chemicalSubProduction*, serving either *acid* or *base* production. This extension increases the number of possible system configurations exponentially.

Figure 4 shows the specification of the extended problem with an SES and Figure 5 shows step by step the pruning to derive a possible PES.

The SES. The first four layers of the tree correspond to the SES in Figure 3. In the fourth layer, the entity node *chemicalProduction* was added as a further category of a *partialPlant*. The configuration of the *powerStation* and *wasteTreatment* type subplants is not illustrated.

The multi-aspect *chemicalProductionMASP* with the subsequent entity node *chemicalSubProduction* describes the composition of any entity *chemicalProduction* from interface-compatible entities *chemicalSubProduction*. Through the subsequent specialization node *chemicalSubProductionSPEC*, a categorization of each entity *chemicalSubProduction* into *acid* or *base* production is performed.

The box above the SES tree specifies the necessary SESvars, SESfcns, and node attributes. Their semantics are explained below in the step-by-step description of pruning to derive exactly one system configuration.

A pruning example. The SES input interface is extended by the two SESvars *NumChemicalSubProductions* and *ChemicalProductionTypes*, which are formally defined in the Semantic Condition. To derive a system variant, the four variables must be assigned values before pruning. We derive a system configuration as depicted in Figure 5.

1st and 2nd pruning steps: The 1st pruning step is performed analogously to Section 3.1. According to the *numRep1* attribute at the *plantMASP* node, four entity nodes *partialPlant* are generated. Then, the 2nd pruning step is executed analogously to Section

SES

```

SESvar = {NumPartialPlants, PartialPlantTypes}
SemanticCondition = {NumPartialPlants ∈ ℕ ∧
  PartialPlantTypes=[e1, e2, ..., en] ∧ ei ∈ {'ps', 'wt'} ∧ n==NumPartialPlants}
SESfcn: ppTypesFun(_partialPlant, PartialPlantTypes) numRep = NumPartialPlants
  return(PartialPlantTypes(_partialPlant)) specrule = {ppTypesFun(_partialPlant, PartialPlantTypes) == "ps" → powerStation
  ppTypesFun(_partialPlant, PartialPlantTypes) == "wt" → wasteTreatment }

```

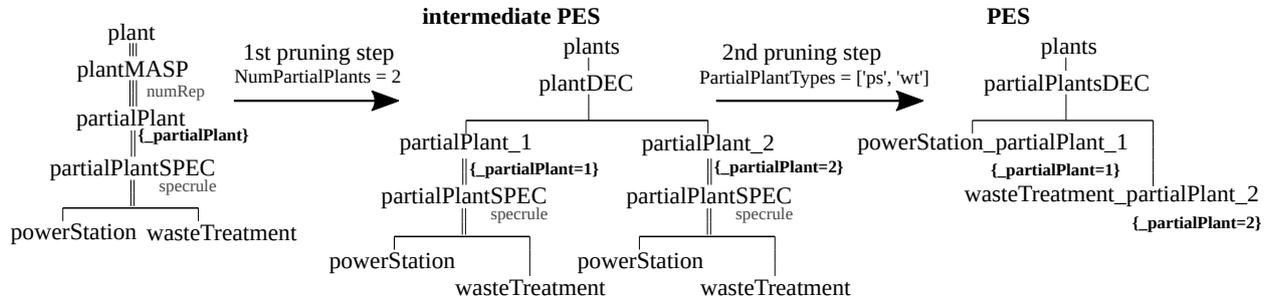


Figure 3: SES with a multi-aspect followed by a specialization and stepwise derivation of a possible PES.

3.2. By pruning the node *partialPlantSPEC* with the attribute *specrule1*, the four previously created entity nodes *partialPlant_1 ... partialPlant_4* are specialized according to the value assignment of the SESvar *PartialPlantTypes* to: *powerStation_partialPlant_1*, *chemicalProduction_partialPlant_2*, *chemicalProduction_partialPlant_3*, and *wasteTreatment_partialPlant_4*. The last operation of the 2nd pruning step is to attach the remaining subtree of the SES to the two entity nodes *chemicalProduction_partialPlant_2* and *chemicalProduction_partialPlant_3*. The result of these pruning steps is illustrated in Figure 5 as *intermediate PES 2*.

3rd pruning step: In the third pruning step the two nodes *chemicalProductionMASP* are resolved. For each of the two nodes, the number of replications of the *chemicalSubProduction* node to generate is computed by the attribute *numRep2* using the SESfcn *cpNumFun*. The principle operation of *cpNumFun* corresponds to *ppTypesFun* in Section 3.2 (Figure 3). Function *cpNumFun* evaluates the underscore attribute *_partialPlant* at the parent in the *intermediate PES 2* as well as the SESvar *PartialPlantTypes* and *NumChemicalSubProductions* and calculates the number of entities to be generated. Besides, an underscore attribute is implicitly added for each *chemicalSubProduction_i* node generated. The variability of a multi-aspect is resolved after entity replication. It is renamed to an aspect and the remaining subtree of the SES is attached to each node created. In this case, the subtree starts with *chemicalSubProductionSPEC*. The result of the 3rd pruning

step is called *intermediate PES 3* in Figure 5. It should be noted that the couplings attribute must also be adjusted, as shown in Section 3.1.

4th pruning step: In the fourth pruning step, the specialization *chemicalSubProductionSPEC* is resolved for each entity *chemicalSubProduction_i*. This results in the categorization of each *chemicalSubProduction_i* into an *acid* or *base* production. The value assignments of the SESvar *ChemicalProductionTypes* define the system configuration to be derived in this respect. The necessary selection rules are variably defined in the *specrule2* attribute of the SES using the SESfcn *cpTypesFun*. The operation of *cpTypesFun* corresponds to the previous explanations of the SESfcn. The return value 'ac' or 'ba' decides which specialization to select. Deleting the specialization node and uniting the selected child node with the parent node is done analogously to Section 3.2. The final result of pruning is shown with the PES in Figure 5.

Short evaluation. The implicitly managed underscore attribute on entities generated during pruning in combination with SESfcns allows automated derivation of a system configuration (PES) specified with SESvar. However, the complexity of SESfcns increases significantly as the number of multi-aspects in a path increases. The axiom of uniformity specified for the SES states: nodes with the same name need to have the same variables and isomorphic subtrees. In the SES this axiom is fulfilled, but relaxed in the PES. In Figure 5 the nodes *chemicalProductionDEC* do not have isomorphic

SES

```

SESvar = {NumPartialPlants, PartialPlantTypes, NumChemicalSubProductions, ChemicalProductionTypes}
SemanticCondition = {NumPartialPlants ∈ ℕ ∧
    PartialPlantTypes=[e1, e2, ... ,en] ∧ ei ∈ {'ps', 'cp', 'wt'} ∧ n==NumPartialPlants ∧
    NumChemicalSubProductions=[e1, e2, ... ,en] ∧ ei ∈ ℕ ∧ n==|{x | x ∈ (PartialPlantTypes=='cp')}| ∧
    ChemicalProductionTypes=[e1, e2, ... ,en] ∧ ei=[a1, a2, ... ,am] ∧ n==|NumChemicalSubProductions|
    ∧ ai ∈ {'ac', 'ba'} ∧ m==NumChemicalSubProductions(ei)
}

SESfcn: ppTypesFun(_partialPlant, ppTypes)
    #map _partialPlant to ppTypes
    #return type_of_partial_plant
    cpTypesFun(_partialPlant, _chemicalSubProduction, ppTypes, cpTypes)
    #map _partialPlant to ppTypes
    #if ppType is 'cp'
    # map _partialPlant and _chemicalSubProduction to cpTypes
    # return type_of_chemical_subProduction
cpNumFun(_partialPlant, ppTypes, cpNum)
    #map _partialPlant to ppTypes
    #if ppType is 'cp'
    # map to cpNum
    # return number_of_chemical_subProductions

numRep1 = NumPartialPlants
specrule1 = {ppTypesFun(_partialPlant, PartialPlantTypes) == "ps"→ powerStation
    ppTypesFun(_partialPlant, PartialPlantTypes) == "cp"→ chemicalProduction
    ppTypesFun(_partialPlant, PartialPlantTypes) == "wt"→ wasteTreatment
}

numRep2 = cpNumFun(_partialPlant, PartialPlantTypes, NumChemicalSubProductions)
specrule2 = {cpTypesFun(_partialPlant, _chemicalSubProduction, PartialPlantTypes, ChemicalProductionTypes) == "ac"→ acid
    cpTypesFun(_partialPlant, _chemicalSubProduction, PartialPlantTypes, ChemicalProductionTypes) == "ba"→ base}
    
```

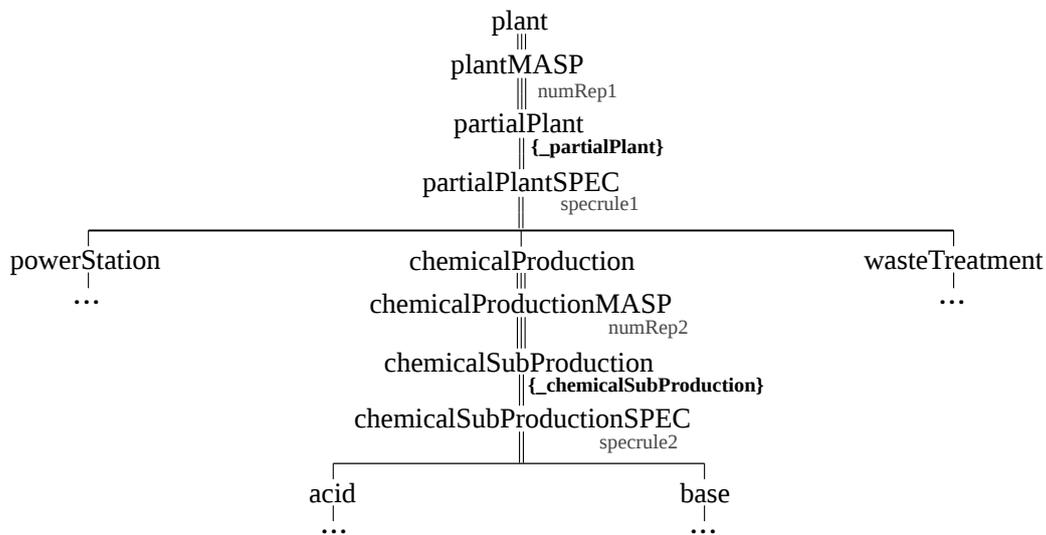


Figure 4: SES with two multi-aspects in one path and each followed by a specialization.

subtrees. In the context of simulation engineering this does not pose any problem. The extended pruning approach is implemented in the Python-based toolset [14].

4 Conclusion

The example has shown that the combination of multiple multi-aspects with subsequent specializations in an SES path supports a compact specification of a large number of system variants. The introduction

of an implicitly managed attribute on entities generated by multi-aspects during pruning supports an automated derivation of a goal-directed system configuration (PES). Thus, an automatic model generation using the extended SES/MB architecture is supported.

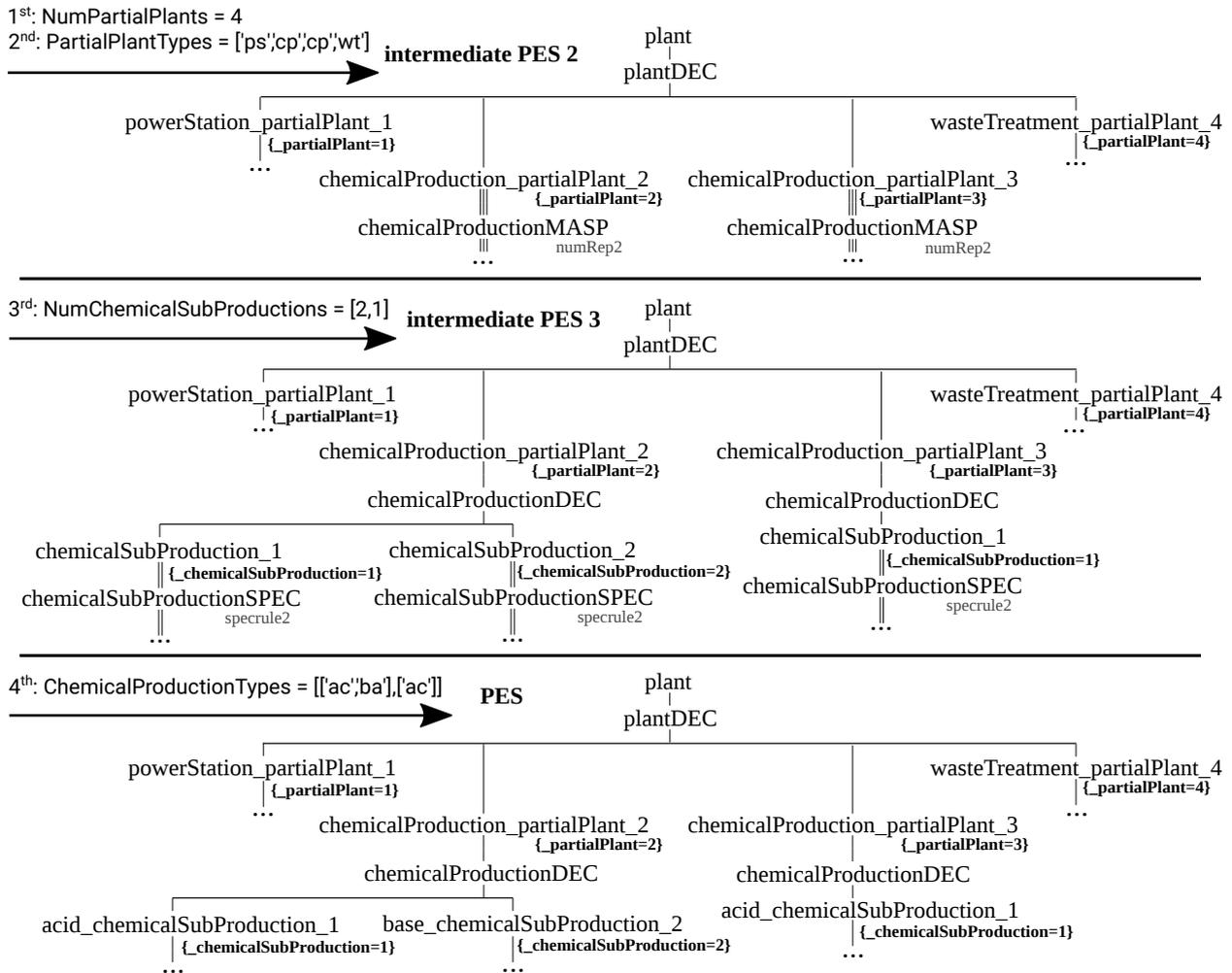


Figure 5: Step-by-step derivation of a system variant by pruning.

References

- [1] Zeigler BP. *Multifaceted Modelling and Discrete Event Simulation*. USA: Academic Press Professional, Inc. 1984.
- [2] Rozenblit JW, Zeigler BP. Representing and constructing system specifications using the system entity structure concepts. In: *Proceedings of the 25th Winter Simulation Conference, Los Angeles, California, USA, December 12-15, 1993*, edited by Evans GW, Mollaghasemi M, Russell EC, Biles WE. ACM Press. 1993; pp. 604–611.
- [3] Zeigler BP, Kim TG, Praehofer H. *Theory of Modeling and Simulation*. USA: Academic Press, Inc., 2nd ed. 2000.
- [4] Rozenblit JW, Huang Y. Rule-Based Generation of Model Structures in Multifaceted Modeling and System Design. *INFORMS J Comput.* 1991;3(4):330–344.
- [5] Zeigler BP, Hammonds PE. *Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange*. Orlando, FL, USA: Academic Press, Inc. 2007.
- [6] Zeigler BP, Sarjoughian HS. *Guide to Modeling and Simulation of Systems of Systems*. Simulation Foundations, Methods and Applications. Springer. 2013.
- [7] Santucci JF, Capocchi L, Zeigler BP. System Entity Structure Extension to Integrate Abstraction Hierarchies and Time Granularity into DEVS Modeling and Simulation. *Simulation.* 2016;92(8):747–769.
- [8] Pawletta T, Schmidt A, Zeigler BP, Durak U. Extended Variability Modeling Using System Entity Structure Ontology Within MATLAB/Simulink. In: *Proceedings of the 49th Annual Simulation Symposium, ANSS '16*. San Diego, CA, USA: Society for Computer Simulation International. 2016; pp. 22:1–22:8.

- [9] Schmidt A. Variant Management in Modeling and Simulation Using the SES/MB Framework. Ph.D. thesis, Rostock University. 2019.
- [10] Folkerts H, Pawletta T, Deatcu C. Model Generation for Multiple Simulators Using SES/MB and FMI. *SNE Simulation Notes Europe*. 2019;31(1):25–32.
- [11] Folkerts H, Pawletta T, Deatcu C, Zeigler BP. Automated, Reactive Pruning of System Entity Structures for Simulation Engineering. In: *Proceedings of the 2020 Spring Simulation Conference, SpringSim '20*. San Diego, CA, USA: Society for Computer Simulation International. 2020; .
- [12] Deatcu C, Folkerts H, Pawletta T, Durak U. How to Define SES Trees for Variability Modeling. *SNE Simulation Notes Europe*. 2019;29(3):117–126.
- [13] RG CEA. MATLAB SES Tbx. https://github.com/cea-wismar/SES_Tbx_Matlab. 2023. Accessed Feb. 28, 2023.
- [14] RG CEA. Python-based SES/MB Architecture. https://github.com/cea-wismar/SESMB_Inf_Python. 2023. Accessed Feb. 28, 2023.
- [15] DECHEMA. Modulare Anlagen – Flexible chemische Produktion durch Modularisierung und Standardisierung - Status quo und zukünftige Trends. *Tech. rep.*, Germany. 2017.
- [16] von Wedel LA. An Environment for Heterogeneous Model Management in Chemical Process Engineering. Ph.D. thesis, RWTH Aachen. 2004.
- [17] Hady L, Wozny G. Multikriterielle Aspekte der Modularisierung bei der Planung verfahrenstechnischer Anlagen. *Chemie Ingenieur Technik*. 2012;84:597–614.

Hestia.jl: A Julia Library for Heat Conduction Modeling with Boundary Actuation

Stephan Scholz*, Lothar Berger

Control and Process Engineering, Ravensburg-Weingarten University of Applied Sciences, Germany
Web: <https://forschung.rwu.de/forschungsgruppen/control-and-process-engineering>

*stephan.scholz@rwu.de

SNE 33(1), 2023, 27-30, DOI: 10.11128/sne.33.sn.10634
Selected ASIM SST 2022 Postconf. Publication:2023-02-01;
Received Revised Improved:2023-02-27; Accepted:2023-03-10
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Heat conduction modeling in three dimensions with boundary actuation plays an important role in thermal process engineering, for example in case of heating plates, laser welding or 3D printing. Here, the actuators and the induced energy have to be described exactly for such processes to guarantee a high simulation quality. We introduce Hestia.jl, a software library to model three-dimensional heat conduction with multiple spatially distributed heat sources on the boundary.

Introduction

The heat equation is a standard example in numerical analysis and control theory. Several software tools and libraries like OPENFOAM [1], FENICS [2], TRIXI.JL [3] and VORONOI FVM.JL [4] exist to solve heat equation models in one, two or three dimensions. These software tools are general purpose solvers for (specific types of) partial differential equations, which means they are also applicable for other models like the Poisson or Burgers' equation. However, they often require solid knowledge in the theory of finite volume or finite element methods.

We present HESTIA.JL [5], a Julia library [6] to simulate heat conduction in one, two and three dimensions with boundary actuation. A discretized heat conduction model is created in few steps using HESTIA.JL, without deep knowledge of numerical analysis. This heat conduction model is solved in time using high-order numerical integrators provided by DIFFERENTIALEQUATIONS.JL [7].

1 Problem formulation

Heat conduction is often modeled to occur inside geometrical objects like one-dimensional rods $\Omega = (0, L)$, two-dimensional plates $\Omega = (0, L) \times (0, W)$ and three-dimensional cuboids $\Omega = (0, L) \times (0, W) \times (0, H)$. These objects are implemented in HESTIA.JL as data types HeatRod, HeatPlate and HeatCuboid to store the original dimensions and the spatial approximation including sampling and number of grid points.

We distinguish linear and quasi-linear heat conduction depending on the definition of its physical properties: thermal conductivity λ , specific heat capacity c and mass density ρ . Constant properties (in case of linear heat conduction) are stored as a StaticIsoProperty whereas temperature-dependent properties (in case of quasi-linear heat conduction) are stored as a DynamicIsoProperty. In particular, temperature-dependent properties are assumed to be modeled as power series, e.g. $\lambda(\Theta) = \sum_{n=1}^N a_n \Theta^{n-1}$, and the coefficients are saved in an array, e.g. $[a_1, \dots, a_N]$. So, we assume the quasi-linear heat conduction model

$$\rho(\vartheta) c(\vartheta) \frac{\partial \vartheta(t, x)}{\partial t} = \operatorname{div} [\lambda(\vartheta) \nabla \vartheta(t, x)]$$

with $(t, x) \in (0, T) \times \Omega$ and final time $T > 0$ to be a generalization of the linear heat equation.

On boundary $\partial\Omega$ the variation of temperature ϑ is affected by the boundary conditions, namely linear heat transfer

$$-h(x) ([\vartheta(\cdot, x) - \Theta_{amb}(x)])$$

and nonlinear heat radiation

$$-\varepsilon(x) \sigma [\vartheta(\cdot, x)^4 - \Theta_{amb}(x)^4]$$

to the environment with ambient temperature Θ_{amb} , heat transfer coefficient h , emissivity ε and Stefan-

Boltzmann constant σ , see also [8]. The parameters h , ε and Θ_{amb} are stored in data type `Emission` and can be defined for each boundary side separately, see Table 1.

West	$\{0\} \times [0, W] \times [0, H]$
East	$\{L\} \times [0, W] \times [0, H]$
South	$[0, L] \times \{0\} \times [0, H]$
North	$[0, L] \times \{W\} \times [0, H]$
Underside	$[0, L] \times [0, W] \times \{0\}$
Topside	$[0, L] \times [0, W] \times \{H\}$

Table 1: Names and positions of boundary sides.

2 Actuator configuration

Actuators are only assumed on boundary sides - not inside the geometrical object. The actuated boundary sides (west, east, etc.) are partitioned using a checkerboard pattern and for each partition β_n an actuator and its spatial configuration b_p can be defined as introduced in [9]. This spatial configuration describes the possible spatially distributed intensity of actuation ranging from zero (no actuation) to one (full actuation). We define a radial symmetric configuration as

$$b_p(x) = \begin{cases} m_p \cdot \exp(-\|M_p(x - x_{c,n})\|^{2\nu_p}) & \text{for } x \in \beta_n, \\ 0 & \text{for } x \in B_A \setminus \beta_n \end{cases}$$

with scaling $m \in [0, 1]$, curvature matrix $M \in \mathbb{R}^{3 \times 3}$, power $\nu \in \mathbb{N}_{>0}$ and central point $x_{c,n} \in \beta_n$ of the n -th partition. These coefficients are stored as `RadialConfiguration`. The choice of a spatial configuration shall approximate real-world scenarios where heating elements may not be able to induce the same amount of energy at each position of its surface. An example partition and its related actuator configuration are illustrated in Figure 1.

3 Demonstration example

Next, we explain how to build a simulation of a cooling-down and heating-up process for a three-dimensional cuboid. A full listing can be found on GitHub [10].

Setting up the model

We specify the physical properties of a quasi-linear heat conduction model as $\rho := 7800$,

$$c(\Theta) := 330 + 0.4 \Theta \quad \text{and} \\ \lambda(\Theta) := 10 + 0.2 \Theta - 10^{-4} \Theta^2.$$

These coefficients are stored in arrays as noted in Section 1 and a `DynamicIsoProperty` is created as in Listing 1.

```
property =
  createDynamicIsoProperty(
    [10.0, 0.2, -1e-4], [7800], [330, 0.4])
```

Listing 1: Create property type for quasi-linear dynamics.

We design a cuboid of length $L = 0.3$, width $W = 0.2$ and height $H = 0.1$, which is discretized by 40 cells in x_1 -, 24 cells in x_2 - and 10 cells in x_3 -direction. The model is built as in Listing 2

```
cuboid =
  HeatCuboid(0.3, 0.2, 0.1, 40, 24, 10, property)
```

Listing 2: Create cuboid model.

The ambient temperature of the boundary conditions is set to 300 Kelvin. On boundaries `west` and `east` we assume the heat transfer coefficient $h = 10$ and emissivity $\varepsilon = 0.6$, see Listing 3. On boundaries `north` and `east` we consider only heat transfer with $h = 10$ and no heat radiation, all other boundary sides are considered with zero-Neumann boundary conditions ($h = 0$ and $\varepsilon = 0$) and do not have to be implemented explicitly.

```
boundary = initBoundary(cuboid)
emission = createEmission(10, 0.6, 300)
setEmission!(boundary, emission, :west)
```

Listing 3: Specify boundary conditions.

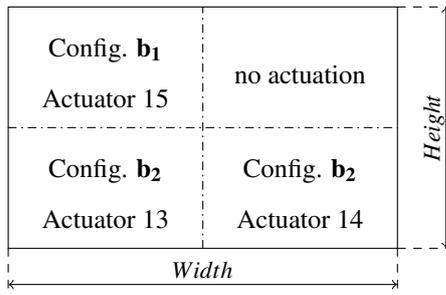
Cooling-down process

Now, the heat equation is approximated in space and forms an ordinary differential equation (ODE) which is solved with `DIFFERENTIALEQUATIONS.JL`. The specification (geometry, property, boundary conditions) and the temperatures are handed over to the `diffusion!` function to compute the right-hand side of the ODE, see Listing 4. The `COOL_DOWN!` function forms a standard interface for numerical ODE integration methods provided by `DIFFERENTIALEQUATIONS.JL`.

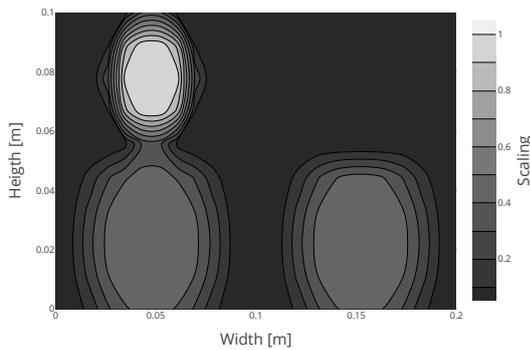
```
cool_down!(dv, v, p, t) =
diffusion!(dv, v, cuboid, property, boundary)
```

Listing 4: Define interface for ODE solver for cooling-down process.

The cooling-down process is not depicted here because we focus on the heating-up process as described next.



(a) Boundary partition



(b) Actuator configuration

Figure 1: Partition (a) and configuration (b) on east boundary at $x_1 = L$ with configurations $b_1: (m_1, M_1, v_1) = (1.0, 50 I_{3 \times 3}, 3)$ and $b_2: (m_2, M_2, v_2) = (0.5, 30 I_{3 \times 3}, 2)$.

Heating-up process

The heating-up process extends the previous steps by specifying the position and spatial configuration of actuators on the boundary sides as discussed in Section 2. In this example, we assume actuation on boundaries UNDERSIDE and EAST. The underside is subdivided in 4×3 partitions and for each of it an individual actuator with configuration b_1 where $m_1 = 1, M_1 = 50 I_{3 \times 3}, v_1 = 3$ is specified as in Listing 5.

```
config = setConfiguration(1.0, 3, 50)
```

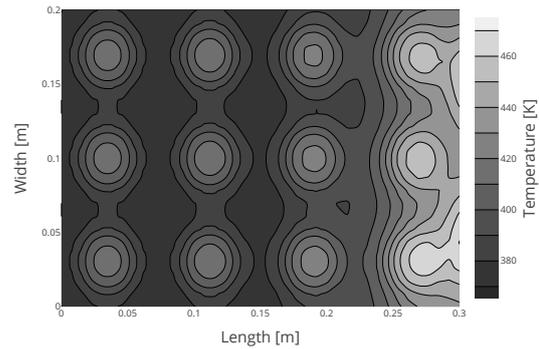


Figure 2: Temperature on the underside at $x_3 = 0$.

```
setIOSetup!(actuation, cuboid, (4,3),
config, :underside)
```

Listing 5: Specify actuation on underside.

Boundary EAST is subdivided manually with 2×2 partitions, as portrayed in Figure 1, where two fields are defined by configuration b_2 with $m_2 = 0.5, M_2 = 30 I_{3 \times 3}, v_2 = 2$, one field is defined by b_1 as noted above and one field is not actuated. See also the complete listing [10].

A constant heat input $u_n(t) = 4 \cdot 10^5$ for actuator $n = 1, \dots, 15$ (12 actuators on the underside and 3 on the east boundary) is set and the ODE interface is implemented as in Listing 6.

```
u_in = 4e5 * ones(15)
heating_up!(dv, v, param, t) =
diffusion!(dv, v, cuboid, property,
boundary, actuation, u_in)
```

Listing 6: Define interface for ODE solver for heating-up process.

The heating-up process is simulated for $T_f = 200$ seconds and the final temperature distribution on the underside (at $x_3 = 0$) and east boundary (at $x_1 = L$) are portrayed in Figure 2 and 3. They unveil the strong influence of the actuator configuration on the resulting temperature distribution which reaches up to 470 Kelvin. A three-dimensional temperature distribution is illustrated in Figure 4 for temperatures above 360 Kelvin.

4 Conclusion

We introduced the software library HESTIA.JL for modeling of three-dimensional heat conduction with boundary actuation. The recent version is able to approximate linear and quasi-linear (isotropic) heat conduction and

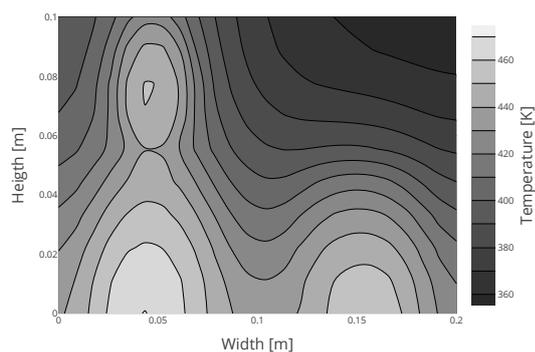


Figure 3: Temperature on the east boundary at $x_1 = L$.

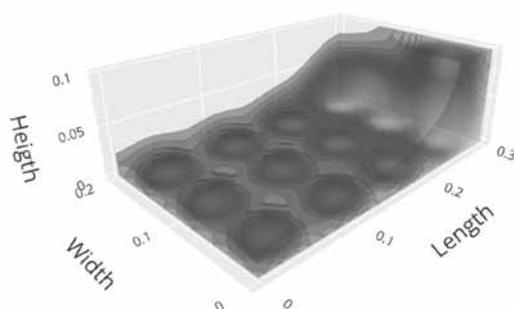


Figure 4: Temperature distribution in the cuboid for temperatures higher than 360 Kelvin.

to handle radial symmetric actuator configurations. Our further research will focus on the development of optimal control modules for HESTIA.JL, and on its good integration in Julia's Scientific Machine Learning ecosystem.

References

- [1] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*. 1998; 12(6): 620–631.
- [2] Logg A, Wells GN. DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software*. 2010; 37(2), 1-28.
- [3] Schlottke-Lakemper M, Gassner GJ, Ranocha H, Winters AR, Chan J. Trixi.jl. Zenodo. 2022. Available: <https://zenodo.org/record/6372038>
- [4] Fuhrmann J, contributors. VoronoiFVM.jl: Finite volume solver for coupled nonlinear partial differential equations. Zenodo. 2022. Available: <https://doi.org/10.5281/zenodo.6151074>
- [5] Scholz S. Hestia.jl. Zenodo. 2023. Available: <https://doi.org/10.5281/zenodo.7685941>
- [6] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. *SIAM Review*. 2017; 59(1): 65-98.
- [7] Rackauckas C, contributors. SciML/DifferentialEquations.jl. Zenodo. 2022. Available: <https://doi.org/10.5281/zenodo.5837925>
- [8] Baehr HD, Stephan K. *Heat and mass transfer*. Springer Science & Business Media, 2011.
- [9] Scholz S, Berger L. Modeling of a multiple source heating plate. *arXiv preprint arXiv:2011.14939*. 2020.
- [10] Scholz S. HestiaDemonstration.jl. GitHub. 2022. Available: <https://github.com/stephans3/HestiaDemonstration.jl>

Parameter-Free Approximation Method for Controlling Discrete Event Simulation by Reinforcement Learning

Daniel Pasterk*, Andreas Körner

Institute of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

* daniel.pasterk@tuwien.ac.at

SNE 33(1), 2023, 31-34, DOI: 10.11128/sne.33.sn.10635
 Selected ASIM SST 2022 Postconf. Publication:2023-02-01;
 Received Revised Improved:2023-02-27; Accepted:2023-03-10
 ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. A novel k NN-based approximation method for Reinforcement Learning (RL) is used to control and optimize a Discrete Event Simulation (DES). The method does not require design parameters, is suitable for unknown and new simulation environments, and can handle irregular and partially sparse state space. We show in a demonstrative queuing simulation that the method is more robust than artificial neural networks and achieves comparable performance.

Introduction

In this contribution we want to show how Discrete Event Simulations can be controlled and optimized using reinforcement learning. From a general perspective, it is promising to optimize simulations using RL. It provides the ideal framework to represent time-dependent decision problems in a very natural approach, while the simulations can generate sufficient data. Changes in the dynamics of environment can also be addressed via different exploration strategies. Although a large number of powerful algorithms have emerged since the 1960s, the broad availability of powerful computational resources, has extended its applicability in many areas. Starting with the impressive achievements of DeepQ-Networks [8] in 2013, this form of machine learning was becoming increasingly popular. Controlling a simulation with RL follows the framework of Markov Decision Processes (MDPs), where an agent interacts with an environment. The agent receives the state of the environment and decides on an action, while the environment responds with a new state and a reward. The agent tries to choose his actions in a way that the cumu-

lative reward is maximized. The simulation therefore represents the environment, in which the agent interacts within the described loop.

While the basic architecture is very clear, the difficulty often arises at dealing with the formal requirements of the framework. It is an important requirement to represent the simulation as an MDP. While the optimization goals and control options can often be represented very directly with a corresponding reward and action set, the so-called Markov property must be fulfilled for the state variable: The entire (relevant) information must be able to be encoded in a single state which has to be independent of its history. In order to fulfill these preconditions feasibly, there are various approaches: The simulation can already be developed with the background of this requirement and the corresponding state variables can be equipped with a well-suited encoding. Though it has to be mentioned here that special knowledge about the nature of the simulation is relevant. This somewhat undermines the claim of a model-free method. As an alternative to the "hand-crafted" states, one can take the internal simulation variables directly. These can basically be used as states in the MDP framework and should satisfy the Markov property, but it sadly leads to a very high complexity. In addition, the resulting state space is very irregular - many combinations of the individual components of the state vector are not even possible due to the simulation logic. Large areas of the state space are therefore not used, while in other areas there is a lot of information in a very dense form. Hence, a suitable approximation method has to be found, which can handle these specific properties sufficiently well. This contribution presents a non-parametric approximation method based on the idea of k -nearest-neighbor classification algorithm, which is very well suited for the mentioned challenges. We demonstrate the method in the context of classical Q-learning and Monte-Carlo algorithms.

1 Related Work

We want to give a small selection of publications, which illustrate the successful application and then go into specific implementations: In 1998, Mahadevan and Theodorou [7] use the customized algorithm "SMART" to optimize transfer lines from a fab including predictive maintenance. The results show superior performance compared to the classical 'KANBAN' heuristic. In Waschneck et al. (2018) [11], DQN is successfully applied to control a digital twin of a semiconductor fab. The DQN actions control specific dispatch heuristics to increase overall performance. Shuhui Qu, Jie Wang, and Shivani (2016) [9] use RL to solve a dispatching problem in factories. A simulation was created special as MDP and the learned rules are more cost effective than most well-known heuristics. Doltsinis, Ferreira, and Lohse (2014) [3] use RL to optimize a production startup in an 'automatic assembly station'. The use of a batched Q-learning algorithm shows the general suitability of this decision-supported approach. Some selection of general approaches, which focuses on the general connectivity of RL with simulations: Capocchi et al. (2022) [5] show how DEVS properties can be used within a RL integration focusing on temporal, hierarchical and multi-agent aspects. Lang et al. (2021) [6] demonstrate the application of DEVS to a production planning problem. In this approach, it is shown how control can be performed using a standardised interface (OpenAi Gym). In Greasley (2020) [4], various software products from simulation and machine learning are examined for connectivity. Some interesting considerations are also made about the structure of a useful connectivity component. In Choo et al. (2020)[2], the general problem that discrete event simulations cannot be directly translated into MDP is pointed out. It is proposed to translate the state space and action space of the simulation into their counterparts in the RL formalism using equivalence classes. The approach sounds interesting, but unfortunately no statement is made about the concrete form of the relationship.

2 Method

2.1 Architecture

The architecture of our setup can be summarized in Figure 1. For the interface of the simulation we use the interface of OpenAi-Gym [1], by which the states, actions

and the reward are exchanged. The simulation-internal variables are made accessible to the agent as a state vector without further processing. The agent has no further information about the model apart from the state vector and the simulation is technically controlled by this model-free agent. Both the implementation of the code for the approximation and learning feature, as well as the implementation of the simulation was done in the programming language Julia. The framework "SimJulia" was used for the simulation and the modeling follows the agent-based paradigm.

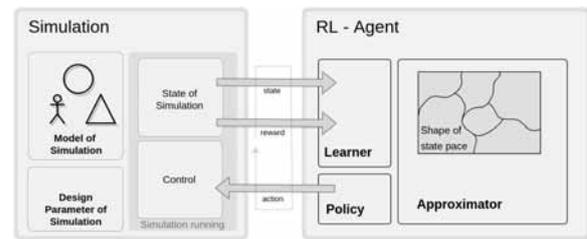


Figure 1: General overview of the architecture of the simulation and the RL agent. The approximator must represent the underlying state space sufficiently well.

2.2 Approximation Method for Reinforcement Learning

On the agent side, we strongly separate between the elements of "learner", "approximator" and "policy". This is also illustrated in Figure 1. For the case-study below, we use the update rules from classical Q-learning. For the state space S , action space A , $\alpha \in \mathbb{R}$, $\gamma \in \mathbb{R}$ and a possible reward $r \in \mathbb{R}$ the update rule can be written as

$$q(s, a) := q(s, a) + \alpha(r + \gamma \max_{a \in A(s')} (q(s', a)) - q(s, a)). \quad (1)$$

with $s \in S$ and $a \in A$. We use this update rule in connection with various approximation methods such as "State Aggregation", "Tile-Coding" and "Linear Regression".

We also use Monte-Carlo-1st-Visit [10] and a Deep Q-Network [8] with two different capacities as a reference.

Since A is finite, an optimal policy π is immediately calculated as

$$\pi(s) := \operatorname{argmax}_{a \in A(s)} q(s, a), \quad (2)$$

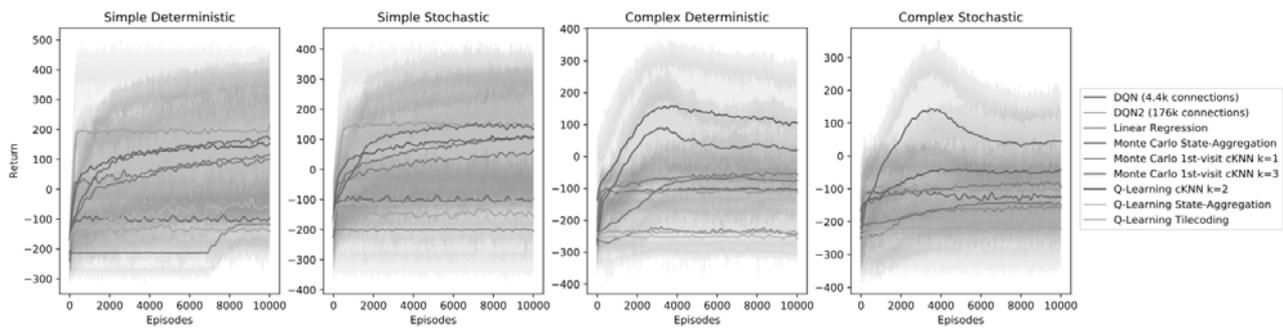


Figure 2: Performance of k -NN approximation in combination with Q -Learning and first-visit Monte Carlo compared to other common approximation methods, in simple-deterministic ($n = 20$), simple stochastic ($n = 20$), complex-deterministic ($n = 20$), and complex-stochastic tasks ($n = 20$).

where ties are broken randomly, after having computed the action-value function q .

Our method for approximating $q(s, a)$ is based on the idea that the collected interactions between agents and environment are grouped into $(k$ -d)-Trees to the respective action $a \in A$. The RL update rules provide an error that is added to the current estimate of an existing query. To obtain $q(s, a)$ in a query, every KD-tree is searched individually for the k nearest neighbors of a state for each action and the q-value is calculated by the arithmetic mean over the k stored values.

3 Queuing Simulation as Case Study

3.1 Simulation Description

We consider 4 different scenarios from a typical queuing simulation at a service desk. In the simple model, the agent can change the configuration once in each time window between 0 and 5 open cash registers, while in the more complex model 10 changes can be made per time window and it is possible to open up to 7 cash registers simultaneously. The service time for each customer is 0.25 hours, and the staff budget is 5 hours in both cases. In the stochastic case, the arrival times are sampled from a Poisson distribution where we have peaks in the later morning and in afternoon. The patience p of the customers is sampled from a uniform distribution between 3 and 5, and the service time is sampled from a normal distribution with mean 0.25 hours and standard deviation $\sigma = 0.05$ hours. In the more complex case, the agent can make a decision and rede-

ploy staff 10 times per hour. Therefore, the complexity increases slightly. This simulation is designed in such a way that in the deterministic case, 60 customers visit the bank per day, and 15 staff hours would theoretically be required to serve for all customers. However, there are far too few resources. The state-space of the agent consists of the time of day, the length of the queue, and the number of open desks. One episode is divided into ten-time units. A reward of +10 is received for a completed request and -10 for an abort.

3.2 Experimental Results

Over 10k episodes were simulated for all 4 scenarios. Mean and standard deviation were determined over 20 independent runs. Known approximation methods from the literature were taken up as a comparison. These were configured to the best of our knowledge. The results can be seen in Figure 2.

In the simple case, classical DQN can adapt very well to both the deterministic and the stochastic variants. Linear regression, even in the simple case, is able to provide useful state approximation only at a very late stage. In the more complex case, the kNN variants lead very quickly to a good policy, while other methods would probably require further and more costly adaptation of the design parameters. We can verify that the approximation by the k NN method works sufficiently well in both cases and that classical methods by state aggregation do not have sufficiently well approximation properties in connection with RL. Neural networks also provide a good performance, but were complex to configure in order to achieve a learning behavior.

4 Discussion and Conclusion

We observe that the state space approximation by k -NN is superior to classical simple methods. In performance comparison with DQN, the universal approximation property of neural networks can provide good performance here. A fairly big advantage of this method in the context of simulations is the robustness against proper ranges of values in the state representation and the necessary design parameters in the approximation itself. These are serious advantages in the development and actual usage as an optimization method. While in parametric approximation methods one has to find a very balanced dimensioning for a good performance, this is not necessary in the present case. Furthermore, the simulation variables can be taken directly and do not need to be specially prepared by some pre-processing. This is remarkable especially in the case of neural networks, where the performance can be very relevant mostly due to the size of the network and also the encoding of the information in the first layer. While the k NN approximation does not require any design parameters, for the other methods a complex evaluation process was necessary to show any learning behavior at all. Especially the choice of the neural network leaves only a narrow range between over and underfitting. Therefore, we see the advantages of the k -NN approximation in connection with classical RL algorithms especially in the parameter-free operation.

5 Future Work

Subsequently, we will look at how well this method works for more complex simulations with non-equidistant decision logic. We also want to extend our basic architecture to agent-based simulations that are not explicitly designed for use within an MDP. Another opportunity is to apply this non-parametric method with current policy gradient algorithms in the actor-critic design pattern. We expect it to be particularly well suited in the stochastic case.

References

- [1] Brockman G et al. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).
- [2] Choo B et al. "Reinforcement Learning from Simulated Environments: An Encoder Decoder Framework". In: *2020 Spring Simulation Conference (SpringSim)*. May 2020, pp. 1–12.
- [3] Doltsinis S, Ferreira P, and Lohse N. "An MDP Model-Based Reinforcement Learning Approach for Production Station Ramp-Up Optimization: Q-Learning Analysis". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.9 (Sept. 2014), pp. 1125–1138.
- [4] Greasley. A. "Architectures for Combining Discrete-event Simulation and Machine Learning". In: *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications - SIMULTECH*, INSTICC. SciTePress, 2020, pp. 47–58.
- [5] L C and Jean-François S. "Discrete Event Modeling and Simulation for Reinforcement Learning System Design". In: *Information* 13.3 (2022).
- [6] Lang S, Kuetgens M, Reichardt P, and Reggelin T. "Modeling Production Scheduling Problems as Reinforcement Learning Environments based on Discrete-Event Simulation and OpenAI Gym". In: *IFAC-PapersOnLine* 54.1 (2021). 17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021, pp. 793–798.
- [7] Mahadevan S and Theodorou G. "Optimizing Production Manufacturing Using Reinforcement Learning". In: *Proceedings of the Eleventh International Florida Artificial Intelligence Research Society Conference*. AAAI Press, 1998, pp. 372–377.
- [8] Mnih V et al. "Playing Atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).
- [9] Qu S, Wang J, and Shivani G. "Learning adaptive dispatching rules for a manufacturing process system by using reinforcement learning approach". In: *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. Sept. 2016, pp. 1–8.
- [10] Sutton RS and Barto AG. *Reinforcement Learning: an Introduction*. 2nd. The MIT Press, 2018.
- [11] Waschneck B et al. "Optimization of global production scheduling with deep reinforcement learning". In: *Procedia CIRP* 72 (2018). 51st CIRP Conference on Manufacturing Systems, pp. 1264–1269.

A Hybrid User Model for Virtual Stochastic Sensors

Claudia Krull

Faculty of Computer Science, Otto-von-Guericke-University, Magdeburg, 39106 Germany ; claudia.krull@ovgu.de

SNE 33(1), 2023, 35-43, DOI: 10.11128/sne.33.tn.10636
 Selected ASIM SST 2022 Postconf. Publication:2023-02-01;
 Received Revised Improved:2023-02-27; Accepted:2023-03-10
 ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Virtual stochastic sensors (VSS) enable the reconstruction of unobserved behavior of discrete or hybrid stochastic systems from observable output. Augmented stochastic Petri nets (ASPN) are user models for VSS and describe discrete stochastic models that produce discrete output symbols based on the transitions or system states. Hybrid ASPN (H-ASPN) can describe hybrid stochastic systems with continuous quantities that are influenced by and interact with the discrete system parts, producing samples of these continuous system quantities as observable output. Real-world systems often contain both of these types of observable output. In household energy models, the total consumption is a continuous quantity that can be sampled regularly. Additionally, the usage behavior of some appliances might be known in advance or can be monitored easily, resulting in observable discrete symbols. Being able to utilize both of these for behavior reconstruction, promises better results than using either one alone. In this paper, we describe an extended H-ASPN paradigm for modeling symbol output as well as sample measurement for partially observable hybrid stochastic systems. We demonstrate the paradigm on a small non-intrusive appliance load monitoring (NIALM) example and test the behavior reconstruction. The extended H-ASPN modeling paradigm enables faster and more reliable behavior reconstruction results, when using both observable symbols and samples. The experimental results indicate that extended H-ASPN could lead the way to practically feasible VSS for hybrid systems.

Introduction

Simulation and modeling are used in a forward manner to observe existing systems, build abstract representations and experiment with these in order to draw conclu-

sions on the original systems behavior. Many systems are however not directly observable, but only through their output or interaction with the environment. To analyze these partially observable systems, a backward approach of behavior reconstruction based on the observed output becomes necessary.

Virtual stochastic sensors (VSS) formalize and solve the inverse problem of determining unobservable likely stochastic system behavior based on observable stochastic output. In [1] VSS for discrete and hybrid systems are formalized and tested on academic and real-world applications. Two VSS user models are introduced: Augmented stochastic Petri nets (ASPN) can describe discrete systems that generate output in the form of discrete observable symbols from an alphabet. This output can be triggered by the firing of a transition or based on the current system state. Hybrid augmented stochastic Petri nets (H-ASPN) describe hybrid stochastic systems with continuous reward measures that are sampled independently of the system behavior to generate output protocols.

In a partially observable real-world system, the symbols or samples that are observable can be very different in nature and generated by a wide range of processes: light barriers can detect the passing of an object or person; physical sensors can measure temperature, air pressure, or light intensity; RFID readers can detect the time and ID of an item passing or residing within their range. Most of these can be categorized as either sample or discrete symbol, but a single system may generate both types of observable output. e.g. smart meters can record household energy consumption, which is a continuous measure, and electrical usage monitors can detect when a specific appliance was switched on and off, which can be interpreted as a specific symbol. Currently, ASPN can model discrete signal emissions, and H-ASPN the samples of continuous measures, but there exists no modeling paradigm, that can incorporate both types of emissions. Restricting the model to only one type of emission would disregard readily available information and therefore likely lead to mediocre results

compared to an approach including both types of emissions. Therefore, we introduce a model type that can represent both samples and symbols.

In Section 1 some background on VSS and other related work is given. The formalization of an extended hybrid augmented stochastic Petri net is shown in Section 2, also addressing necessary changes to the underlying solution algorithm. We demonstrate the paradigm and the behavior reconstruction capability using a NIALM problem in Section 3, showing the benefit of combining symbol and sample output. Section 4 will discuss the results and implications.

1 State of the Art

The concept of virtual stochastic sensors (VSS) was introduced in [2] and has since evolved into a framework to describe and solve backward problems in modeling and simulation ([1]). VSS enable the behavior reconstruction of a broad range of partially observable stochastic systems based on the observable output.

Originally, hidden non-Markovian models (HnMM) were developed as computational models for VSS ([3]). Increasing feasibility, conversive hidden non-Markovian models restrict the modeling power to enable a much more efficient solution ([4]). Hybrid hidden non-Markovian models (HHnMM) expand the paradigm to include continuous measures ([5]). The Proxel method ([6]) is used for the actual behavior reconstruction task. It is a state space-based simulation method, which enables a deterministic analysis of stochastic models with arbitrarily distributed process durations employing the method of supplementary variables. Behavior reconstruction for discrete systems of realistic size is currently feasible. However, the extension to hybrid systems increases the computational complexity drastically, making hybrid VSS only applicable to small scale academic models so far.

The user models defined for VSS are augmented stochastic Petri-nets (ASPN), which were first introduced in [4]. ASPN are based on well known stochastic Petri net paradigms ([7, 8]) and expand these using ideas from hidden Markov models (HMM) ([9]) by emissions of discrete symbols. These emissions can either depend on the current system state or can be triggered by the firing of a transition. In both cases, the emission time is recorded in a protocol along with the emitted symbol. The hybrid components in H-ASPN are modeled using ideas from stochastic reward nets

(SRN) ([10]) and fluid stochastic Petri nets (FSPN) ([11, 12]). The observable output of a hybrid ASPN is generated by an independent sampling process, that records the values of one or more of the continuous quantities in a protocol along with the sampling time stamp. [1]

By utilizing both continuous and discrete output for VSS behavior reconstruction, we hope to increase the degree of observability of the systems. The concept of observability is also found in control theory, where Kalman filters are used to estimate the development of a system. What is also similar to VSS is the goal of determining a systems internal state from external measurements. Originally, Kalman Filters are designed to estimate the development of deterministic linear systems in discrete steps. Since then, Kalman Filters have been extended to deal with stochastic and non-linear systems and are widely used to predict unobserved or future states of such systems. The general idea is to predict the system development based on an estimate of the current state, and then correct the prediction based on possibly noisy measurements. [13, 14]

VSS differ from Kalman Filters in that it is not necessary to formalize the exact structure of the state and observation equations, including the noise and error terms. Instead, by mimicking the system development and exploring the expanded model state space step by step, we not only avoid the mathematical complexity of Kalman Filters, but also allow for infrequent measurements, observations stemming from state changes and completely unobservable states. VSS represent an addition to the myriad of tools for estimating unobservable quantities and can be employed when the structure and dynamics of the partially observable system are known and can be described by a discrete or hybrid stochastic model such as the ones presented here.

2 Hybrid Augmented Stochastic Petri Nets

In this section, we describe the combination of ASPN and H-ASPN in detail, including the individual model components. The notation and semantics use the formalization from [1]. Therefore, the individual elements do not differ from the ones described there, but only their combination.

Formally, an H-ASPN is a tuple

$$HnMM = (P, T, I, O, H, M_0, V, b, W, w_0, GF, ir, rr)$$

with the following elements:

- $P = \{p_1 \dots p_{|P|}\}$ is a set of *Places* representing physical locations or system states. The marking of the H-ASPN is given by the distribution of tokens in the places of the net $M \in \hat{M} = \mathbb{N}^{|P|}$.
- $T = \{t_1, \dots, t_{|T|}\}$ is a set of transitions, which is partitioned into a set of immediate transitions T_I and a set of timed transitions T_T .
 - An element of T_I is associated with a probability $T_I \rightarrow [0, 1]$, denoting the firing probability in competition situations. If the probability is not stated, it defaults to 1. Immediate transitions can fire as soon as they are enabled, according to their assigned probability.
 - An element of T_T is associated with an arbitrary continuous distribution function and possibly a memory policy (race-age or race-enable). This distribution describes the firing time, which needs to elapse between enabling and firing of a transition.
- $I, O, H : P \times T \rightarrow \mathbb{N}_0$ are incidence functions and specify the connection between places and transitions.
 - If $I(p, t) > 0$, an input arc leads from place p to transition t . The value of $I(p, t)$ determines the number of tokens that need to be present in place p for transition t to be enabled, and likewise the number of tokens destroyed in p , when t fires.
 - If $O(p, t) > 0$, an output arc leads from transition t to place p . The value of $O(p, t)$ determines the number of tokens that are created in place p when t fires.
 - If $H(p, t) > 0$, an inhibitor arc leads from place p to transition t . The value of $H(p, t)$ determines the number of tokens that need to be present in p for transition t to be disabled.
- $M_0 = (m_1 \dots m_{|P|})$ ($M_0 : P \rightarrow \mathbb{N}_0$) holds the initial marking of the H-ASPN, where m_i denotes the number of tokens in place p_i at the initialization of the system.
- $V = \{v_1 \dots v_{|V|}\}$ is the set of discrete output symbols of the net.
- $b : V \times P \cup V \times T \rightarrow [0, 1]$ describes the output behavior of the net, mapping the element generating

the output and the output symbol to an output probability. In most cases, outputs are associated either to transitions or to places, a combination is however also conceivable and therefore not ruled out.

- If the output is generated depending on the current system state, then $b : V \times P \rightarrow [0, 1]$. If a state can produce output symbols, then the sum of the probabilities of all output symbols of the state must be 1: $\exists b(v_i, p_j) > 0 \Rightarrow \sum_{k=1 \dots |V|} b(v_k, p_j) = 1$.
- If the output is generated depending on the ASPN transitions, then $b : V \times T \rightarrow [0, 1]$. If a transition can produce output symbols, then the sum of the probabilities of all output symbols of that particular transition must be 1: $\exists b(v_i, t_j) > 0 \Rightarrow \sum_{k=1 \dots |V|} b(v_k, t_j) = 1$.
- $W : \{w_1 \dots w_{|W|}\}$ is a set of variables, representing continuous system quantities. The current values of these quantities are given in $w_{|W|} \in \hat{W} = \mathbb{R}^{|W|}$.
- $w_0 = (w_1 \dots w_{|W|})$ contains the initial values of the continuous system quantities.
- $GF : T \times \mathbb{N}^{|P|} \times \mathbb{R}^{|W|} \rightarrow 0, 1$ describes the marking dependent guard of each transition. If the Boolean expression evaluates to 1, the transition is enabled, and it is disabled otherwise.
- $ir : T \times \hat{M} \times \hat{W} \rightarrow \mathbb{R}$ (or $\mathbb{R}^{|W|}$) describes a type of impulse reward, which can change the value of a continuous quantity immediately. The change can be dependent on the current marking of the ASPN.
- $rr : P \times \hat{M} \times \hat{W} \rightarrow \mathbb{R}$ (or $\mathbb{R}^{|W|}$) describes a type of rate reward, which can change the value of a continuous quantity continuously. The rate of change can be dependent on the current marking of the net. The rate reward can be associated to a specific place or can be completely independent of the discrete system state and be represented by an ODE.

The marking of the places and the values of the continuous quantities together form the potential state space of the H-ASPN $\hat{M} \times \hat{W}$. The elements P, T, I, O, H, M_0 were taken from Petri nets and are widely known. For more details on the semantics and dynamics of SPN refer to general Petri net literature [7, 8].

Elements V and b hold the augmentation information. The specific output semantics are the following:

when a transition with associated output symbols fires, one of these symbols is emitted, sampled according to their probabilities. When symbols are associated with places or specific system states, observations are made through an independent process. The symbol is sampled depending on the current system state and associated output symbols. Observed outputs are collected in a protocol with their respective emission or sampling time stamps. Samples of the continuous measures can be taken at any time, and will be collected in a protocol associated with the time when they were drawn. The protocol can contain values of different continuous quantities, but not every continuous system quantity must be represented.

2.1 Modifications to Computational Model and Solution Algorithm

In order to enable behavior reconstruction for this modified H-ASPN model type, we also need to adapt the computational model and solution algorithm. Analogously to the combination of the user models, the computational model was created by combining the output processes of hidden non-Markovian models (HnMM) and hybrid hidden non-Markovian models (HHnMM) [1] to incorporate discrete symbol emissions as well as samples of continuous model variables.

As solution method, we extended the Proxel-based analysis algorithm for hybrid models (see [1, 5]) to perform behavior reconstruction for extended H-ASPN. The trace can now contain samples of the continuous model quantities as well as discrete output symbols, and both will be handled accordingly. As described in [5], the parametrization process for the Proxel method for hybrid systems is tedious, since the inclusion of continuous variables considerably increases the complexity of the models. Countering the inherent problem of state-space explosion of the Proxel method becomes more difficult with more method parameters. Instead of having a binary decision on the validity of a Proxel, as with discrete outputs, the estimate of the continuous measure in the Proxel can be within ϵ of the observed sample. The size of ϵ and the cutoff probability or number for pruning the Proxel tree both need to be balanced, in order to achieve a useful reconstruction result in a computationally feasible manner.

In the following experiment section, we will show how the behavior reconstruction works for the extended paradigm.

3 Example and Experiments

For our proof of concept, we will use a small example constructed from the UMass Smart* Data Set for Sustainability [15] (2013 release). The data was gathered from several private homes over the period of several months, and includes aggregate power readings, circuit and appliance level consumption as well as environmental data. The goal of non-intrusive appliance load monitoring (NIALM) is to dis-aggregate a cumulative household energy consumption into the contributions of individual appliances or circuits. For demonstration purposes, we have chosen a subset of three circuits with different characteristics from House A, namely the *Dryer*, *MasterLights* and *CounterOutlets1*. For a detailed description of the model construction, please refer to [16]. The data set uses UNIX time stamps, and thus seconds as basic time unit. The goal for the experiments is to determine the unobserved system behavior in terms of appliance state changes from the cumulative power consumption, with and without information on individual appliances state changes.

3.1 NIALM Example System

We parameterized the model using the circuit level data of five consecutive days (May 6th to 10th), and performed behavior reconstruction using the aggregate power consumption of the following day (May 11th). The graphical representation of the H-ASPN can be found in Figure 1. Well known Petri-net elements are represented as follows, places as circles, timed transitions as open rectangles, immediate transitions as vertical bars associated with a probability, and arcs as solid line arrows. The places represent tangible appliance usage states (*Dr1*, *Dr2*, *Dr3*, *ML1*, *ML2*, *CO1*, *CO2*). The timed transitions associated with the state changes of the *MasterLights* (*MLon*, *MLoff*) and the *CounterOutlets* (*COon*, *COoff*) emit a symbol when firing, denoted by a dotted arrow. The different rate rewards in the different appliance states are depicted by dashed arrows annotated with the average consumption rate in that state. Since the appliances have separate state spaces, the rewards of the currently active states are added to form the overall current consumption (collected in *Reward r*). For clarity reasons, not all elements of the H-ASPN are named.

In the formal description of the H-ASPN in Figure 2, *H*, *ir* and *GF* are omitted, since the system does not

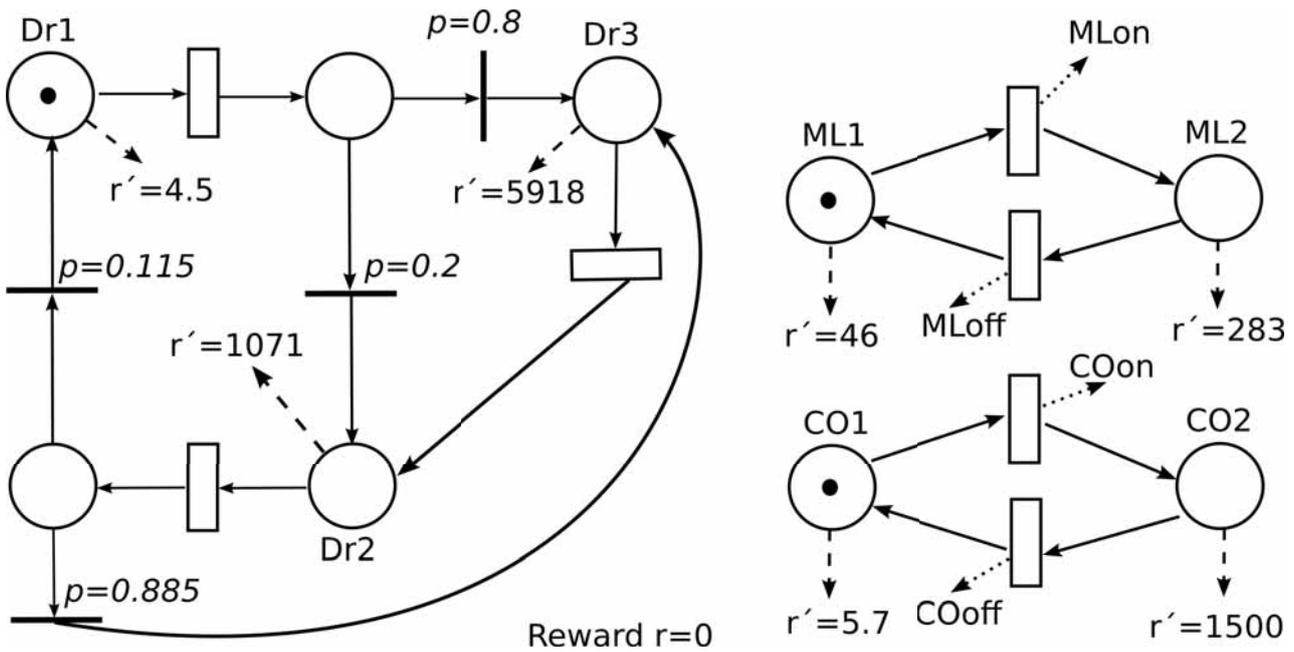


Figure 1: H-ASPN of the NIALM Example Process

include inhibitor arcs, impulse rewards or guard functions.

The timed transitions governing the appliance state changes were fitted with Weibull distributions and Exponential distributions with the same expected values. The Weibull distribution was used, since it mimics the actual switching behavior of the appliances best [16], and the Exponential distribution as comparison, because it is least restrictive, being memoryless. In contrast to the models used in [16], where histograms are used to represent a small number of different consumption levels per state, we only used one average power consumption value per state.

3.2 Validation Experiment

The cumulative consumption trace of May 11th was extracted from the Smart* Data Set, as were the state change points of the *MasterLights* and the *CounterOutlets1*. The time stamps were normalized to start at 0 with the first item of the trace. An excerpt of the trace is depicted in Table 1. The ground truth, in terms of state changes in all three appliances is also known from the data set and depicted in Table 2, where one can see, that on the test day, the *Dryer* was not in use at all, but stayed in state *Dr1* with the lowest consumption.

In the experiments, we want to test, how includ-

Trace	
Time Stamp	Sample/Symbol
...	...
5169	287789
6175	466894
6175	<i>MLon</i>
6178	467708
6287	498265
6358	518035
6362	519155
6554	529992
6554	<i>MLoFF</i>
6714	538616
...	...

Table 1: Excerpt of the Cumulative Consumption Trace with Symbols

ing some symbols in the trace affects the accuracy of the reconstruction, compared to a reconstruction based only on samples. Therefore, we constructed the model in different variations: (0) no symbol emissions at all, (1) state changes of the *MasterLights* and *CounterOutlets* are detectable, however cannot be distinguished through the symbol emitted, (2) the state changes of *MasterLights* and *CounterOutlets* are detectable and can be distinguished from their symbols, which corresponds to the model formalized in the previous section.

$$\begin{aligned}
 \text{Circuits} &= (P, T, I, O, M_0, V, b, W, \vec{v}, rr) \\
 P &= \{Dr1, Dr2, Dr3, Dr1_{Done}, Dr2_{Done}, \\
 &\quad ML1, ML2, CO1, CO2\} \\
 T_I &= \{Dr1to2, Dr1to3, Dr2to1, Dr2to3\} \\
 T_T &= \{Dr1_T, Dr2_T, Dr3_T, ML1_T, ML2_T, \\
 &\quad CO1_T, CO2_T\} \\
 I(p, t) &= \begin{cases} 1: & (p = Dr1 \wedge t = Dr1_T) \\
 & \vee (p = Dr3 \wedge t = Dr2_T) \\
 & \vee (p = Dr2 \wedge t = Dr3_T (\\
 & \quad \vee (p = Dr1_{Done} \wedge \\
 & \quad (t = Dr1to2 \vee t = Dr1to3)) \\
 & \vee (p = Dr2_{Done} \wedge \\
 & \quad (t = Dr2to1 \vee t = Dr2to3)) \\
 & \vee (p = ML1 \wedge t = ML1_T) \\
 & \vee (p = ML2 \wedge t = ML2_T) \\
 & \vee (p = CO1 \wedge t = CO1_T) \\
 & \vee (p = CO2 \wedge t = CO2_T) \\
 0: & \text{else} \end{cases} \\
 O(p, t) &= \begin{cases} 1: & p = Dr1_{Done} \wedge t = Dr1_T \\
 & \vee p = Dr2_{Done} \wedge t = Dr2_T \\
 & \vee p = Dr1 \wedge t = Dr2to1 \\
 & \vee p = Dr2 \wedge t = Dr1to2 \\
 & \vee p = Dr2 \wedge t = Dr3to2 \\
 & \vee p = Dr3 \wedge t = Dr1to3 \\
 & \vee p = Dr3 \wedge t = Dr2to3 \\
 & \vee p = ML1 \wedge t = ML2_T \\
 & \vee p = ML2 \wedge t = ML1_T \\
 & \vee p = CO1 \wedge t = CO2_T \\
 & \vee p = CO2 \wedge t = CO1_T \\
 0: & \text{else} \end{cases} \\
 M_0 &= (1, 0, 0, 0, 0, 1, 0, 1, 0) \\
 V &= \{MLon, MLoFF, COon, COoff\} \\
 &\quad b(MLon, ML1_T) = 1 \\
 &\quad b(MLoFF, ML2_T) = 1 \\
 &\quad b(COon, CO1_T) = 1 \\
 &\quad b(COoff, CO2_T) = 1 \\
 &\quad \text{else } b(v, t) = 0 \\
 W &= r \\
 \vec{w}_0 &= (0) \\
 rr(Dr1) &= 4.5 \\
 rr(Dr2) &= 1071 \\
 rr(Dr3) &= 5918 \\
 rr(ML1) &= 46 \\
 rr(ML2) &= 283 \\
 rr(CO1) &= 5.7 \\
 rr(CO2) &= 1500
 \end{aligned}$$

Figure 2: Formal description of H-ASPN of the NIALM Example Process

Generating Path	
Time Stamp	State Change
6175	<i>ML1_T</i>
6554	<i>ML2_T</i>
6982	<i>ML1_T</i>
7381	<i>ML2_T</i>
7659	<i>CO1_T</i>
7874	<i>CO2_T</i>

Table 2: Ground Truth for Appliance State Changes

In case (0), the trace only contains samples of the continuous quantity, and the times when they were measured. In cases (1) and (2) the trace also contains symbols with the time stamps of when they were emitted.

For the experiments, we used a maximum simulation time of 10,000, encompassing all symbols and state changes in the ground truth, and a Proxel discretization time step of 60 seconds. For all experiments shown here, we varied the ϵ cutoff threshold and Proxel cutoff number to result in a feasible run, with smallest possible ϵ . Both parameters affect the number of paths discovered and the runtime [5, 1], and therefore will not be compared here directly. The ϵ parameter had to be chosen at 10,000 or more, since the time step of 60 could result in large gaps between actual and reconstructed state change times, resulting in larger deviations between the actual and the reconstructed consumption.

As a first test, we ran the Proxel analysis algorithm with the version (0) models and traces without symbol emissions. The model with Weibull distributions resulted in paths with 166-170 state changes, which corresponds to one state change in almost every time step of the analysis, most of these of *MasterLights* or *CounterOutlet*. Thus the reconstructed paths using the model with Weibull distributions bear no resemblance to the ground truth path, and are therefore not practically useful. The most likely path reconstructed using the model with Exponential distributions is shown in Table 3. The algorithm failed to reconstruct the state changes in the *MasterLights*, mis-matching two of them as *CounterOutlet* switches, but at least matching the switch timings. Considering the ambiguity and large time step size of the reconstruction, this is a mediocre result.

As a next step, we used model variant (2), where *MasterLight* and *CounterOutlets1* state changes are detectable and can be distinguished by the symbols emitted. Unfortunately, the paths reconstructed using the model with Weibull distributions again contained one state change in almost every time step, and were there-

Reconstructed Path	
Time Stamp	State Change
6120	<i>CO1_T</i>
6240	<i>CO2_T</i>
7620	<i>CO1_T</i>
7920	<i>CO2_T</i>

Table 3: Reconstructed Path Without Detectable State Changes

fore not useful here. The most likely path reconstructed using the model with Exponential distributions is shown in Table 4. The path resembles the ground truth very closely, only the *Dryer* state change from *Dr1* to *Dr2* at 6120 and back again 2 minutes later does not correspond to the actual system behavior. This shows that, even if only some state changes are detectable, the reconstruction resembles the ground truth much better than without detectable state changes.

Reconstructed Path	
Time Stamp	State Change
6120	<i>DR1_T</i>
6120	<i>DR1to2</i>
6180	<i>ML1_T</i>
6240	<i>DR2_T</i>
6240	<i>DR2to1</i>
6600	<i>ML2_T</i>
7020	<i>ML1_T</i>
7440	<i>ML2_T</i>
7680	<i>CO1_T</i>
7920	<i>CO2_T</i>

Table 4: Reconstructed Path With Distinguishable State Changes

In model variant (1) all detectable state changes emit the same symbol, which decreases the systems degree of observability by making the state changes indistinguishable. The reconstructed most likely path still corresponds to the one with distinguishable state changes depicted in Table 3, correctly matching *MasterLight* and *CounterOutlets1* state changes.

3.3 Performance Analysis

We also want to compare the algorithm performance when including or excluding symbols. We will only use the two models using Exponential distributions, since only these resulted in useful reconstructions. When comparing the runtime for both algorithms with the same ϵ and cutoff threshold parameters, the analysis of

the model without symbol emissions needs 20 seconds, whereas the analysis when including symbols needs only about 4 seconds.

Figure 3 shows the development of the number of valid paths over time. The upper graph shows the number oscillating around 1500 but staying above 500 after an initial period. The lower graph shows more pronounced drops in this number, some of which can be associated to observed symbols. This is due to the higher degree of observability of the model including detectable state changes, and thus a larger number of paths becoming invalid throughout the analysis.

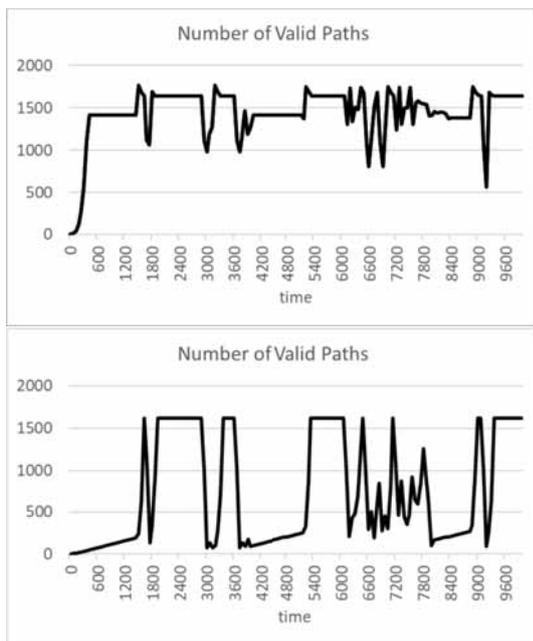


Figure 3: Valid Paths Over Simulation Time Without (Top) And With (Bottom) Detectable State Changes

Figure 4 shows the development of the trace probability in log scaling over time. In both cases, the probability decreases gradually, until the first detectable state change, also in the model without state change detection. In the lower graph, each detectable state change leads to a more noticeable drop in trace probability, since invalidating paths decrease the remaining probability in the analysis.

3.4 Result Discussion

Since the application example presented here is intended only as a proof of concept for the usefulness of

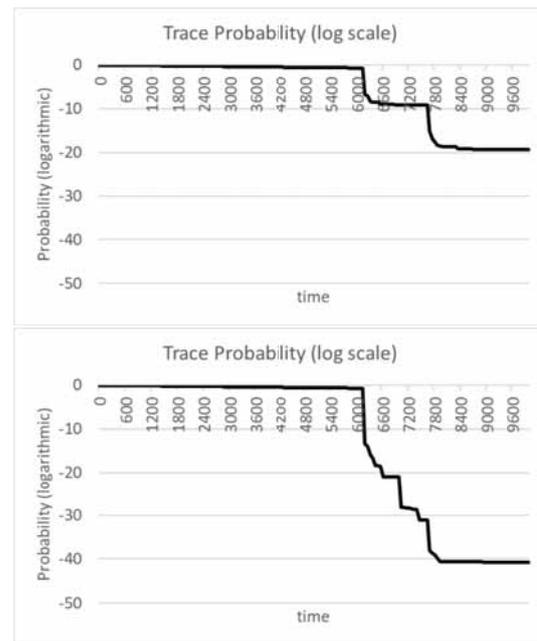


Figure 4: Trace Probability Over Simulation Time Without (Top) And With (Bottom) Detectable State Changes

extended Hybrid-ASPN, improving the results further is future work, and needs more extensive research. Based on these experiments, we can conclude, that behavior reconstruction based on a combination of discrete symbol emissions and samples of continuous measures is possible using the Proxel-based method. Furthermore, is the reconstruction accuracy of the method considerably improved by using the combination of symbols and samples. In the example tested here, the model was too coarse to allow accurate reconstruction only based on samples of the continuous quantity. Only the inclusion of samples resulted in reconstructed paths close to the ground truth. Increasing the models degree of observability through the inclusion of symbols, results in a 5 times faster runtime for the example investigated here. This is due to a smaller number of possible paths that need to be tracked during the analysis, thus resulting in a reduction in state space explosion.

4 Conclusion and Outlook

The paper introduced a user model for virtual stochastic sensors to perform behavior reconstruction of partially observable hybrid systems based on observable output. In contrast to the user models so far, the extended

Hybrid-ASPN enable the modeling of two different types of system output, samples of continuous measures as well as discrete symbols. The new paradigm is demonstrated using a NIALM example problem. The reconstruction experiment shows that being able to utilize both symbols and samples for behavior reconstruction considerably increases the method performance and result accuracy. The increase in performance when including symbols along with sample output shows a way of making virtual stochastic sensors for hybrid systems feasible. Future work must include further investigation of the presented application in NIALM. Complete household models should be tested, as well as a more detailed representation of the state output. This and further real world application scenarios can lead to practically feasible VSS for partially observable hybrid stochastic systems.

References

- [1] Krull C. Virtual Stochastic Sensors: Formal Background and Example Applications Reconstructing the Behavior of Partially Observable Discrete and Hybrid Stochastic Systems. Habilitation thesis, Otto-von-Guericke-University Magdeburg. 2021.
- [2] Krull C, Buchholz R, Horton G. Virtual Stochastic Sensors: How to gain Insight into Partially Observable Discrete Stochastic Systems. In: *The 30th IASTED International Conference on Modelling, Identification and Control, 14th-16th February 2011, Innsbruck, Austria*. 2011; .
- [3] Krull C, Horton G. Hidden Non-Markovian Models: Formalization and Solution Approaches. In: *Proceedings of 6th Vienna Conference on Mathematical Modelling, Vienna, Austria*. 2009; pp. 682–693.
- [4] Buchholz R. Conversive Hidden Non-Markovian Models. Ph.D. thesis, submitted to Otto-von-Guericke-University Magdeburg. 2012.
- [5] Krull C, Horton G. Virtual Stochastic Sensors for Hybrid Systems: Mutual - Influence between Continuous and Discrete System Parts. In: *Proceedings of ASIM 2014 - 22. Symposium Simulationstechnik, Berlin*. 2014; .
- [6] Lazarova-Molnar S. The Proxel-Based Method: Formalisation, Analysis and Applications. Ph.D. thesis, Otto-von-Guericke-University Magdeburg. 2005.
- [7] Bobbio A, Puliafito A, Telek M, Trivedi KS. Recent Developments in Non-Markovian Stochastic Petri Nets. *Journal of Systems Circuits and Computers*. 1998;8(1):119–158.
- [8] Bause F, Kritzinger PS. *Stochastic Petri Nets*. Vieweg. 2002.
- [9] Fink GA. *Markov Models for Pattern Recognition*. Berlin, Heidelberg: Springer. 2008.
- [10] Ciardo G, Blakemore A, Jr PFC, Muppala JK, Trivedi KS. Automated generation and analysis of Markov reward models using stochastic reward nets. In: *Linear Algebra, Markov Chains, and Queueing Models*, pp. 145–191. Springer. 1993; .
- [11] Wolter K, Horton G, German R. Non-Markovian Fluid Stochastic Petri Nets. *Tech. rep.*, TU Berlin. 1996.
- [12] Ciardo G, Nicol D, Trivedi K. Discrete-event simulation of fluid stochastic Petri nets. *IEEE transactions on software engineering*. 1999;25(2).
- [13] Ma H, Yan L, Xia Y, Fu M. *Kalman Filtering and Information Fusion*. Springer, Science Press Beijing. 2020.
- [14] Triantafyllopoulos K. *Bayesian Inference of State Space Models: Kalman Filtering and Beyond*. Springer. 2021.
- [15] Baker S, Mishra A, Irwin D, Cecchet E, Shenoy P, Albrecht J. Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes. In: *Proceedings of SustKDD'12, Beijing, China*. 2012; .
- [16] Krull C, Thiel M, Horton G. Testing Applicability of Virtual Stochastic Sensors for Non-Intrusive Appliance Load Monitoring. In: *Proceeding of the Ninth International Workshop on Practical Applications of Stochastic Modelling*. 2017; .

Modular Platform for Route Guidance in the Cyber-Physical Laboratory Test Field

Marian Göllner*, Sven Jacobitz, Taihao Li, Xiaobo Liu-Henke

Institute of Mechatronics, Ostfalia University of Applied Sciences,
Salzdahlumer Str. 46/48, D-38302 Wolfenbüttel; *mar.goellner@ostfalia.de

SNE 33(1), 2023, 45-52, DOI: 10.11128/sne.33.tn.10637
Selected ASIM SST 2022 Postconf. Publication:2023-02-01;
Received Revised Improved:2023-02-27; Accepted:2023-03-10
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper presents a domain-specific configurable, modular platform for route guidance and trajectory planning (doplar) of intelligent vehicles in different cyber-physical traffic systems, which can be used in projects of different domains. Due to its modular structure, this platform can be used both in road traffic and in the context of Industry 4.0 or Smart Home applications. The big advantage of this platform is that core modules such as the route guidance can be retained and only individual modules, e.g. for wireless communication, must be adjusted. The goal of the entire platform is to plan an optimized vehicle operation according travel time and energy consumption, incorporating dynamic environmental data available from wireless communication within the cyber-physical transport system.

Introduction

A key characteristic of a cyber-physical system is the blurring of boundaries between mechatronic components that communicate via a network infrastructure such as the internet. This steadily increasing degree of networking and functional diversity is leading to increasingly complex, distributed systems with more and more intelligent functions. Such systems are characterized by the integration of a wide variety of components with different requirements for real-time capability, safety and timing, which must all interact reliably within the framework of the CPS. Accordingly, modeling, synthesis as well as validation are complex. Real-time realization and testing in particular requires a lot of effort due to the lack of a real environment and real communication partners [1].

An intelligent vehicle must be capable of performing the four basic tasks of measuring, recognition, planning and execution in order to achieve its intrinsically

set or extrinsically motivated goal. First of all, it must detect (measure) its environment with sensors and process (recognize) the sensor information for environment perception. Based on that, the autonomous system makes decisions about its behavior and its interaction with the environment (planning) and finally realizes these decisions with the help of its actuators to execute the basic movement.

Motivation

At Ostfalia University, several transdisciplinary joint projects are being carried out in which cyber-physical systems are investigated in various domains, such as road traffic, Industry 4.0 or Smart Home.

This paper presents the domain-specific configurable, modular platform for route guidance and trajectory planning (doplar) of automated transportation systems in cyber-physical traffic system, which can be applied to every of these above introduced and further projects in order to develop reuseable functions. The aim of this platform is the planning of an optimized vehicle operation with regard to travel time and energy consumption, including dynamic environmental data from wireless communication within the cyber-physical transport system. The paper will detail on the route guidance and its exemplary application for a cyber-physical laboratory test field for intelligent mobility applications [2] developed at Ostfalia.

Methodology

It is obvious that the development of cyber-physical systems in general or of the platform doplar requires a clearly structured, methodical approach due to the complex and interconnected individual functions. In order to master the overall complexity and the interdisciplinary research and development process in the fields of mechanics, electronics as well as control, information and communication technology, a systematic structuring of the entire mechatronic system is necessary.

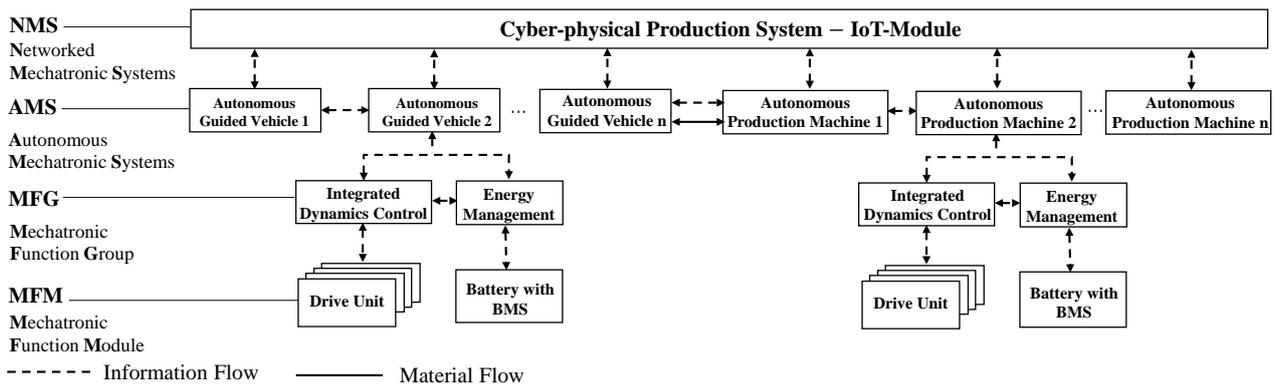


Fig. 1: Mechatronic structuring and hierarchization of a cyber-physical industry 4.0 production plant

With the help of mechatronic structuring according to [1], the entire system is structured hierarchically in a top-down process starting from the main function and subdivided into partial and/or sub-functions. The individual functions are encapsulated in modules with defined interfaces, which can be combined into groups to fulfill higher functionalities. This mechatronic structuring is carried out on four levels:

- **Mechatronic Function Module (MFM):** The lowest hierarchical level is made up of the MFM, consisting of mechatronic systems that cannot be further divided, including the mechanical support structure, sensors, actuators, bus communication and information processing. Each encapsulated MFM has a defined functionality and describes the dynamics of the system.
- **Mechatronic Function Group (MFG):** MFGs are created by coupling several MFMs and adding a higher-level information processing. MFGs enable the implementation of more sophisticated functions by using the subordinate MFMs with their actuators.
- **Autonomous Mechatronic System (AMS):** The overall mechatronic system forms the next hierarchical level of the AMS by combining several MFGs. After processing the available information, set values for subordinate MFGs and MFMs are generated.
- **Networked Mechatronic System (NMS):** If several AMSs are operated side by side, e.g. to process a customer order, a higher level of coordination is required. This coupling with information technology at the highest level is a NMS, or in this case a cyber-physical production system, which corre-

sponds to an industrial 4.0 production line or an industrial 4.0 factory. The product information is managed by the high-level cyber-physical production system and forwarded to all relevant components of the production line, e.g. via WLAN.

Figure 1 exemplary illustrates the modular and hierarchical structure of an autonomous industrial 4.0 production plant consisting of several AGVs and production machines as focussed in the project Synus, which is the basis of the pilot application focused in this article. By adding or exchanging MFM, MFG or AMS, the NMS can be configured as required.

At the lowest level for the AGVs there are four MFMs with smart drive units for the realization of the set forces and moments and one MFM for the power supply, which consists of the battery modules and the battery management, which, among other things, monitors the battery during operation and balances the charge. The integrated dynamics control for controlled vehicle dynamics and the energy management for controlling and monitoring the energy flows are arranged hierarchically above. AGVs and production machines as mechatronic complete systems are arranged hierarchically one level higher than AMS.

The highest level of the NMS is formed by the coupling with information technology of the AGVs and production machines in the production plant. Mechatronic structuring is followed by mechatronic composition. In a bottom-up process, each module, starting with the lowest hierarchy level of the MFM, is designed, validated and successively integrated into the higher-level overall system in a model-based, verification-oriented process using the defined interfaces.

1 State of the Art

The following is an overview of the state of the art with regard to all functions necessary for the route guidance of the doplar platform.

1.1 Mapping

The basis for the route guidance are maps, which can first be divided into geometric and topological maps as well as hybrid intermediate forms. Topological maps are based on graphs and only provide nodes and weightings without directly accessible links to the real world. These maps are used for route planning [3]. Geometric maps, on the other hand, reproduce the environment exactly by projecting it e.g. onto a two-dimensional surface (grid maps such as building ground plans, [4] or three-dimensionally with elevation data as terrain maps, [5]). A special form is created by assigning features to special contours on the map (feature maps, edges and walls, etc., [6], [7]). The result of the environment perception by the vehicle sensors are environment maps, whereby the form of representation depends very much on the sensors used and the output data of the fusion structure. The maps must be converted to topological.

1.2 Route planning

Conventional navigation functions are mainly based on topological maps and rarely directly on geometric maps or hybrid hybrids. Simple approaches for direct navigation on grid maps would be the Manhattan metric [8] or the more centralistic French railway metric [9]. More general are the navigation algorithms based on topological maps using graphs on which the methods of graph theory can be applied. Navigation in this context is only a problem of the shortest paths, if the nodes and weightings of the graph have been reasonably defined in advance. The exact formulation of the problem is crucial to answer the complexity question. If the graph is set without negative weights, the Dijkstra algorithm [10], which is also regarded as the basis of map navigation systems, offers the shortest runtime. If you limit the search field with e.g. the A* algorithm [11], you get even shorter runtimes, but at the expense of the reliable identification of the shortest path. Here it is possible that a shorter way is not found due to the restriction although it can be proven. If negative weights are set up, e.g. to favour certain paths or edges or to reward them for use, the Bellman-Ford algorithm [12] achieves the

shortest runtime. These algorithms all start at a start point and propagate to the end point. As soon as this is reached, the algorithm usually ends, since this is the shortest path as the abort criterion.

This is different with the algorithms of [13] and [14], which are founded on the work of [15] and are each based on finding the shortest paths between all node pairs. The approach here is that if the route between any two points is to be retrived, the individual partial paths of this route are already minimal in themselves. If the shortest paths between the respective points are known, the shortest path is composed of the already known shortest paths of the partial paths. In the ideal case, even the shortest path between the searched start and end points is already included. These methods are particularly suitable for static problems, since an initial high calculation effort is necessary, but does not have to be performed a second time, whereas they are less suitable for dynamic applications in traffic.

2 Concept of Doplar

The aim of doplar is to plan optimized vehicle operation by route and trajectory planning with regard to journey duration and energy consumption, taking into account dynamic environmental data from wireless communication and vehicle and environmental sensors.

A problem with current approaches is the largely specific development of individual functions for certain vehicles in defined domains without taking into account a transfer to related domains, which results in a high, double development effort. This problem results in the essential requirement that the functions presented in this paper should be used for different domains without much effort. The functional structure proposed to achieve the goal and the solution of the problem, the domain-specific configurable modular platform for route guidance and trajectory planning of intelligent vehicles (doplar), is shown in Figure 2 .

In order to achieve the goal of optimized vehicle operation, a route guidance function is required to determine an optimal route, which is then planned out by the trajectory generation for realization. A human-machine interface (HMI) is required for route guidance, e.g. to enable vehicle occupants or other users to set targets. Optionally, fleet management could also intervene in route planning in order to coordinate several vehicles. For the route guidance, mapping is necessary, which provides map data in the correct form and updates it,

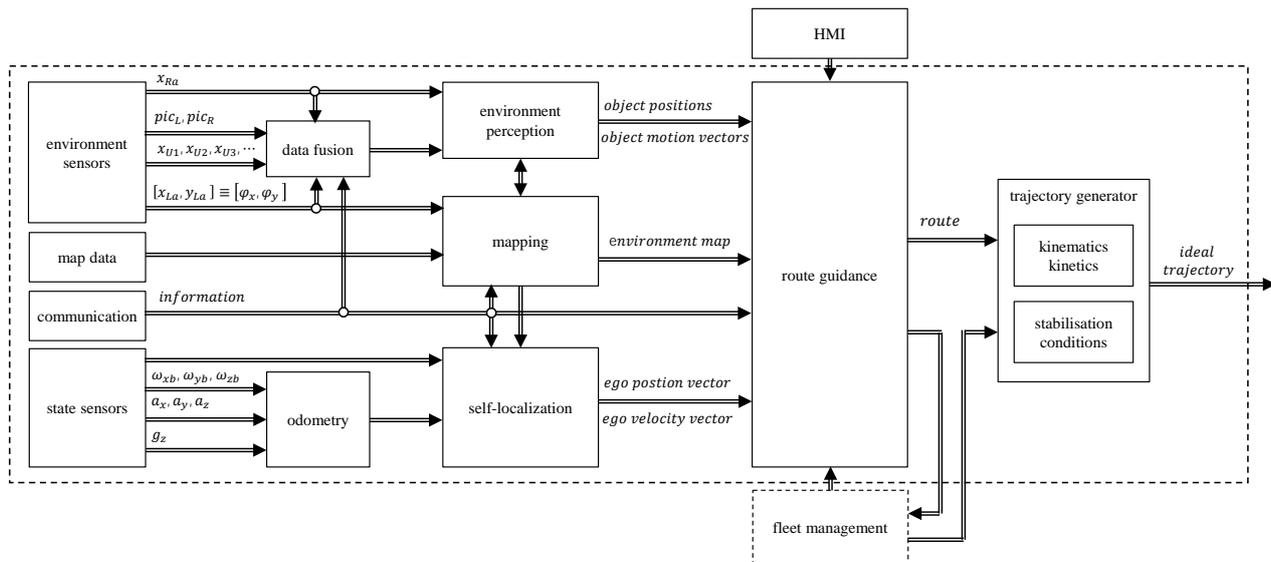


Fig. 2: Concept of the domain-specific configurable modular platform for route guidance and trajectory planning of intelligent vehicles (doplar)

self-localization to determine the current position, environment perception to determine the driving environment and possibly deviating map data and communication, which enables an exchange of information between the ego vehicle and the environment.

All these functions are encapsulated in different modules with defined interfaces, so that the exchange of individual modules is possible as long as the interfaces are maintained. This ensures that the platform can be used in different domains, since domain-specific modules, e.g. for environment perception, can be easily exchanged without having to change core algorithms such as route guidance.

3 Design of Doplar’s Route Guidance

This chapter describes the design of the functions for providing map data and route guidance of doplar.

3.1 Mapping

The mapping module serves to provide and update the map material as a necessary basis for route guidance. The map data has to be described mathematically as graph $G = (N, E)$, which consists of a finite set N of nodes (x, y) and a finite set E of weighted edges. A node is a point with fixed coordinates in x and y posi-

tion and an edge is the connection between two nodes. The edges thus correspond to road segments that are connected to each other by the nodes. The origin of the map data can also be domain-specific: For applications in road traffic, for example, the OpenStreetMap can be used, whereas for applications in Industry 4.0 plants or the Smart Home floor plans can be converted into graphs. A further possibility for generating or updating the map material is the use of a SLAM algorithm, which evaluates the vehicle’s environment sensors and provides information about its environment.

The map forced in this paper is a geometrical (G) as well as a topological (T) hybrid form, a hybrid G+T map, which links a 2.5 dimensional grid map with a graph map and thus enables real-time, event-based on-line navigation. This map goes back to [16] and was adapted in the context of this work. The map is generated by splitting the fused sensor data in the vertical axis and projecting them onto a grid on the position of the vehicle’s center of gravity. The following Figure 3 shows the projection on the grid map with a corresponding resolution. This map is now only insufficiently suitable to carry out the route guidance. Only an assignment of each individual grid cross as node and the execution of the Manhattan metric would be conceivable. Since this follows fixed paths, i.e. the grid, the vehicle would roll in x - and y - direction and possibly not find the shortest way, because nodes cannot be skipped.

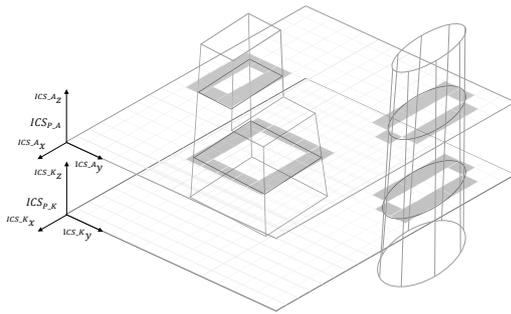


Fig. 3: Creating a 2.5-dimensional, geometric map from data of the environment perception

A logical improvement is the definition of a graph map based on the grid map by transforming the grids into nodes and connecting the nodes with weighted edges depending on the actual distances from each other. Thus simple diagonal journeys are possible. An optimal route, however, still cannot be found under the premise that the grid is evenly distributed, since the paths always run at a minimum of 45° .

Therefore, the approach chosen sorts only characteristic points to the graph map, namely those corners which describe the maximum extent of an object. The edges of the obstacles and objects are detected and pre-sorted according to relevance using the global target vector. The distance of these points can be determined by means of Euclidean distance and taken over as weighted edges between the nodes. Edges that are covered by an object get a very large weighting factor and are therefore disadvantaged by the navigation algorithm. The new map (Figure 4) can now be treated as a pure graph map and can be traced back to the grid sectors of the grid map by assigning the nodes.

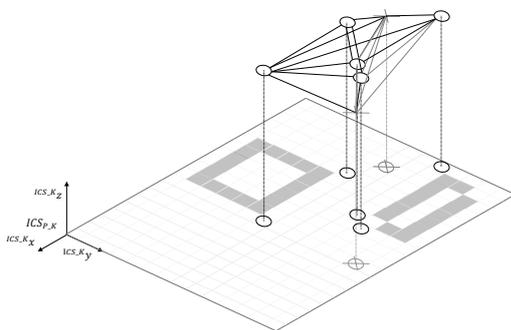


Fig. 4: Creating a topological graph map based on a 2.5-dimensional geometric map

3.2 Route guidance

The route guidance module used according to [17] aims at finding an optimal route from a start point to a destination point in a directed graph G , which describes the traffic network, related to a defined quality criterion. Route guidance is based on Dijkstra's algorithm, which belongs to the methods of width search. Using the cost function

$$J_i = g_s \cdot s_i + g_t \cdot t_i + g_E \cdot E_i \quad (1)$$

and the weighting factors g_s , g_t and g_E , the costs J of each edge i are calculated from the information of the edge weights about distance s , duration t and energy consumption E between two nodes. Dijkstra's algorithm always converges to the optimal route if the graph does not contain loops or negative edge costs.

The total cost of each node is initialized to infinity with the start of the algorithm except for the start node, which is initialized with zero. The algorithm determines the costs of the unvisited neighboring nodes from the start node as the sum of its own costs plus the costs of the connecting edge and updates the total costs of the neighboring nodes if the recalculated costs are less than the previous costs. All considered neighbor nodes are added to a waiting list, the start node is marked as visited and the node with the lowest cost is selected from the waiting list. Starting from this node, the procedure described is repeated until the destination node is reached with minimal total costs (see Figure 5).

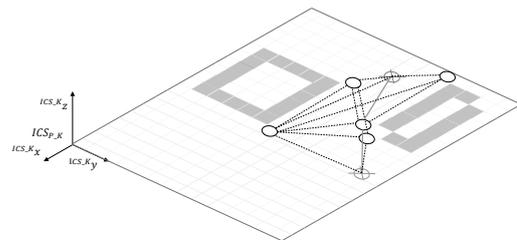


Fig. 5: Result of Dijkstra's algorithm for finding the optimal route in a topological graph map

The result of minimizing the total costs from start to destination is the route r , which consists of nodes connected by edges:

$$r = \min_{n \in N} \sum_{i=start}^{n=dest} J_i \quad (2)$$

The integration of dynamic information from wireless communication in the cyber-physical traffic system is realized by a temporary adaptation of the edge weights in the graph. For this purpose, the position of the message is first used to determine which edge is to be assigned the information. The edge weights are then changed according to the type of information. If, for example, a closed road occurs as a result of road works, the distance, duration and energy consumption for the corresponding edge are set to infinity, so that the edge can no longer be part of the optimum route regardless of the weighting factors. If, on the other hand, a delay occurs due to congested traffic, only the duration and energy consumption of the edge are changed, since the length of the route does not change.

3.3 Trajectory planning

Trajectory planning is used to generate target polynomials for longitudinal and lateral dynamics, which are then passed on as set points to the dynamic control of an AGV in order to follow the optimum route determined in the route guidance module and execute the transport order. The return value from the route guidance is the already optimized group \underline{r} of points (x_i, y_i) in order of the direction to be departed:

$$\underline{r} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} : x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_k; i = 0 \dots k \quad (3)$$

In order to take into account the ideally shortest path, which consists only of straight lines between the nodes, additional conditions are placed on the course of the polynomial. By inserting support nodes $j \supset ia$ a closer approximation to the optimale route is made possible. Defining start and end angles at which the polynomial runs in, guarantees an initially linear acceleration and deceleration.

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = j \cdot \begin{cases} \begin{bmatrix} x_i \\ y_i \end{bmatrix} & j \pmod 2 = 1 \\ \begin{bmatrix} \frac{x_{i+1} - x_i}{2} + x_i \\ \frac{y_{i+1} - y_i}{2} + y_i \end{bmatrix} & j \pmod 2 \neq 1 \end{cases} \quad (4)$$

The nodes and weighted edges of the optimal route are used to generate trajectories. Figure 6 shows the generated trajectory. As already described in chapter 2, these trajectories are polynomials of n^{th} order depend-

ing on the nodes obtained from the route guidance.

$$p'_i \begin{pmatrix} x_j \\ y_j \end{pmatrix} \doteq p'_{i-1} \begin{pmatrix} x_j \\ y_j \end{pmatrix}; p''_i \begin{pmatrix} x_j \\ y_j \end{pmatrix} \doteq p''_{i-1} \begin{pmatrix} x_j \\ y_j \end{pmatrix} \quad (5)$$

First the mathematical boundary conditions for the construction, in this case the natural boundary suitability under the minimization of the computation effort, are to be included.

$$p''_0 \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = 0; p''_{n-1} \begin{pmatrix} x_n \\ y_n \end{pmatrix} = 0 \quad (6)$$

But the shortest way is not the fastest way in most cases. In order to adapt the trajectory to this, an optimization is carried out with the help of a simple dynamic model of the vehicle. The main goal is to follow the trajectory as fast as possible, which results in deviations that in their entirety represent a fast path along the same support points.

The trajectory found in this way is transferred to the underlying model predictive trajectory control, presented e.g. in [18] and [19]. It works cascaded and transfers its setpoints to subordinate dynamic controllers, which influence the system and thus let it follow the calculated trajectory.

4 Demonstration of the Platform Doplar

This chapter shows the realization of the doplar platform on the basis of a pilot application in the domain of industry 4.0, which is focussed in the project Synus, and exemplary results.

4.1 Pilot application in the domain of Industry 4.0

In this subchapter an overview of the AGV's used in the cyber-physical laboratory test field is presented first. Figure 5 shows the system structure of the considered AGV. The vehicle sensors determine vehicle conditions and environmental information, which are used by the central information processing. The information processing also evaluates information from wireless communication and generates messages for the environment if necessary. With the help of the collected information, control algorithms are executed, which provide the set points of the vehicle actuators and are realized by them, so that the functionality of the AGV is accomplished.

On the vehicle side, the AGV has four electric drives close to the wheels, which are powered by a battery.

The realization of the AGV is shown in Figure 6. Mecanum wheels are used to transmit the drive torque to the road surface in order to perform omnidirectional driving maneuvers. Compared to transport vehicles with conventional wheels, the AGV does not require any maneuvering space and rotations around the vertical axis can be realized (Technologie-Netzwerk, 2016). Taking into account the environment-friendliness of the drive, the AGV's power supply is provided by a battery pack. The concept of active load handling is achieved by a conveyor belt with a height-adjustable lifting system.

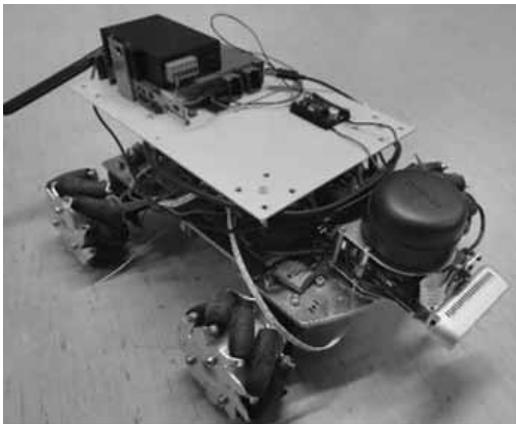


Fig. 6: Realization of the AGV

In the pilot application focussed in this paper, a spare part for a vehicle test bench (bottom left) shall be collected from an employee (top right). The AGVs are to jointly award this transport order by evaluating their optimal routes in order to achieve a minimum transport time. An optimal trajectory is to be planned for the chosen AGV to fulfill the transport order.

4.2 Exemplary results

First, the doplar was used to generate a graph from the floor plan, which enables route guidance. The fleet management initially initiates the route guidance for the three AGVs. On the basis of the routes generated, the time to complete the order is determined so that the AGV with the shortest transport time can be selected for the order. A result of the route guidance system is shown in Figure 8. The three determined routes, which also seem optically plausible, are recognizable. AGV

3 has the shortest route to the order location and therefore also requires the shortest transport time so that this AGV should be selected for the transport order.

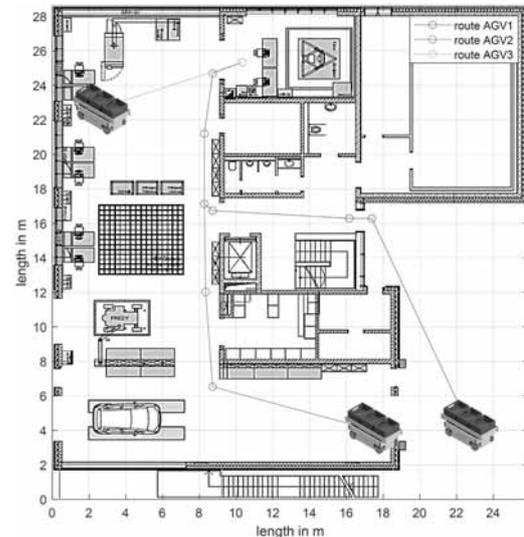


Fig. 7: Result of the route guidance for the three AGVs as basis for the fleet management for the distribution of the transport order

This pilot application shows exemplarily the functionality of the platform doplar from generating a graph map from a floor plan to route guidance. further examples are concluded in [2].

5 Conclusion and Outlook

In this article the domain-specific configurable modular platform for route guidance and trajectory planing of intelligent vehicles (doplar) was presented, which can be used in different projects across domains. The great advantage of this platform is that core components such as the navigation and guidance algorithm can be retained and only individual modules, e.g. for wireless communication, have to be adapted. The aim of the entire platform is to plan optimized vehicle operation with regard to journey time and energy consumption, taking into account dynamic environmental data available from wireless communication within the cyber-physical transport system. The platform specifications serve as target values for subordinate systems such as integrated vehicle dynamics control or for awarding an order in an industrial 4.0 production plant. The doplar platform was demonstrated in a pilot application for an industrial 4.0 production plant in the context of the project Synus, in

which three AGVs can be used for transport tasks. The fleet management initiates a target guidance of all available AGVs, which forms the basis for the decision to award a contract. The AGVs exchange their forecasted routes with each other and jointly allocate the order. For the AGV that has received the order, doplar generates an optimal trajectory for its completion. In the following work steps, the developed platform doplar is to be integrated into other domains and real systems, examined under real-time conditions and further optimized.

Acknowledgement

This contribution evolved from the research project Model-based Conception and Evaluation of Industry 4.0 Solutions for Networking Mechatronic Components in Production Facilities by Digitalization (MIMec) which is part of the collaborative project Methods and Tools for the Synergetic Conception and Evaluation of Industry 4.0 Technologies (Synus). It is funded by the European Regional Development Fund (EFRE: ZW 6 85012454).

References

- [1] Liu-Henke X. *Mechatronische Entwicklung der aktiven Feder-/Neigetchnik für das Schienenfahrzeug RailCab: Zugl.: Paderborn, Univ., Diss., 2005*, vol. 589 of *Fortschritt-Berichte VDI Reihe 12, Verkehrstechnik/Fahrzeugtechnik*. Düsseldorf: VDI-Verl., als ms. gedr ed. 2005.
- [2] Jacobitz S, Göllner M. Seamless validation of cyber-physical systems under real-time conditions by using a cyber-physical laboratory test field. In: *IEEE International Conference on Recent Advances in Systems Science and Engineering*. 2021; .
- [3] Kortenkamp D, Weymouth T. Topological Mapping for Mobile Robots Using a Combination of Sonar and Vision Sensing. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2)*, AAAI'94. USA: American Association for Artificial Intelligence. 1994; pp. 979–984.
- [4] Thrun S, Burgard W, Fox D. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Machine Learning*. 1998;31(1/3):29–53.
- [5] Kweon IS, Kanade T. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1992; 14(2):278–292.
- [6] Neira J, Tardos JD. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*. 2001; 17(6):890–897.
- [7] Leonard JJ, Durrant-Whyte HF. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*. 1991; 7(3):376–382.
- [8] Benkert M, Widmann F, Wolff A. The Minimum Manhattan Network Problem: A Fast Factor-3 Approximation.
- [9] Walter R. Metrische Räume. In: *Einführung in die Analysis*, edited by Walter R, pp. 270–327. Berlin: de Gruyter. 2007;.
- [10] Dijkstra EW. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959;1(1):269–271.
- [11] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*. 1968;4(2):100–107.
- [12] Bellman R. On a routing problem. *Quarterly of Applied Mathematics*. 1958;16(1):87–90.
- [13] Floyd RW. Algorithm 97: Shortest path. *Communications of the ACM*. 1962;5(6):345.
- [14] Warshall S. A Theorem on Boolean Matrices. *Journal of the ACM*. 1962;9(1):11–12.
- [15] Kleene SC. Representation of Events in Nerve Nets and Finite Automata. In: *Automata Studies. (AM-34)*, edited by Shannon CE, McCarthy J, Annals of Mathematics Studies, pp. 3–42. Princeton, NJ: Princeton University Press. 1956;.
- [16] Golfarelli M, Maio D, Rizzi S. Elastic correction of dead-reckoning errors in map building. In: *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 2. 1998; pp. 905–911 vol.2.
- [17] Scherler S, Liu-Henke X, Henke M. Predictive Energy Management for an Electric Vehicle with Fuel Cell Range Extender in Connected Traffic Systems. In: *2020 19th International Conference on Mechatronics - Mechatronika (ME)*. IEEE. 02.12.2020 - 04.12.2020; pp. 1–8.
- [18] Göllner M, Zhang J, Liu-Henke X. Model predictive trajectory control for automated driving of a spherical electrical drive. In: *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE. 10/1/2018 - 10/3/2018; pp. 1–6.
- [19] Göllner M. Fusion structure for environment perception and model-predictive navigation of autonomous vehicles. 25.06.2019.



EUROSIM Data and Quick Info

	 Dutch Benelux Simulation Society	EUROSIM CONGRESS 2023 July 3-5, 2023, Amsterdam, The Netherlands www.eurosim2023.eu
		ASIM SPL 2023 20 th ASIM Dedicated Conference on Simulation in Production and Logistics September 13-15, 2023, Ilmenau, Germany www.asim-fachtagung-spl.de
		I3M 2023 International Multidisciplinary Modeling & Simulation Multiconference September 18-20, 2023, Athens, Greece www.msc-les.org/i3m2023 Winter Simulation Conference 2023 , December 10-13, 2023, San Antonio, TX, USA www.wintersim.org
	UKSIM 2023 April 12-14, 2023, Cambridge uksim.info	PTSK 2023 – 27. Int. Workshop May 25-27, 2023, Zakopane www.ptsk.pl
		SIMS 2023 – 64. Int. Conference September 26-27, 2023, Västerås www.scansims.org



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 modelling and simulation in industry, research, and development – by publication and conferences.

www.eurosim.info

EUROSIM members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation.

Full Members are ASIM, CEA-SMSG, CSSS, DBSS, KASIM, LIOPHANT, LSS, PTSK, NSSM, SIMS, SLOSIM, UKSIM. Observer Members are ALBSIM and ROMSIM. Former Members (societies in re-organisation) are: CROSSIM, FRANCOSIM, HSS, ISCS.

EUROSIM is governed by a **Board** consisting of one representative of each member society, president, past president, and SNE representative.

President	M. Mujica Mota (DBSS), m.mujica.mota@hva.nl
Past President	Emilio Jiménez (CEA-SMSG), emilio.jimenez@unirioja.es
Secretary	P. M. Scala (DBSS), p.m.scala@hva.nl
Treasurer	Felix Breitenecker (ASIM) felix.breitenecker@tuwien.ac.at
Webmaster	Irmgard Husinsky (ASIM), irmgard.husinsky@tuwien.ac.at

SNE SNE – Simulation Notes Europe is EUROSIM's membership journal with peer reviewed scientific contributions about all areas of modelling and simulation, including new trends as big data, cyber-physical systems, etc.

The EUROSIM societies distribute e-SNE in full version to their members as official membership journal. The basic version of e-SNE is available with open access. Publishers are EUROSIM, ARGESIM and ASIM,

www.sne-journal.org

office@sne-journal.org

SNE-Editor: Felix Breitenecker (ASIM)
felix.breitenecker@eic@sne-journal.org

EUROSIM Congress and Conferences

Each year a major EUROSIM event take place, as the EUROSIM CONGRESS organised by a member society, SIMS EUROSIM Conference, and MATHMOD Vienna Conference (ASIM).

EUROSIM Congress 2023, the 11th EUROSIM Congress, will be organised by DBSS, the Dutch Benelux simulation society, in Amsterdam, July 3-5, 2023:

www.eurosim2023.eu

Furthermore, EUROSIM Societies organize local conferences, and EUROSIM co-operates with the organizers of I3M Conference and WinterSim Conference Series:

www.msc-les.org/i3m2023

www.wintersim.org/



EUROSIM Member Societies



ASIM German Simulation Society Arbeitsgemeinschaft Simulation

ASIM is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria.

President	Felix Breiteneker, <i>felix.breiteneker@tuwien.ac.at</i>
Vice President	Sigrid Wenzel, <i>s.wenzel@uni-kassel.de</i> Thorsten Pawletta, <i>thorsten.pawletta@hs-wismar.de</i> Andreas Körner, <i>andreas.koerner@tuwien.ac.at</i>

ASIM is organising / co-organising the following international conferences: ASIM SPL Int. Conference 'Simulation in Production and Logistics' (biannual), ASIM SST 'Symposium Simulation Technique' (biannual), MATHMOD Int. Vienna Conference on Mathematical Modelling (triennial). Furthermore, ASIM is co-sponsor of WSC - Winter Simulation Conference and of the *I3M* and conference series.

ASIM Working Committees

GMMS: Methods in Modelling and Simulation

U. Durak, *umut.durak@dlr.de*

SUG: Simulation in Environmental Systems

J. Wittmann, *wittmann@informatik.uni-hamburg.de*

STS: Simulation of Technical Systems

W. Commerell, *commerell@hs-ulm.de*

SPL: Simulation in Production and Logistics

S. Wenzel, *s.wenzel@uni-kassel.de*

EDU: Simulation and Education

A. Körner, *andreas.koerner@tuwien.ac.at*

Working Group Big Data: Data-driven Simulation in

Life Sciences, N. Popper, *niki.popper@dwh.at*

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

Contact Information

www.asim-gi.org

info@asim-gi.org, *admin@asim-gi.org*

ASIM – Inst. of Analysis and Scientific Computing,
TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna,
Austria

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control. The association is divided into national thematic groups, one of which is centered on Modeling, Simulation and Optimization (CEA-SMSG).

President	José L. Pitarch, <i>jlpitarch@isa.upv.es</i>
Vice President	Juan Ignacio Latorre, <i>juanignacio.latorre@unavarra.es</i>

Contact Information

www.ceautomatica.es/grupos/

simulacion@cea-ifac.es

CEA-SMSG / Emilio Jiménez, Department of Electrical Engineering, University of La Rioja, San José de Calasanz 31, 26004 Logroño (La Rioja), Spain



CSSS – Czech and Slovak Simulation Society

CSSS is the Simulation Society with members from the two countries: Czech Republic and Slovakia. The CSSS history goes back to 1964.

President	Michal Štepanovský <i>michal.stepanovsky@fit.cvut.cz</i>
Vice President	Mikuláš Alexík, <i>alexik@frtk.fri.utc.sk</i>

Contact Information

cssim.cz

michal.stepanovsky@fit.cvut.cz

CSSS – Český a Slovenský spolek pro simulaci systémů, Novotného lávka 200/5, 11000 Praha 1, Česká republika



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.

President	M. Mujica Mota, <i>m.mujica.mota@hva.nl</i>
Vice President	A. Heemink, <i>a.w.heemink@its.tudelft.nl</i>
Secretary	P. M. Scala, <i>p.m.scala@hva.nl</i>



Contact Information

www.DutchBSS.org

a.w.heemink@its.tudelft.nl

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

KA-SIM Kosovo Simulation Society

The Kosova Association for Modeling and Simulation (KA-SIM) is closely connected to the University for Business and Technology (UBT) in Kosovo.

President	Edmond Hajrizi, ehajrizi@ubt-uni.net
Vice President	Muzafer Shala, info@ka-sim.com

Contact Information

www.ubt-uni.net

ehajrizi@ubt-uni.net

Dr. Edmond Hajrizi
Univ. for Business and Technology (UBT)
Lagjja Kalabria p.n., 10000 Prishtina, Kosovo



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, courses and internships focused on M&S applications.

President	A.G. Bruzzone, agostino@itim.unige.it
Director	E. Bocca, enrico.bocca@liophant.org

Contact Information

www.liophant.org

info@liophant.org

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

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.

President	Yuri Merkuryev, merkur@itl.rtu.lv
Vice President	Egils Ginters, egils.ginters@rtu.lv

Contact Information

www.itl.rtu.lv/imb/

merkur@itl.rtu.lv, Egils.Ginters@rtu.lv

Prof. Yuri Merkuryev, Dept. of Modelling and Simulation, Riga Technical University, Kalku street 1, Riga, LV-1658, Latvia



NSSM – National Society for Simulation Modelling (Russia)

NSSM – The National Society for Simulation Modelling (Национальное Общество Имитационного Моделирования – НОИМ) was officially registered in Russia in 2011.

President	R. M. Yusupov, yusupov@iias.spb.su
Chairman	A. Plotnikov, plotnikov@sstc.spb.ru

Contact Information

www.simulation.su

yusupov@iias.spb.su

NSSM / R. M. Yusupov, St. Petersburg Institute of Informatics and Automation RAS, 199178, St. Petersburg, 14th line, h. 39

PTSK – Polish Society for Computer Simulation

PTSK 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.

President	Tadeusz Nowicki, Tadeusz.Nowicki@wat.edu.pl
Vice President	Leon Bobrowski, leon@ibib.waw.pl

Contact Information

www.ptsk.pl

leon@ibib.waw.pl

PSCS, 00-908 Warszawa 49, ul. Gen. Witolda Urbanowicza 2, pok. 222



SIMS – Scandinavian Simulation Society

SIMS is the Scandinavian Simulation Society with members from the five Nordic countries Denmark, Finland, Norway, Sweden and Iceland. The SIMS history goes back to 1959.

President	Tiina Komulainen, <i>tiina.komulainen@oslomet.no</i>
Vice President	Erik Dahlquist, <i>erik.dahlquist@mdh.se</i>

Contact Information

www.scansims.org

vadime@wolfram.com

Vadim Engelson, Wolfram MathCore AB,
Teknikringen 1E, 58330, Linköping, Sweden



SLOSIM – Slovenian Society for Simulation and Modelling

The Slovenian Society for Simulation and Modelling was established in 1994. It promotes modelling and simulation approaches to problem solving in industrial and in academic environments by establishing communication and cooperation among corresponding teams.

President	Goran Andonovski, <i>goran.andonovski@fe.uni-lj.si</i>
Vice President	Božidar Šarler, <i>bozidar.sarler@fs.uni-lj.si</i>

Contact Information

www.slosim.si

slosim@fe.uni-lj.si, vito.logar@fe.uni-lj.si

SLOSIM, Fakulteta za elektrotehniko, Tržaška 25,
SI-1000, Ljubljana, Slovenija

UKSIM - United Kingdom Simulation Society

The UK Modelling & Simulation Society (UKSim) is the national UK society for all aspects of modelling and simulation, including continuous, discrete event, software and hardware.

President	David Al-Dabass, <i>david.al-dabass@ntu.ac.uk</i>
Secretary	T. Bashford, <i>tim.bashford@uwtsd.ac.uk</i>

Contact Information

uksim.info

david.al-dabass@ntu.ac.uk

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

Observer Members

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.

Contact Information

florin_h2004@yahoo.com

ROMSIM / Florin Hartescu, National Institute for Research in Informatics, Averescu Av. 8 – 10, 011455 Bucharest, Romania

ALBSIM – Albanian Simulation Society

The Albanian Simulation Society has been initiated at the Department of Statistics and Applied Informatics, Faculty of Economy at the University of Tirana, by Prof. Dr. Kozeta Sevrani.

Contact Information

kozeta.sevrani@unitir.edu.al

Albanian Simulation Goup, attn. Kozeta Sevrani, University of Tirana, Faculty of Economy, rr. Elbasanit, Tirana 355, Albania

Former Societies / Societies in Re-organisation

- CROSSIM – Croatian Society for Simulation Modelling
Contact: Tarzan Legović, *Tarzan.Legovic@irb.hr*
- FrancoSim – Société Francophone de Simulation
- HSS – Hungarian Simulation Society
Contact: A. Gábor, *andrasi.gabor@uni-bge.hu*
- ISCS – Italian Society for Computer Simulation

The following societies have been formally terminated:

- MIMOS – Italian Modeling & Simulation Association; terminated end of 2020.



Association Simulation News



SNE – Simulation Notes Europe

SNE

ARGESIM is a non-profit association generally aiming for dissemination of information on system simulation – from research via development to applications of system simulation. ARGESIM is closely co-operating with EUROSIM, the Federation of European Simulation Societies, and with ASIM, the German Simulation Society.

ARGESIM is an 'outsourced' activity from the *Mathematical Modelling and Simulation Group* of TU Wien, there is also close co-operation with TU Wien (organisationally and personally).

→ www.argesim.org

→ office@argesim.org

→ ARGESIM/Math. Modelling & Simulation Group,
Inst. of Analysis and Scientific Computing, TU Wien
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria
Attn. Prof. Dr. Felix Breitenecker

ARGESIM is following its aims and scope by the following activities and projects:

- Publication of the scientific journal *SNE – Simulation Notes Europe* (membership journal of EUROSIM, the *Federation of European Simulation Societies*) – www.sne-journal.org
- Organisation and Publication of the ARGESIM Benchmarks for *Modelling Approaches and Simulation Implementations*
- Publication of the series ARGESIM Reports for monographs in system simulation, and proceedings of simulation conferences and workshops
- Publication of the special series *FBS Simulation – Advances in Simulation / Fortschrittsberichte Simulation* - monographs in co-operation with ASIM, the German Simulation Society
- Support of the Conference Series *MATHMOD Vienna* (triennial, in co-operation with EUROSIM, ASIM, and TU Wien) – www.mathmod.at
- Administration of ASIM (German Simulation Society) and administrative support for EUROSIM www.eurosim.info
- Simulation activities for TU Wien

ARGESIM is a registered non-profit association and a registered publisher: ARGESIM Publisher Vienna, root ISBN 978-3-901608-xx-y and 978-3-903347-xx-y, root DOI 10.11128/z...zz.zz. Publication is open for ASIM and for EUROSIM Member Societies.

The scientific journal *SNE – Simulation Notes Europe* provides an international, high-quality forum for presentation of new ideas and approaches in simulation – from modelling to experiment analysis, from implementation to verification, from validation to identification, from numerics to visualisation – in context of the simulation process. *SNE* puts special emphasis on the overall view in simulation, and on comparative investigations.

Furthermore, *SNE* welcomes contributions on education in/for/with simulation.

SNE is also the forum for the ARGESIM Benchmarks on *Modelling Approaches and Simulation Implementations* publishing benchmarks definitions, solutions, reports and studies – including model sources via web.

SNE Editorial Office /ARGESIM

→ www.sne-journal.org

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

Johannes Tanzler (Layout, Organisation)
Irmgard Husinsky (Web, Electronic Publishing)
Felix Breitenecker EIC (Organisation, Authors)

ARGESIM/Math. Modelling & Simulation Group,
Inst. of Analysis and Scientific Computing, TU Wien
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

SNE, primarily an electronic journal, follows an open access strategy, with free download in a basic version (B/W, low resolution graphics). *SNE* is the official membership journal of EUROSIM, the *Federation of European Simulation Societies*. Members of (most) EUROSIM Societies are entitled to download the full version of e-*SNE* (colour, high-resolution graphics), and to access additional sources of benchmark publications, model sources, etc. (group login for the 'publication-active' societies; please contact your society). Furthermore, *SNE* offers EUROSIM Societies a publication forum for post-conference publication of the society's international conferences, and the possibility to compile thematic or event-based *SNE* Special Issues.

Simulationists are invited to submit contributions of any type – *Technical Note*, *Short Note*, *Project Note*, *Educational Note*, *Benchmark Note*, etc. via *SNE*'s website:

→ www.sne-journal.org,



ASIM



ASIM



ASIM



ASIM Books – ASIM Book Series – ASIM Buchreihen

Monographs / Proceedings

- Proceedings Langbeiträge ASIM Workshop 2023 STS/GMMS/EDU - ASIM Fachgruppenworkshop 2023**
Univ. Magdeburg, März 2023; C. Krull; W. Commerell, U. Durak, A. Körner, T. Pawletta (Hrsg.)
ARGESIM Report 21; ASIM Mitteilung 185; ISBN ebook 978-3-903347-61-8, DOI 10.11128/arep.21, ARGESIM Verlag, Wien, 2023
- Kurzbeiträge & Abstract-Beiträge ASIM Workshop 2023 STS/GMMS/EDU - ASIM Fachgruppenworkshop 2023**
Univ. Magdeburg, März 2023; C. Krull; W. Commerell, U. Durak, A. Körner, T. Pawletta (Hrsg.)
ARGESIM Report 22; ASIM Mitteilung 186; ISBN ebook 978-3-903347-62-5, DOI 10.11128/arep.22, ARGESIM Verlag, Wien, 2023
- Proceedings Langbeiträge ASIM SST 2022 -26. ASIM Symposium Simulationstechnik, TU Wien, Juli 2022**
F. Breitenecker, C. Deatcu, U. Durak, A. Körner, T. Pawletta (Hrsg.), ARGESIM Report 20; ASIM Mitteilung AM 181
ISBN ebook 978-3-901608-97-1, DOI 10.11128/arep.20, ARGESIM Verlag Wien, 2022; ISBN print 978-3-903311-19-0, TU Verlag
- Proceedings Kurzbeiträge ASIM SST 2022 -26. ASIM Symposium Simulationstechnik, TU Wien, Juli 2022**
F. Breitenecker, C. Deatcu, U. Durak, A. Körner, T. Pawletta (Hrsg.), ARGESIM Report 19; ASIM Mitteilung AM 179
ISBN ebook 978-3-901608-96-4, DOI 10.11128/arep.19, ISBN print 978-3-901608-73-5, ARGESIM Verlag Wien, 2022
- Simulation in Production and Logistics 2021 – 19. ASIM Fachtagung Simulation in Produktion und Logistik**
Online Tagung, Sept. 2021, J. Franke, P. Schuderer (Hrsg.), Cuvillier Verlag, Göttingen, 2021,
ISBN print 978-3-73697-479-1; ISBN ebook 978-3-73696-479-2; ASIM Mitteilung AM177
- Proceedings ASIM SST 2020 – 25. ASIM Symposium Simulationstechnik, Online-Tagung**
14.-15.10.2020; C. Deatcu, D. Lückerrath, O. Ullrich, U. Durak (Hrsg.), ARGESIM Verlag Wien, 2020;
ISBN ebook: 978-3-901608-93-3; DOI 10.11128/arep.59; ARGESIM Report 59; ASIM Mitteilung AM 174

Book Series Fortschrittsberichte Simulation – Advances in Simulation

- Cooperative and Multirate Simulation: Analysis, Classification and New Hierarchical Approaches.** I. Hafner, FBS 39
ISBN ebook 978-3-903347-39-7, DOI 10.11128/fbs.39, ARGESIM Publ. Vienna, 2022; ISBN print 978-3-903311-07-7, TUVerlag Wien, 2022
- Die Bedeutung der Risikoanalyse für den Rechtsschutz bei automatisierten Verwaltungsstrafverfahren.** T. Preiß, FBS 38
ISBN ebook 978-3-903347-38-0, DOI 10.11128/fbs.38, ARGESIM Publ. Vienna, 2020; ISBN print 978-3-903311-14-5, TUVerlag Wien, 2020
- Methods for Hybrid Modeling and Simulation-Based Optimization in Energy-Aware Production Planning.** B. Heinzl, FBS 37
ISBN ebook 978-3-903347-37-3, DOI 10.11128/fbs.37, ARGESIM Publ. Vienna, 2020; ISBN print 978-3-903311-11-4, TUVerlag Wien, 2020
- Konforme Abbildungen zur Simulation von Modellen mit verteilten Parametern.** Martin Holzinger, FBS 36
ISBN ebook 978-3-903347-36-6, DOI 10.11128/fbs.36, ARGESIM Publ. Vienna, 2020; ISBN print 978-3-903311-10-7, TUVerlag Wien, 2020
- Fractional Diffusion by Random Walks on Hierarchical and Fractal Topological Structures.** G. Schneckenreither, FBS 35
ISBN ebook 978-3-903347-35-9, DOI 10.11128/fbs.35, ARGESIM Publ. Vienna, 2020
- A Framework Including Artificial Neural Networks in Modelling Hybrid Dynamical Systems.** Stefanie Winkler, FBS 34
ISBN ebook 978-3-903347-34-2, DOI 10.11128/fbs.34, ARGESIM Publ. Vienna, 2020; ISBN print 978-3-903311-09-1, TUVerlag Wien, 2020
- Modelling Synthesis of Lattice Gas Cellular Automata and Random Walk and Application to Gluing of Bulk Material.** C. Rößler, FBS 33
ISBN ebook 978-3-903347-33-5, DOI 10.11128/fbs.33, ARGESIM Publ. Vienna, 2020; ISBN print 978-3-903311-08-4, TUVerlag Wien, 2020
- Combined Models of Pulse Wave and ECG Analysis for Risk Prediction in End-stage Renal Disease Patients.** S. Hagmair, FBS 32
ISBN ebook 978-3-903347-32-8, DOI 10.11128/fbs.32, ARGESIM Publ. Vienna, 2020
- Mathematical Models for Pulse Wave Analysis Considering Ventriculo-arterial Coupling in Systolic Heart Failure.** S. Parragh, FBS 31
ISBN ebook 978-3-903347-31-1, DOI 10.11128/fbs.31, ARGESIM Publ. Vienna, 2020
- Variantenmanagement in der Modellbildung und Simulation unter Verwendung des SES/MB Frameworks.** A. Schmidt, FBS 30;
ISBN ebook 978-3-903347-30-4, DOI 10.11128/fbs.30, ARGESIM Verlag, Wien 2019; ISBN print 978-3-903311-03-9, TUVerlag Wien, 2019
- Classification of Microscopic Models with Respect to Aggregated System Behaviour.** Martin Bicher, FBS 29;
ISBN ebook 978-3-903347-29-8, DOI 10.11128/fbs.29, ARGESIM Publ. Vienna, 2017; ISBN print 978-3-903311-00-8, TUVerlag Wien, 2019
- Model Based Methods for Early Diagnosis of Cardiovascular Diseases.** Martin Bachler, FBS 28;
ISBN ebook 978-3-903347-28-1, DOI 10.11128/fbs.28, ARGESIM Publ. Vienna, 2017; ISBN print 978-3-903024-99-1, TUVerlag Wien, 2019
- A Mathematical Characterisation of State Events in Hybrid Modelling.** Andreas Körner, FBS 27;
ISBN ebook 978-3-903347-27-4, DOI 10.11128/fbs.27, ARGESIM Publ. Vienna, 2016
- Comparative Modelling and Simulation: A Concept for Modular Modelling and Hybrid Simulation of Complex Systems.** FBS 26,
N. Popper, FBS 26; ISBN ebook 978-3-903347-26-7, DOI 10.11128/fbs.26, ARGESIM Publ. Vienna, 2016
- Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBE-Ansatzes.** Gunnar Maletzki, FBS 25;
ISBN ebook 978-3-903347-25-0, DOI 10.11128/fbs.25, ARGESIM Publ. Vienna, 2019; ISBN Print 978-3-903311-02-2, TUVerlag Wien, 2019
- A Comparative Analysis of System Dynamics and Agent-Based Modelling for Health Care Reimbursement Systems.** P. Einzinger,
FBS 24; ISBN ebook 978-3-903347-24-3, DOI 10.11128/fbs.24, ARGESIM Publ. Vienna, 2016
- Agentenbasierte Simulation von Personenströmen mit unterschiedlichen Charakteristiken.** Martin Bruckner, FBS 23;
ISBN ebook Online 978-3-903347-23-6, DOI 10.11128/fbs.23, ARGESIM Verlag Wien, 2016
- Deployment of Mathematical Simulation Models for Space Management.** Stefan Emrich, FBS 22;
ISBN ebook 978-3-903347-22-9, DOI 10.11128/fbs.22, ARGESIM Publisher Vienna, 2016
- Lattice Boltzmann Modeling and Simulation of Incompressible Flows in Distensible Tubes for Applications in Hemodynamics.**
X. Descovich, FBS 21; ISBN ebook 978-3-903347-21-2, DOI 10.11128/fbs.21, ARGESIM, 2016; ISBN Print 978-3-903024-98-4, TUVerlag 2019
- Mathematical Modeling for New Insights into Epidemics by Herd Immunity and Serotype Shift.** Florian Miksch, FBS 20;
ISBN ebook 978-3-903347-20-5, DOI 10.11128/fbs.20, ARGESIM Publ. Vienna, 2016; ISBN Print 978-3-903024-21-2, TUVerlag Wien, 2016
- Integration of Agent Based Modelling in DEVS for Utilisation Analysis: The MoreSpace Project at TU Vienna.** S. Tauböck, FBS 19
ISBN ebook 978-3-903347-19-9, DOI 10.11128/fbs.19, ARGESIM Publ., 2016; ISBN Print 978-3-903024-85-4, TUVerlag Wien, 2019



The EUROSIM Congress 2023, the **11th EUROSIM Congress**, will be organized by DBSS, the Dutch Benelux Simulation Society, in Amsterdam, July 3-5, 2023.

This time, the EUROSIM Congress's theme is **Simulation for a sustainable future**. And there is no better place than Amsterdam to celebrate the link between simulation and sustainability. You will be pleased to discover the fascinating and vibrant city of Amsterdam, a hub for many sectors such as logistics, ICT, art, financial and business services, and fashion, among others.

www.eurosim2022.eu



Als größte europäische Simulationstagung für Produktion und Logistik präsentiert die ASIM Fachtagung alle zwei Jahre zukunftsweisende Trends und aktuelle Entwicklungen, wissenschaftliche Arbeiten sowie interessante Anwendungen in der Industrie. Der thematische Fokus der nächsten Fachtagung lautet Nachhaltigkeit in Produktion und Logistik. Damit greift die Tagung eine wichtige gesellschaftliche Thematik auf und fokussiert gleichzeitig auf aktuelle Forschungsthemen der Simulationswelt.

www.asim-fachtagung-spl.de/



Every year, the I3M Multiconference renovates the opportunity to bring together researchers, scientists and practitioners, from the Mediterranean Area, Latin & North Americas, Europe, Asia, Africa and Australia, who are concerned with Modeling and Simulation in Industry and Academia. As a result of the joint effort of many scientific excellences, I3M is an opportunity to meet and discuss and, as usually happens, constructive debate makes people open up, brings food for thoughts as well as new ways to explore encouraging multi-disciplinary cooperation and collaborations.

Paper Submission Deadline: April 15th, 2023



The Winter Simulation Conference 2023 highlights the vital role that simulation plays in designing, planning, and operating resilient systems under uncertainty. In an increasingly inter-connected world, it is more critical than ever to ensure that systems quickly recover from and adapt to major disruptions. With its uncertainty modeling and explainable analytics capabilities, simulation is one of the key technologies that lie at the heart of building resilient systems. We invite papers that emphasize the latest advances in simulation theory and applications showcasing the integrated use of simulation with technologies ranging from the Internet of Things and statistics to AI/ML and optimization. We particularly encourage applications of simulation to improve resiliency in a wide range of domains, including but not limited to aviation, disaster response, education, energy, finance, healthcare, infrastructure, manufacturing, national security, space systems, and supply chains.



Simulation for a Sustainable Future

July, 3-5, 2023, Amsterdam, The Netherlands

The 11th edition of the EUROSIM Congress is dedicated to a sustainable future. And there is no better place than Amsterdam for celebrating the link between simulation and sustainability.

Topics and Tracks:

- Passenger Operations – Chairs: Michael Schultz, Bundeswehr University Munich
- Simulation/Optimization– Chairs: Idalia Flores, National University of Mexico- Mario L Rus, University of Córdoba Spain
- Simulation in Agroindustries– Chairs: Rodrigo Romero Silva, Wageningen University
- Simulation and Data – Chairs: Arnold Heemink, TU Delft
- Simulation for Digital Twins in Industry and Logistics – Chair: Francesco Longo/ Antonio Padovano, University of Calabria
- Methodology and Risk Assessment – Prof. Edmond Hajrizi, UBT
- Supply Chain Management, Logistics and Transportation – Chair M. Bagamanova, AUAS – Peter Meincke, DLR
- Industrial Case Studies
- Complex, Intelligent, Adaptive and Autonomous Systems (CIAAS)
- Environment and Sustainability Applications – Chairs: L. M. Fernández-Ahumada & M. Varo -Martinez, U. of Córdoba
- Simulation and ML Technology– Chair Dr. Mohammad Dehghani, Northeastern University
- Rare Events Analysis
- Gaming
- Epidemiological
- PhD & MSc/Bach Award – Chairs Dr. Luis Manuel Fernández-Ahumada, U. of Córdoba – Dr Miguel Mujica Mota, Amsterdam U. of Applied Sciences
- Multimodal Transport Simulation – Chairs: Raquel Salamanca, ISDEFE; Ashish Verma, IISC, Bangalore
- Simulation in Human Behaviour – Chairs: Jose Maria Ortiz-Gomez, Zayed University/ Christina Rott, Vrije Universiteit Amsterdam
- Simulation in Education– Cristina Ruiz Martin, Carleton University
- Agent-based Simulation – Hai Xiang Lin, TU Delft
- Energy Transition
- Aviation Track – Chairs: Benedikt Badanik, U. of Zilina -Miguel Mujica Mota, AUAS- Bruno Desart EUROCONTROL
- Monitoring and Control – Chairs: Dr. Igor Škrjanc, University of Ljubljana
- Healthcare Applications – Chair: Nikki Popper, TU WIEN
- Hybrid Simulation
- Manufacturing Applications – Chairs Dr. Antonella Petrillo&Dr. Fabio de Felice, U.of Napoli “Parthenope”
- Circular Economy – Chairs: Abdel Al Makhloufi & Amir Ghasemi, Amsterdam U. of Applied Sciences
- Military Applications – Chair Oliver Rose, Bundeswehr University
- Advanced Tutorials/workshops
- Applications in Engineering and Architecture –
- Financial Simulation
- Extraction Industries
- Poster Session – Chair Dr. Paolo Scala, Fedex

www.eurosim2023.eu