

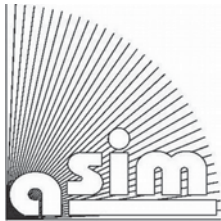
Planung Architektur Produktion Umwelt
Geowissenschaften Modelltransformation kombinierte geeigneter
Simulationen experimenten Kennzahlen
Games Modellbildung Cloud Robotik
Learning Simulations multikriterielle Virtual numerische Ökosystemen KI-Methoden Computing
Ausbildung Elektronikenwicklung Benutzerschnittstellen Modellierungssprachen Anwendungsfelder Serious
Simulation Steuerungstechnik Softwarearchitekturen Bio 2018 Modellbildung Gestaltung Reality
Spezielle standards Modellierungstechniken Messdaten Medizintechnik Simulationsumgebungen Wirtschaftsprozessen parallele Stadtplanung Parametrisierung Verfahren
System Bauwesen Machine Aufbereitung Erfassung kontinuierliche Auswertung Systemoptimierung Logistik verteilte Fahrzeugtechnik Verkehrssystemen Entwicklung
System Bauwesen Machine Aufbereitung Erfassung kontinuierliche Auswertung Systemoptimierung Logistik verteilte Fahrzeugtechnik Verkehrssystemen Entwicklung
Spezielle standards Modellierungstechniken Messdaten Medizintechnik Simulationsumgebungen Wirtschaftsprozessen parallele Stadtplanung Parametrisierung Verfahren
System Bauwesen Machine Aufbereitung Erfassung kontinuierliche Auswertung Systemoptimierung Logistik verteilte Fahrzeugtechnik Verkehrssystemen Entwicklung
Spezielle standards Modellierungstechniken Messdaten Medizintechnik Simulationsumgebungen Wirtschaftsprozessen parallele Stadtplanung Parametrisierung Verfahren

ASIM 2018

24. Symposium Simulationstechnik

4. - 5. Oktober 2018

HafenCity Universität Hamburg



Tagungsband

ASIM 2018
24. Symposium Simulationstechnik

4. bis 5. Oktober 2018
HafenCity Universität Hamburg

Zusammenfassung der Beiträge
Christina Deatcu (Hrsg.)
Thomas Schramm (Hrsg.)
Kay Zobel (Hrsg.)



Arbeitsgemeinschaft Simulation ASIM in der Gesellschaft für Informatik GI

ARGESIM Report 56 * ASIM Mitteilung AM 168

ISBN 978-3-901608-12-4 (print) * ISBN 978-3-901608-17-9 (ebook) * DOI: 10.11128/arep.56

ARGESIM Reports

Published by **ARGESIM** on behalf of **ASIM**, www.asim-gi.org,
(Arbeitsgemeinschaft Simulation, GI Fachausschuss)

Series Editors:

Felix Breitenecker
TU Wien; Felix.Breitenecker@tuwien.ac.at
Thorsten Pawletta
Univ. Wismar, Germany, Thorsten.Pawletta@hs-wismar.de

ARGESIM Report AR 56 ASIM Mitteilung AM 168

Title: Tagungsband ASIM 2018 – 24. Symposium Simulationstechnik

Subtitle: ASIM SST 2018, 4.10. – 5.10.2018, HafenCity Universität Hamburg

Editors: Christina Deatcu, Thomas Schramm, Kay Zobel
Christina.Deatcu@hs-wismar.de

ISBN print 978-3-901608-12-4

ISBN ebook 978-3-901608-17-9

DOI 10.11128/arep.56

© by ARGESIM / ASIM, Wien, 2018

Printer: WIRmachenDRUCK GmbH, www.wir-machen-druck.de
Print Availability: ASIM Office Wismar, www.asim-gi.org

Publication Date: print 6.10.2018, ebook 10.10.2018

Pages: 326 + x pages

ARGESIM Publisher

c/o F. Breitenecker, Math. Modelling and Simulation Group, TU Wien
Wiedner Hauptstrasse 8 - 10, A - 1040 Vienna
Tel: +43-1-58801-10115, Fax: +43-1-58801-910115
Email: info@argesim.org; WWW: www.argesim.org

Copyright Information / Regulations ARGESIM

ARGESIM is a non-profit scientific society generally aiming for dissemination of information on system simulation - from research via development to applications of system simulation. ARGESIM's primary publication is the journal SNE – Simulation Notes Europe with open access to all contributions; generally, the authors retain the copyright of their SNE contributions. This copyright regulation holds also for ARGES-IM Reports and ARGESIM Advances in Simulation publishing conference volumes for ASIM, MTHMOD, and EUROSIM (in consideration of copyright regulations for related conference publications) and monographs on system simulation (mainly PhD theses).

Tagungsleitung:

Thomas Schramm, HafenCity Universität Hamburg

Christina Deatcu, Hochschule Wismar

Programmkomitee:

Fernando Barros, Universidade de Coimbra

Felix Breitenecker, TU Wien

Kurt Chudej, Universität Bayreuth

Walter Commerell, Hochschule Ulm

Christina Deatcu, Hochschule Wismar

Umut Durak, DLR Braunschweig

Alexander Fay, Helmut-Schmidt-Universität Hamburg

Leo Gall, Bausch-Gall GmbH, München

Torsten Gerlach, DLR Braunschweig

Joachim Haase, Fraunhofer IIS/EAS Dresden

Peter Junglas, PHWT Vechta/Diepholz

Andreas Körner, TU Wien

Heinz-Theo Mammen, Hella KGaA Hueck & Co., Lippstadt

Klaus Panreck, Fachhochschule Bielefeld

Thorsten Pawletta, Hochschule Wismar

Nikolas Popper, dwh GmbH, Wien

Markus Rabe, TU Dortmund

Oliver Rose, Universität der Bundeswehr München

Thomas Schramm, HafenCity Universität Hamburg

Michael Striebel, HTWG Konstanz

Adeline M. Uhrmacher, Universität Rostock

Gabriel Wainer, Carleton University Ottawa

Sigrid Wenzel, Universität Kassel

Jochen Wittmann, HTW Berlin

Tagungsort:

HafenCity Universität Hamburg

Überseeallee 16

20457 Hamburg

ASIM 2018
24. Symposium Simulationstechnik
Tagungsband

INHALT

INHALT

Editorial	1
 Plenarvorträge	
Simulating and Observing the Global Water Cycle - Challenges and Strategies <i>Annette Eicker</i>	3
Entwicklungen in der Systemsimulation: von allgemein, direkt, erwartet, neu, klassisch, notwendig, praktikabel,... zu speziell, überraschend, unerwartet, wiederholt, exotisch, möglich, dogmatisch... <i>Felix Breitenecker</i>	5
 Simulation Technischer Systeme I	
Dynamic behaviour of power window regulator system <i>Sergey Petkun</i>	7
Optimization of operational parameters in biogas plants using the anaerobic digestion model number 1 (ADM1) <i>David Wagner, Wolfgang Schlüter</i>	17
Validierung von Näherungsverfahren zur Berechnung von langwelliger Austauschstrahlung im Rauminneren von Gebäuden <i>Katrin Reblinsky, Habbo Heinze</i>	21
 Methoden & deren Anwendung I	
Towards an Open Simulation Scenario Infrastructure <i>Bikash C. Karmokar, Bharvi Chhaya, Umut Durak, Shafagh Jafer, Sven Hartmann</i>	29
How to Define SES Trees for Variability Modeling <i>Christina Deatcu, Hendrik Folkerts, Thorsten Pawletta, Umut Durak</i>	35
<i>SimLack</i> : Simulation-based Optimization and Scheduling of Generic Powder Coating Lines <i>Lukas Hollenstein, Adrian Lötscher, Fabian Luccarini</i>	45
 Simulation in Ausbildung und Training I	
oHMint: An Online Mathematics Course and Learning Platform for MINT Students <i>Helena Barbas, Franz Konieczny, Alexander Lohse, Thomas Schramm</i>	51
An Object-Oriented Approach to Modelling Technical Systems <i>Dirk Eisenbiegler, Tio Emmler</i>	55
Online-Komponenten der Lehre an der TU Wien <i>Andreas Körner, Stefanie Winkler, Ruth Leskovar, Franziska Gorgas</i>	65

Produktionssysteme & Energieaspekte (SPL I)

Eignung kombinierter Simulation zur Darstellung energetischer Aspekte in der Produktionssimulation <i>Anna Carina Römer, Martina Rückbrod, Steffen Straßburger</i>	73
Hybrid modeling approach for prediction of energy demand and power peaks in intralogistic systems <i>Armin Siegel, Karsten Turek, Elisa Michelin, Thorsten Schmidt</i>	81
A flexible material flow and energy simulation in the context of Industry 4.0 <i>Andreas Buswell, Wolfgang Schlüter</i>	89
Approach on Evaluating Material Handling Simulation Runs under Consideration of different Target Groups <i>Martin Däumler, Karl-Benedikt Reith, Simon Hochholzer, Thorsten Schmidt</i>	97

Methoden & deren Anwendung II

Simulation-based Evaluation of Dynamic Vehicle Routing Problem Features for Algorithm Selection <i>Thomas Mayer, Tobias Uhlig, Oliver Rose</i>	107
Hybride Modellbildung in Verbindung mit neuronalen Netzen <i>Stefanie Winkler, Andreas Körner, Felix Breitenecker</i>	117
Erstellung eines mikroskopischen Analogons zu einem makroskopischen Fishing Model <i>Dennis Leser, Martin Bicher, Niki Popper</i>	123
Optimale Steuerung und theoretische Analyse eines mathematischen zwei Serotypen Dengue-Fieber Modells <i>Gaby Albrecht, Kurt Chudej</i>	127

Simulation in Ausbildung und Training II

Analysis of Practical Reasons for Training Simulators in Manufacturing <i>Benjamin Knoke, Christian Gorltd, Klaus-Dieter Thoben</i>	137
Ein neuer Ansatz zur Modellbildung und Simulation mit VR- und AR- Brillen am Beispiel der Fabrikplanung <i>Marc Schlegel, Uwe Bracht</i>	143
Die lernende Lernfabrik – eine intelligente Lehr-Lernumgebung zur Energie- und Ressourceneffizienz (ILehLe) <i>Mathias Bös, Bastian Thiede, Gerrit Posselt, Christoph Herrmann</i>	149

Logistik-, Lagerungs-, Verkehrssysteme (SPL II)

Zuordnungsmethoden bei Dual-load Fahrzeugen im Horizontalverkehr von Container Terminals <i>Anne Kathrina Schwientek, Fredrik Branding, Ann-Kathrin Lange, Carlos Jahn</i>	157
Influence of TAS' characteristics on the related drayage network <i>Ann-Kathrin Lange, Kristof Ole Kühl, Anne Kathrina Schwientek, Carlos Jahn</i>	167
Praxisbeispiel: Simulationsmodell eines Shuttle-Systems zur Ermittlung des Systemverhaltens <i>Anja Langanki, Thomas Kriehn, Franziska Schloz, Karl-Heinz Wehking, Stefan Kuhlins, Markus Fittinghoff</i>	171
Strategies to Mitigate the Impacts of Climate Change Related Events on Public Transit Networks <i>Daniel Lückerath, Manfred Bogen, Erich Rome, Oliver Ullrich, Rainer Worst, Jingquan Xie</i>	175

Methoden & deren Anwendung III

Agent-based Simulation with Process-interaction Worldview <i>Shufang Xie, Tao Zhang, Oliver Rose</i>	183
Agent-based Simulation of Job Shop Production <i>Tao Zhang, Shufang Xie, Oliver Rose</i>	193
Agentenbasierte Modellierung der Schlaganfall Versorgung in Österreich <i>Dominik Rothschedl, Jakob Rosenberger, Martin Bicher, Niki Popper</i>	203
Modellierung von Bewegung mit Hilfe Zellulärer Automaten <i>Carina Hörandtner, Andreas Körner</i>	213

Simulation Technischer Systeme II

Von der Co-Simulation zu neuronalen Netzmodellen: automatisierte Erzeugung von schnellen, schaltbaren Systemmodellen <i>Philipp Ebeling, Julian Kalmus, Wilhelm Tegethoff</i>	219
LON-Bussimulation mit SimEvents zur Auslastungs- und Fehlerabschätzung <i>Alexander Martens, Olaf Simanski, Christian Bock, Olaf Hagendorf</i>	225
Leitungsmodell für Bordnetzuntersuchungen <i>Leonard Gysen, Joachim Haase</i>	235

Methoden & deren Anwendung IV

Model-based Analysis of Maintenance-induced Availability of Aircraft in an Airline Network <i>Marie Bieber, Barbara Glock, Alexander Plagemann, Nikolas Popper</i>	239
Data Farming und simulationsbasierte Robustheitsanalyse für Fertigungssysteme <i>Thomas Schulze, Niclas Feldkamp, Sören Bergmann, Steffen Straßburger</i>	243
Simulation in Wood Science: a Novel Model for the Process Gluing <i>Carina Rößler, Felix Breitenecker, Martin Riegler</i>	253

Simulation in Umwelt-, Geowissenschaften & Medizin

Dynamische Modelle und Geographische Informationssysteme: Bestandsaufnahme und Anforderungen an ein integriertes Modellieren in Raum und Zeit

Jochen Wittmann 259

Proposing Multiple-Criteria Ranking to Simulate Building Renovation in Cities

Arjun Jamil, Nils Weiss, Thomas Preisler, Wolfgang Renz 265

A Difference Equation Approach for Modeling Arterial Wave Reflection

Laura Lotteraner, Bernhard Hametner, Siegfried Wassertheurer, Felix Breitenecker 269

Koordinations- und Planungsprobleme (SPL III)

Development of a simulation model to analyze the performance of decentral rescheduling algorithms in production systems

Julian Sundermeier, Felix Gehlhoff, Alexander Fay 275

Simulationsbasierte Optimierung des Umgangs mit Chef-Aufträgen im Produktionsbetrieb

Max Melter, Jens Heger, Carsten Wagner 285

Effiziente Logistikplanung mittels angewandter Simulation

Anne-Sophie Sabrowski, Knut Borrmann 291

An approach to a self-organizing production in comparison to a centrally planned production

Torsten Munkelt, Martin Krockert 299

Methoden & deren Anwendung V

Open Simulation Software - Development and Application

Tom Warnke, Frank Krüger, Adelinde M. Uhrmacher 307

Generierung von Steuerungen für Gelenkarmroboter mit simulationsbasiertem Reinforcement-Learning

Georg Kunert, Thorsten Pawletta 313

A Comparison of Simulation Tools for Multibody Models used for Anatomic Joints

Ruth Leskovar, Andreas Körner, Felix Breitenecker 321

ASIM 2018
24. Symposium Simulationstechnik
Tagungsband

Editorial

Simulation ist **die** Methode, um komplexe Systeme zu verstehen und zu optimieren. Sie versucht die Realität nachzubilden und Vorhersagen zu machen. Es gibt in diesem Kontext eine schwer zu widerlegende Hypothese von Nick Bostrom aus dem Jahr 2003 [1]. Etwas lax formuliert: wenn die Menschheit lange genug überlebt, sich technisch weiter entwickelt und neugierig bleibt, wird sie Vorgänger-Simulationen betreiben. Die Wahrscheinlichkeit ist dann extrem hoch, dass wir selbst in einer Simulation leben. Tatsächlich lässt sich wissenschaftlich ernsthaft fragen, ob sich dies prüfen lässt. Eine mögliche Konsequenz aufgrund der notwendig diskreten Raum-Zeit-Simulation wäre z.B. eine Richtungsanisotropie hochenergetischer kosmischer Strahlung, die noch nicht beobachtet wurde. Ein anderer Ansatz wäre nach verborgenem Code in der grundsätzlichen Beschreibung der Welt zu suchen. S. James Gates Jr. scheint tatsächlich fehlerkorrigierenden Code in seiner Adinkra-Methode zu finden, um supersymmetrische Teilchenphysik zu beschreiben [2].

Das alles ist seltsam, zeigt jedoch, dass man nicht nur Simulation betreiben, sondern auch **über** Simulation und ihre Beziehung zur Realität nachdenken kann und sollte. Dass, wenn Bostrom Recht hat, das gigantische Klimaexperiment, welches die Menschheit gerade durchführt, dann nur eine Simulation wäre, hilft -- zumindest uns -- nicht im Geringsten.

Uns helfen aber die Experimente, Messungen und Simulationen mit den gewonnenen Daten, wie Sie Annette Eicker in ihrem Plenar-Vortrag *Simulating and Observing the Global Water Cycle - Challenges and Strategies* vorstellt. Leider müssen wir uns aktuell aber auch mit der Frage auseinandersetzen, dass solche Simulations-Ergebnisse z.B. zur Klimaentwicklung einfach angezweifelt werden, wenn sie politisch oder wirtschaftlich nicht opportun erscheinen. In diesem Kontext ist es natürlich überhaupt nicht hilfreich, wenn dubiose Journale oder Kongresse tatsächlich fake-science veröffentlichen, um des Geldes oder der Karriere der Autoren willen.

Die ASIM 2018 fällt ganz sicher nicht in diese Kategorie, wie die hier zusammengestellten und begutachteten fast 50 Artikel belegen. Auch die kritische Selbst-Reflexion findet statt, wie besonders Felix Breiteneckers Plenar-Vortrag *Entwicklungen in der Systemsimu-*

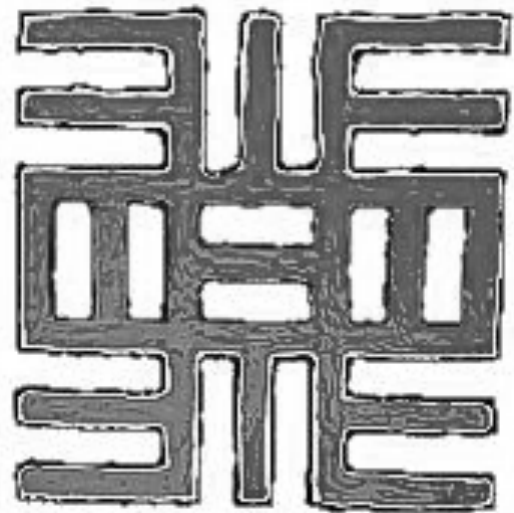
lation aufzeigt. Die in diesem Tagungsband dokumentierten Arbeiten decken einen weiten Bereich von konkreten technischen Simulationen über theoretische Grundlagen und dem Entwurf neuer Konzepte bis hin zu didaktischen und methodischen Überlegungen zur Ausbildung zukünftiger Ingenieurinnen, Ingenieure, Wissenschaftlerinnen und Wissenschaftlern.

Sie zeigen einen Querschnitt der Aktivitäten und den hohen erreichten Standard im Bereich Modellbildung und Simulation.

Ich wünsche der geneigten Leserin und dem Leser interessante Einsichten und natürlich auch Vergnügen beim Studium der Artikel.

Hamburg im Oktober 2018

Tom Schramm



Adinkras sind eine westafrikanische Symbolform, die S. James Gates Jr. zu seinen physikalischen Graphen inspirierte. Dieses Symbol für lebenslanges Lernen und der andauernden Suche nach Wissen bedeutet „NEA ONNIM NO SUA A, OHU“, frei übersetzt: „Er, der nicht weiß, kann wissen durch lernen“; vielleicht auch ein Motto für die Tagung ASIM 2018.

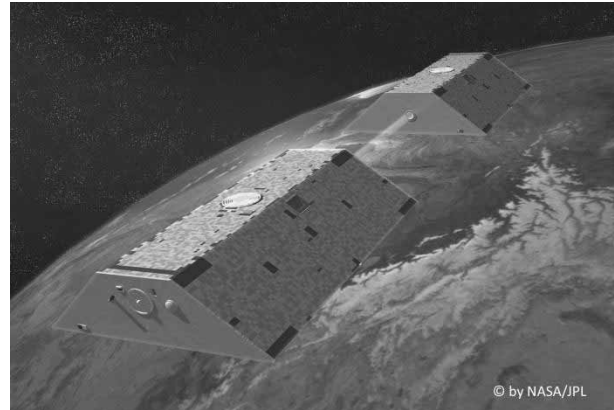
- [1] Bostrom, N. Are you living in a computer simulation? *Philosophical Quarterly*. 2003; Vol 53, No. 211: 243-255.
- [2] Gates, S. James. Symbols of Power: Adinkras and the Nature of Reality. *Physics World*, June 2010.

Simulating and Observing the Global Water Cycle - Challenges and Strategies

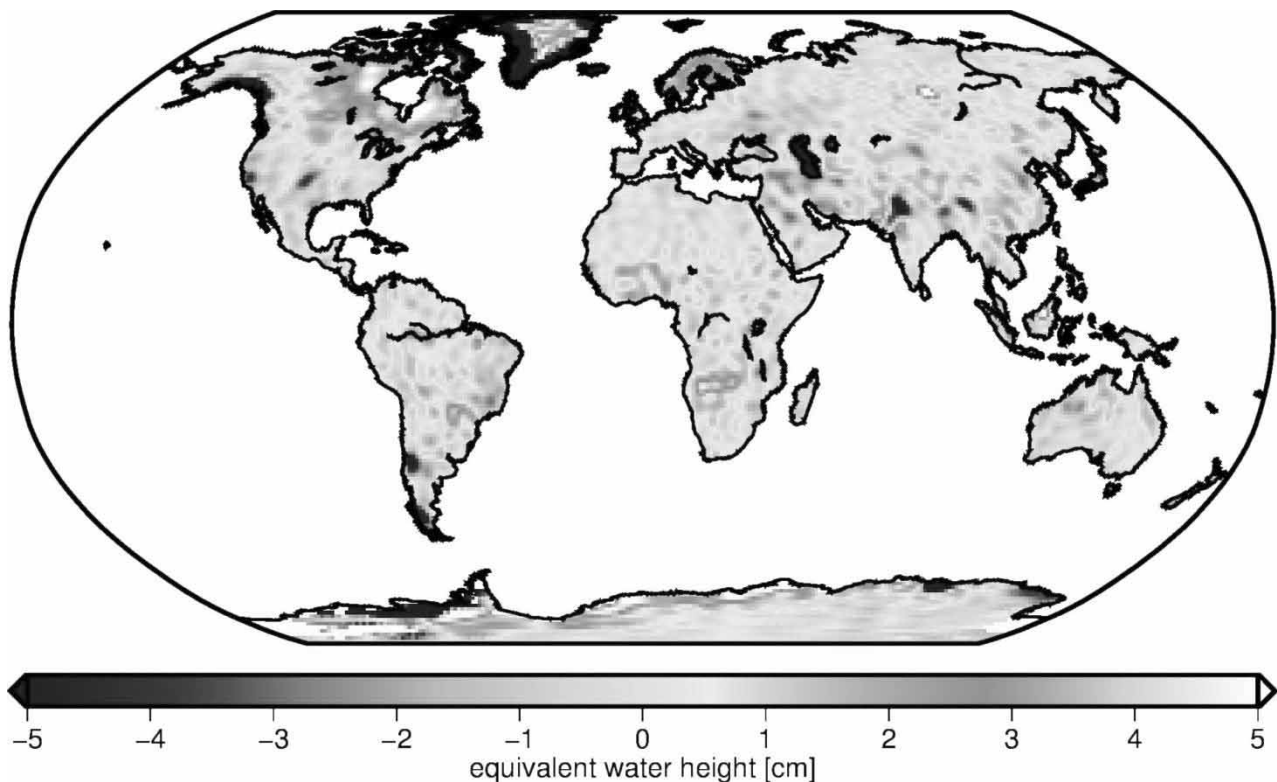
Annette Eicker

HafenCity Universität Hamburg, Studiengang Geodäsie und Geoinformatik,
Überseeallee 16, 20457 Hamburg; annette.eicker@hcu-hamburg.de

Melting glaciers, sea level rise, an increase in droughts and flooding events - many phenomena generally attributed to climate change manifest themselves as observable changes in the global water cycle. Furthermore, also anthropogenic influences, such as groundwater pumping, the embankment of reservoirs, or land use changes can have a strong impact on the distribution and availability of fresh water resources worldwide. In order to better understand such variabilities and to be able to make projections for future developments, there is a strong demand for Earth system models being able to simulate the underlying processes as realistically as possible. However, due to the highly non-linear and very complex nature of water fluxes in the Earth system in combination with an often too sparse data coverage, such models are generally accompanied by large uncertainties. Therefore, independent observations for model validation and model improvements are urgently needed.



This presentation will describe the challenges of Earth system modeling at the example of a global model of the continental water cycle. Furthermore, strategies will be introduced, how geodetic satellite observations can be used to improve the simulations by means of model calibration and data assimilation techniques.



Entwicklungen in der Systemsimulation: von allgemein, direkt, erwartet, neu, klassisch, not- wendig, praktikabel, ... zu speziell, überraschend, unerwartet, wiederholt, exotisch, möglich, dogmatisch...

Felix Breitenecker

Inst. für Analysis und Scientific Computing, Gruppe Mathematische Modellbildung und Simulation, TU Wien,
Wiedner Hauptstraße 8-10, 1040 Wien, Österreich; felix.breitenecker@tuwien.ac.at

Einleitung

Die Systemsimulation hat sich seit ihrem Beginn in den siebziger Jahren des vorigen Jahrhunderts sehr in die Breite entwickelt, in Anwendungsgebieten, und im Bereich der Modellbildung und Methoden. Shannons Definition für Simulation als “the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies for the operation of the system” erlaubt vielfältige methodische Entwicklungen lädt zur Anwendung in verschiedensten Gebieten ein. Eine Übersicht über diese Entwicklungen sprengt den Rahmen einer Monographie und bietet Stoff für umfangreiche Vorlesungen.

Eine andere, kürzere und kurzweiligere Möglichkeit ist die punktuelle Betrachtung von bestimmten meist *kontroversiellen Eigenschaften* der Entwicklung, die interessante Einsichten bietet. Eine kleine Umsortierung des Titels mit Ergänzungen nennt einige dieser kontroversiellen Eigenschaften: allgemein - speziell, direkt – überraschend, erwartet - unerwartet, neu - wiederholt, klassisch – exotisch, notwendig – möglich, praktikabel – dogmatisch, linear – nichtlinear, zu früh – zu spät, Fluch oder Segen, erforscht – noch offen, numerisch – symbolisch, genau – nicht genau. Dieser Kurzbeitrag skizziert plakativ einige dieser kontroversiellen Eigenschaften, der begleitende Vortrag geht auch auf weitere ein.

1 Die Weißen Flecken: erforscht – noch offen

Ein oberflächlicher Blick auf den *Simulationskreislauf* erweckt den Eindruck, dass alle Schritte – von Modellbildung bis Validierung hinreichend methodisch erforscht sind.

Ein genauerer Blick zeigt allerdings, dass die *Validierung* nur in einigen Anwendungsbereichen hinreichend abgedeckt ist. Heuristiken beherrschen weite Bereiche, und die Hoffnung auf gute Ergebnisse.

Ein nahezu *weißer Fleck* auf der Landkarte der Entwicklungen sind *Analysemethoden* – die derzeit kaum über die klassische (nichtlineare) Zeitbereichsanalyse hinausgehen. Shannon verlangt Experimente mit dem Modell, wozu Analysemethoden notwendig sind. An sich sprechen frühe methodische Ansätze vom Experiment als *Anwendung einer Analysemethode auf ein Modell*, heute verwenden wir fast ausschließlich die numerische Zeitbereichskeule, und es gäbe viele andere Analysemethoden – vor allem heute mit den symbolischen Verfahren.

2 Der Selbstbetrug: linear – nichtlinear

Ein Gral in der Systemsimulation: die Systemsimulation beherrscht *nichtlineare Systeme*. Die numerische Zeitbereichskeule meldet erfolgreich die tatsächliche Berechnung non nichtlinearen Zusammenhängen – und täuscht damit. Denn irgendwann landen die Algorithmen bei nichtlinearen Gleichungen, Newton hilft erfolgreich mit seinem Verfahren – durch *Linearisierung*. Spezialisten können dann überlegen, ob eine lokale a-priori – Linearisierung dasselbe Ergebnis liefert.

3 Die Wiedererfindung des Rades: neu – wiederholt, zu früh – zu spät

Manche Entwicklungen kommen zu früh, zu ungünstigen Umständen, oder sie werden nicht hinreichend kommuniziert.

In der Folge der *Modelica* Entwicklung entstand das *Functional Mockup Interface* zum erfolgreichen Model-
laustausch.

Eine Zeitreise in die achtziger Jahre stößt auf die Entwicklung des *DS-Block Interface*, mit sehr ähnlichen Eigenschaften – aber kaum (mehr) im Web zu finden.

4 Faule Agenten: zu früh – zu spät, direkt – überraschend

Agentenbasierte Simulation hat sich zum anerkannten Werkzeug der *mikroskopischen Modellbildung* und Simulation entwickelt.

Heute wird ein *PCE Agent* als *Path Computation Element* identifiziert, doch frühe Arbeiten zur agentenbasierten Simulation sprechen von *Perception-Cognition-Experience* – Agenten, mit eigener Intelligenz. Diese frühen Arbeiten gelangen aber zu einer Zeit mit zu wenig Rechnerleistung, sie kamen zu früh.

Bekannt ist der Zusammenhang zwischen agentenbasierten Modellen und PDE- bzw. ODE – Modellen im Bereich von Diffusionsprozessen bzw. Populationsdynamiken. Die ODEs/PDEs beschreiben dann über direkte Ableitungen im Wesentlichen Dichten von Agenten, und je mehr Agenten im Modell, desto näher kommen die aufsummierten Agentendichten an die Lösungen der ODEs/PDEs für die Dichten.

Es reizt die Frage nach der umgekehrten Richtung: gibt es zu einer ODE, z.B. zur Pendelgleichung, ein agentenbasiertes Modell mit ebendiesem Grenzwertverhalten? Neue Arbeiten zeigen, dass die Pendelgleichung als ODE für die Dichte von Agenten eines wohldefinierten agentenbasierten Modells identifiziert werden kann, mit eigenen Vorwärtsschwing-Agenten, Rückwärtsschwing-Agenten, und faulen Agenten. Doch lohnt diese Frage ?

5 Segen oder Fluch: praktikabel - dogmatisch

Modellbildungsmethoden wie *Modelica* oder *System Dynamics* sind anerkannt und erleichtern die Systemsimulation wesentlich. Sie sind ein praktikables Werkzeug zur Erstellung standardisierter Modelle. Die mathematisch-dogmatische Sicht lehnt diese Modellbildungsebene ab, aus mehreren Gründen.

Die Methoden sind aber in der Tat ein Segen für rasche Entwicklung, aber der Fluch folgt nach. *System Dynamics* beinhaltet als selbsterfüllende Prophezeiung ein lineares Verhalten, und *Modelica* bzw. ähnliche Systeme verführen die Entwickler zu mangelnder Innovationsfreude – die Produkte sehen alle ähnlich aus.

6 Migration, Archäologie, Poesie: klassisch – exotisch, notwendig - möglich

Natur- und Ingenieurwissenschaften sind seit Beginn klassische Domänen der Systemsimulation, Biologie und Medizin folgten bald, ebenso Ökonomie, etc. Systemsimulation wurde zu einem notwendigen Werkzeug.

Aber Systemsimulation hat ihre Meriten auch in exotischen Anwendungen, wie *Archäologie* und *Poesie*, und in soziologischen Prozessen wie *Migration*. Sie ist in diesen Bereichen eine mögliche Erkenntnisquelle für Kausalitäten, im Prinzip sind hier alle Modellbildungsmethoden erlaubt, und Ergebnisse sind als mögliche Szenarien zu sehen – mit unterschiedlichen möglichen Ergebnissen.

Simulation im Bereich der *Archäologie* hat einen eigenen Reiz: sie versucht Dinge vorherzusagen, die geschehen sind, aber das genaue Geschehen ist unbekannt. Validierung ist zwar kaum möglich, die Simulationsergebnisse liefern mögliche Szenarien – und sie erlauben eine Falsifizierung von bestimmten Szenarien.

Linguisten untersuchen Text und Gedichte auf Stilrichtungen und bestimmte Stimmungen mit statistischen Methoden. Modelle mit neuronalen Netzen bzw. Deep Learning können teilweise Stimmungen (Gefühle ?) identifizieren – mögliche Anwendungen denkbar. Und ein klassisches nichtlineares Schwingungsmodell mit Bifurkationen kann das Auf und Ab der Emotionen Petrarcas in seinen Sonetten an Laura nachvollziehen – ein Lehrmodell.

Migrationsmodelle – früher ein Werkzeug für meist qualitative Analysen – haben Tagesaktualität erreicht. Diskrete Bilanzgleichungen können relativ genau Flüchtlings- und Migrationsströme simulieren, und sind teilweise auch validierbar. Szenarien zeigen die Wirkung – und auch die Sinnlosigkeit – von Grenzschließungen und anderen Eingriffen, und können hoffentlich zum Verständnis helfen. Faszinierend ist, dass ein Modell für die Migration 2016 über Balkan und Mittelmeer auch für die Bronzezeit-Migration der sogenannten Seevölker im selben Gebiet verwendet werden kann.

Dynamic behaviour of power window regulator system

Sergey Petkun

Brose Fahrzeugteile GmbH&Co. Kommanditgesellschaft, Bamberg, Berliner Ring 1; sergey.petkun@brose.com

Abstract. The electrical window regulators belong since long time to commodity features of modern cars. They are already standard equipment for any car classes but not rather only premium one. At present it is the only electrical equipment in the car for which there is precise anti trap requirement regulated of law. The high cost pressure in economics leads to continually reduction of it for also this products. First of all it is connected with material reduction of all components the system besides glasses. To get a smaller weight the window regulator drive, it is meant to optimize the running way of glass in the car door. Next cost optimization lays in trend like the reduction number of leading rails to only single one. All this improvements make the system lighter and its behavior more dynamical. The algorithms anti trap system nowadays are based on indirect methods, which are very suitable from economic point of view. The rising dynamics of window glass in the door requires more deeper understanding of mechanical process during the moving for proper control the system. A bright spectrum of working conditions makes the task of control yet more difficult. In this paper will be studied the behavior moving glass in car door based on single guided rail.

Introduction

The dynamics of moving window in door of car has influences not only on functionality like close/open time, safety closing and so on, but also at acoustical, energetic and other aspects. A single guided window regulators are used long time in rear doors because of glass mass in the rear doors is less as glass in front doors. A trend for single guided window regulator also in front door causes necessary the analysis of dynamics the window regulators. Differences in glass masses, constructions and operation modes can not be more ignored. Every simulation should answer on some questions which de-

fine the methods, deepness and tools be used for reaching the aims such simulation. Single guided rail window regulator is nothing else as physical inverse pendulum from mechanics point of view. It has boundary restriction for its movement in form a construction the car door with a corresponding sealing.

The main question in focus of this analysis is a tilting of the glass in car door during the movement. Regarding to the glass tilting can be set different questions like "is tilting will be take place?" or "under which condition glass can be tilted" and many others. In this paper will be trying to build the mechanical model for dynamical analysis with reasonable degree of complexity. The common way on simulation of whole power window regulator system is to simulate all components together for understanding interaction between them. For that the component should be described in some modelling languages like VHDL AMS [1] or Modelica [2]. But for development of model a component, the physical understanding is the first of all what should be done. In this paper the mechanical part of the system is studied for later implementation in some modelling language of basic gained knowledge about dynamics.

1 Model description (premises)

The simple model of some inverse moving pendulum with constrains in rotation may be look like at Figure 1. The number of parameter the model depends on question for the simulation. Even such simply model involves already essential number of parameters without them it is impossible to study the dynamical behaviour. There are necessary some geometrical constrains like freedom degree tolerance on both rotational direction, declination angle of rail relative vertical axis, mass of window glass and driving force as function in time. The acting forces are arisen due to gravity, friction, elasticity. The rotational stiffness of support can be taken into account. The role of rotational stiffness is very contro-

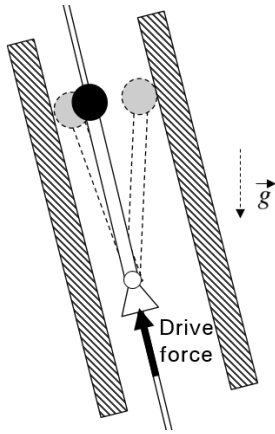


Figure 1: Simple model for dynamic analyse

versial at present. It depends on leading strategy glass in the door. If the leading role belongs a pair "carrier(glass) - rail", the rotational stiffness is very important. If the window leading is supplied on pair ""carrier(glass) - door", the role of rotational stiffness can be neglected at all.

It is desirable to have a minimal parameter set for common analysis. The base mechanical model is built based on Lagrange mechanics, what is more suitable as Newton approach [3]. As research tool is chosen Mathematica [4] because of essential number analytical transformations should be done. The general coordinates are chosen according to simplicity for description: translation movement along the rail and rotation one around the fixed point of carrier. The model is described by means of only two freedom degrees. For tilting analysis it is already minimum degree number. To determine more or less exactly rotation tolerance the window glass is taken like two-dimensional solid body with simple geometry. The simplicity of model does not mean its useless but rather reflects the restricted range of questions to this simulation. In focus the investigation is dynamic behaviour what leads to choice of corresponding parameters for model. Figure 2 shows the extended model with necessary minimal parameter set. The global coordinate axes (X_{CAD}, Z_{CAD}) are refer to earth horizontal and vertical axes. How it is shown at Figure 2 the origin is taken from CAD drawing. An origin of next coordinate system is placed in rotational center (carrier of window) for simplification the calculation. This coordinate system $\{x_0, z_0\}$ is originated with point O belonging geometrically like to carrier also to rail. The axis z_0 moves translational along

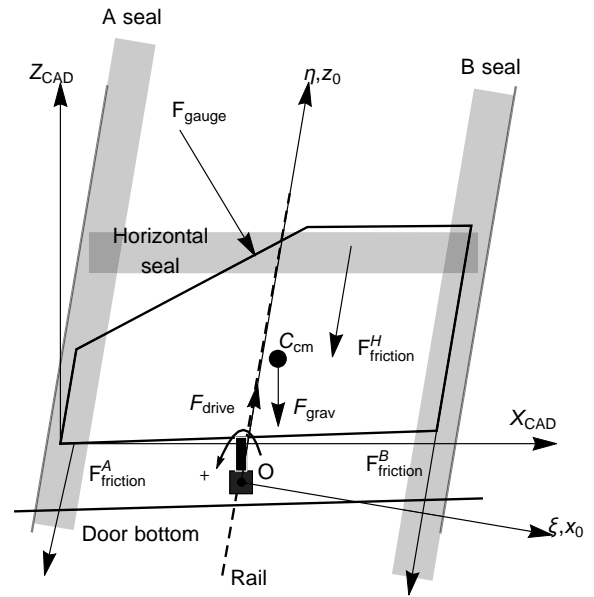


Figure 2: Next model based on physics of simple model

the rail (dashed). The carrier (black rectangle) is fasten fixed to glass(dashed) and they altogether are represented by single solid body with center of mass in point $C_{cm}\{\xi_c, \eta_c\}$. A point O is origin both the moving system $\{x_0, z_0\}, \{\xi, \eta\}$. It presents also the center of rotation the glass during tilting if it has a place. The coordinate system $\{\xi, \eta\}$ is fixed to glass and carrier with origin in point O . It moves together with glass translational along the rail and only rotates about moving point O . There are two generalized coordinate proper suitable for Lagrange approach [5]: translational movement along rail - coordinate $z_0(t)$ and angle $\alpha(t)$ is angle between $z_0(t)$ and $\eta(t)$ what reflects the rotation the window glass. Geometrical constraints are the pillar A(left) and B(right) accordingly to glass. A sealing is drawn in gray color. The window shows the neutral position of glass in the door, black frame is titled glass onto B-pillar. Dashed line represent the rail direction. The vertical gray line on the seal are door pillar (left A-pillar, right - B-pillar). They restrict the windows rotation due to supporting reaction forces. The rail is parallel to A- and B-pillars. The declination the rail and deviation the center of mass lead to stable static position supporting at B side in this case what can be seen at Figure 3 . To not overload the figure rotated position of carrier does not show.

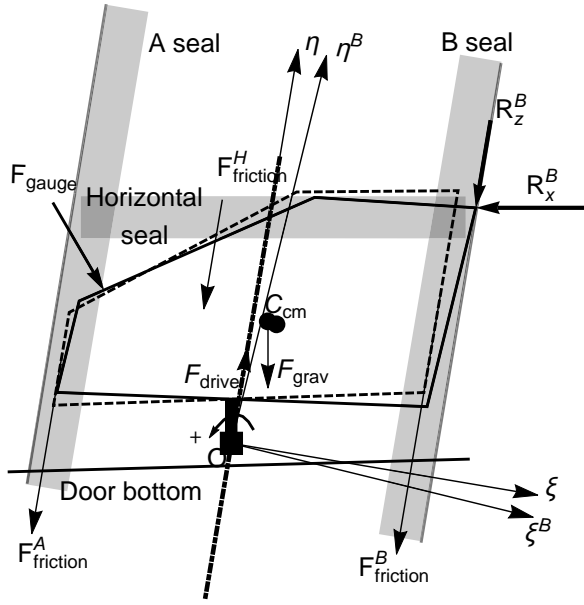


Figure 3: Stable static case like tilted on B-side

2 Lagrange equations

All acting forces at the glass can be divided on two groups: forces with known potentials and without its. The forces having potential are the gravity in C_{cm} and rotational elasticity of carrier in $O\{0,0\}$ and translational stiffness of obstacle F_{gauge} . Remaining forces are generalized forces like friction forces (seal frictions, constrain reactions), driving force. In our case there are two freedom degree and hence two Lagrange equation for $z_0(t)$ and $\alpha(t)$. Lagrange function L is defined as usual

$$L = T_{kinetic} - U_{potential}, \quad \text{where} \quad (1)$$

$$T_{kinetic} = \frac{m \cdot v_{cm}^2}{2} + \frac{J_{cm} \cdot \omega^2}{2}$$

$$L_{potential} = m \cdot g \cdot z_0(t) \cdot \cos(\alpha_0) + \frac{Stiff_{rot} \cdot \alpha^2}{2}$$

hereby m - mass of glass, α_0 - declination angle of rail, $z_0(t) \cdot \cos(\alpha_0)$ - height difference, \vec{v} - velocity the center of mass, J_{cm} -moment of glass inertia, α - rotational angle, $\omega = \dot{\alpha}$ -angular velocity of glass. For two freedom degree there are two Lagrange equations of the second

kind which look like as

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{z}_0} \right) - \frac{\partial L}{\partial z_0} = Q_{z_0} \quad (2)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}} \right) - \frac{\partial L}{\partial \alpha} = Q_{\alpha}$$

where Q_{z_0}, Q_{α} are generalized forces. Any simulation model is only a reflection some side of reality. That is why the right choice of component a building model is really modelling know-how. The proper compromise between precision, complexity and costs on modelling should be done at early phase the development process. Right choice of acting forces is the most important for dynamics studies, and at the same time it is the most difficult question. The acting forces can be taken in account in Lagrange's mechanics by means of different methods. All forces without potentials should be included in generalised ones. Friction and constrain reaction forces belong in this case to generalized ones.

The tolerances are free space between window glass and door pillars. They are necessary for free movement window glass in the car door and tight connected with manufacturing process. Hence it is tight connected with quality and costs at the end. The tolerances of neutral glass position relative to A- and B-pillars are taken equal to each other. But left and right tilt angles are different because of glass geometry. It can be observed from comparison Figures 3 with 4 In spite of the sim-

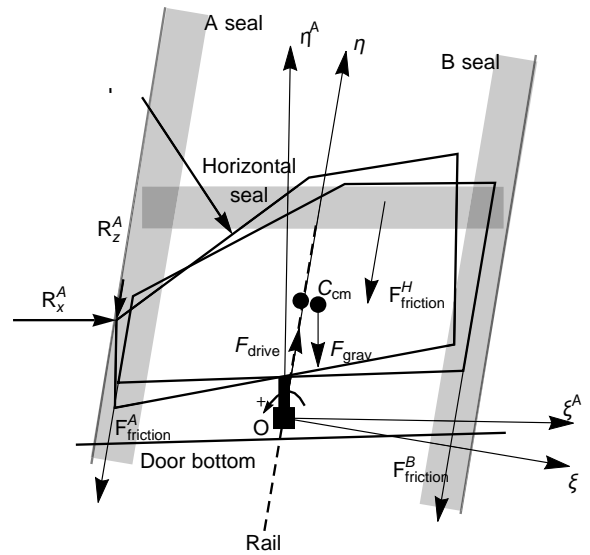


Figure 4: Tilted window glass on A-side

plicity of models from mechanical point of view, there are relative big number of parameter despite the simple geometry, which have to be taken into account. Only glass geometry is involved seven point coordinate what means for two-dimensional case already fourteen parameter values. Assumption the glass like solid body requires only a couple physical parameters (mass m , moment of inertia J_c). The friction definition brings also some pairs physical coefficients.

The main difficulties are occurred because of the principal nonlinearity of physical behaviour during the dynamics: friction direction (against velocity), support on pillar (occasionally) and so on. All this cases causes the new branch in modelling. For example, the contact glass with pillar requires the exact definition of this impact has behaviour: elastic, plastic or even plastic-elastic.

Dependent on set of parameters there are different situations from mechanical point of view: no contact glass with pillars or contact (elastic, plastic and so on). The next alternative is availability of obstacle (at which place, what kind and so on) or simple free run at all without obstacle. Without correct description it is not possible to get exactly answer about glass behaviour during closing process. The equation (1) can be looked In the case of moving without obstacle like this

$$Q_{z_0} = m \cdot (-\eta_c(\ddot{\alpha}(t) \sin(\alpha(t)) + \dot{\alpha}(t)^2 \cos(\alpha(t))) + \xi_c \cdot (\ddot{\alpha}(t) \cdot \cos(\alpha(t)) - \dot{\alpha}(t)^2 \cdot \sin(\alpha(t))) + g \cdot \cos(\alpha_0) + \ddot{z}_0(t)) \quad (3)$$

Second differential equation 2 of movement can be seen like that

$$Q_\alpha = \ddot{\alpha}(t) \cdot (m \cdot (\eta_c^2 + \xi_c^2) + J_{cm}) + \text{Stiff}_{rot} \cdot \alpha(t) - m \cdot z \ddot{0}(t) \cdot (\xi_c \cdot \cos(\alpha(t)) - \eta_c \cdot \sin(\alpha(t))) \quad (4)$$

where Stiff_{rot} is rotational stiffness of pair carrier - rail.

Generalized forces for seal friction. The generalized forces Q_α, Q_{z_0} are determined through calculation their work on virtual generalized displacement. In our case the forces, being taken in account, are two kinds: contact with sealing and supporting at pillar, horizontal chamber (friction) and obstacle force (elasticity with possible). Last forces present not continually but rather occasionally in opposite 3 and 4. The support force depends on dynamics of glass and it is difficult to predict when it has place. The obstacle force (act-

ing measure gauge) is taken into account in parameter set. Some qualitative prediction can be made without simulation: something like that - greater friction between seal and glass will reduce probability of supporting contact glass - pillar and so on.

To determine the generalized forces for coordinate z_0 and α it is used the definition generalized force: $F_u = \delta A_u / \delta u$ where δA_u is mechanical work the forces on virtual displacement δu generalized coordinate u .

All friction forces should be taken into account for calculation of virtual work.

$$Q_{z_0} = \frac{\delta A_{z_0}}{\delta z_0}, \quad Q_\alpha = \frac{\delta A_\alpha}{\delta \alpha} \quad (5)$$

The friction force F_{fric} is determined according to this definition (dry friction)

$$F_{fric} = \begin{cases} -k \cdot N \cdot \frac{\vec{v}(x,z)}{|\vec{v}(x,z)|}, & \text{if } \vec{v}(x,z) \neq \vec{0} \\ -\vec{\Phi}, & \text{if } \vec{v}(x,z) = \vec{0}, |\vec{\Phi}| \leq k \cdot N \\ -k \cdot N \cdot \frac{\vec{\Phi}}{|\vec{\Phi}|}, & \text{if } \vec{v}(x,z) = \vec{0}, |\vec{\Phi}| > k \cdot N \end{cases} \quad (6)$$

where $\vec{\Phi}$ is the resultant of all forces, other than frictional ones, N - magnitude of normal (pressure) force, k - friction coefficient and $\vec{v}(x,z)$ - velocity of point the applied friction force.

Sealing friction is in principle distributed force. But for simplicity the all acting friction forces (glass-sealing pair) can be applied in point as resultant of their action. There are three points for friction forces (A_m, B_m, D_m). It is resultant action on pair glass - A-pillar, B-pillar and horizontal chamber respectively. The resulting forces seal-glass is placed in the middle of glass edges under supposition the full contact all time between glass and seal for simplicity. Friction is directed always against velocity. The application point of resulting friction force places big role only for rotation movement because the velocities are dependable on removal of point from rotation center. For translational movement solid body it is not important. Friction direction is $-\frac{\vec{v}(x,z)}{|\vec{v}(x,z)|}$, where $v(x,z)$ velocity vector in Cartesian coordinate system $\{x, z\}$ accordingly to 6.

The way for determination of Q_α and Q_{z_0} is simple: the first of all - transformation $\{x, z\}$ in $\{\alpha, z_0\}$. Next step is to calculate the work on the virtual displacement δz_0 and $\delta \alpha$. After that the δA should be divided into δA_α and δA_{z_0} . At the end, generalized forces is to calculated like in 5.

For demonstration of complexity the analytical

transformation, the generalized forces are calculated as a example for single friction force in coordinate system $\{x, z_0\}$. The resulting friction force is placed in a point $\{\eta, \xi\}$ at the glass. After first transformation the coordinates are

$$\begin{aligned} x(t) &= \xi \cdot \cos(\alpha(t)) - \eta \cdot \sin(\alpha(t)) \\ z(t) &= \eta \cdot \cos(\alpha(t)) + \xi \cdot \sin(\alpha(t)) + z_0(t) \end{aligned}$$

the velocity of this point remains yet readable

$$\begin{aligned} \dot{x}(t) &= -\eta \dot{\alpha}(t) \cos(\alpha(t)) - \xi \dot{\alpha}(t) \sin(\alpha(t)) \\ \dot{z}(t) &= -\eta \dot{\alpha}(t) \sin(\alpha(t)) + \xi \dot{\alpha}(t) \cos(\alpha(t)) + \dot{z}_0(t) \end{aligned}$$

The friction direction in $\{x(t), z(t)\}$ is

$$\left\{ -\frac{\dot{x}(t)}{\sqrt{\dot{x}(t)^2 + \dot{z}(t)^2}}, -\frac{\dot{z}(t)}{\sqrt{\dot{x}(t)^2 + \dot{z}(t)^2}} \right\} \quad (7)$$

and after substitution, the friction force as vector $\vec{F}_0(F_x, F_z)$ represented with the coordinates

$$\begin{aligned} F_x &= \frac{F_0 \cdot \dot{\alpha}(t) \cdot (\eta \cos(\alpha(t)) + \xi \sin(\alpha(t)))}{\sqrt{(\eta^2 + \xi^2) \dot{\alpha}(t)^2 + 2\dot{\alpha}(t) \cdot \dot{z}_0(t) \cdot R_{\xi\eta} + \dot{z}_0(t)^2}} \\ F_z &= \frac{F_0 \cdot \dot{\alpha}(t) \cdot (\eta \sin(\alpha(t)) - \xi \cos(\alpha(t))) - \dot{z}_0(t)}{\sqrt{(\eta^2 + \xi^2) \dot{\alpha}(t)^2 + 2\dot{\alpha}(t) \cdot \dot{z}_0(t) \cdot R_{\xi\eta} + \dot{z}_0(t)^2}} \end{aligned} \quad (8)$$

$$\text{where} \quad R_{\xi\eta} = (\xi \cos(\alpha(t)) - \eta \sin(\alpha(t)))$$

Next step is to calculate virtual displacement as following

$$\begin{aligned} \delta x(t) &= \delta \alpha \cdot (-\eta \cos(\alpha(t)) - \xi \sin(\alpha(t))) \\ \delta z(t) &= \delta z_0 + \delta \alpha \cdot (\xi \cos(\alpha(t)) - \eta \sin(\alpha(t))) \end{aligned}$$

Now it is already all ready for determination of virtual work like

$$\delta A = \vec{F} \cdot \delta \vec{r}, \quad \delta \vec{r} = \{\delta x(t), \delta z(t)\}$$

Virtual work δA contains two components of generalized forces $\delta A = Q_\alpha \cdot \delta \alpha + Q_{z_0} \cdot \delta z_0$ and can be separated for determination Q_α and Q_{z_0} . It is valid only under assumption that resultant applied point of friction force remains all time the same in in coordinate

system fixed connected with glass. In reality it is big question if it reasonable to replace distributed force field through concentrated force but for our aim it can be (may be) good enough.

From this example, it is clear to understand that the "handmade" proof for correctness of symbolic manipulation is very unsure if it is at all possible. If the points of application seal friction have coordinates $\{\xi_A, \eta_A\}, \{\xi_B, \eta_B\}, \{\xi_H, \eta_H\}$ corresponding to A-, B- and H-horizontal seals, then the generalized force can be calculated like this

$$\begin{aligned} Q_\alpha &= Q_\alpha(\xi_A, \eta_A) + Q_\alpha(\xi_B, \eta_B) + Q_\alpha(\xi_H, \eta_H) \\ Q_{z_0} &= Q_{z_0}(\xi_A, \eta_A) + Q_{z_0}(\xi_B, \eta_B) + Q_{z_0}(\xi_H, \eta_H) \end{aligned} \quad (9)$$

The system of differential equations for determination $\alpha(t)$ and $z_0(t)$ is obtained from combination 2 with 3 and 4. The solution of the system describes the movement of inverse pendulum with friction under constraints.

Boundary constrains A- and B-pillars represent the constraints during the glass movement. They give limits for physical possible rotation.

At the Figures 3 and 4 they can be seen like angles $\alpha_{critA}, \alpha_{critB}$ between axes η and η_A, η_B state limits for left and right rotation correspondingly. When the glass reaches during tilting the critical angles $\alpha_{critA} = \angle \eta O \eta^A$ or $\alpha_{critB} = \angle \eta O \eta^B$ the support reaction forces prevent the further rotation and give a little bit or even all impact energy back.

Mathematically it can be realized by means of changing the direction of rotation velocity on the opposite with or without losses. Such solution is acceptable until glass does not lay all time at the supporting pillar lay continually some period of time during movement. In such cases the theory of mechanical impact is usually used. The most important parameter in this concept is coefficient of restitution for determination the energy balance before and after collision.

The constraints like pillars with geometrical form of glass and sealing are required some geometrical description. By that, the minimal set of points is needed for description the required geometry. This point set based on conception the minimalistic approach to this problem with focus on glass dynamics in the car door. The idea of minimalistic approach is based on hope to implement the tested model later in Modelica or VHDL modelling language.

Such minimal for dynamics required set is shown at

Figure 5 The meaning of most points is self explain-

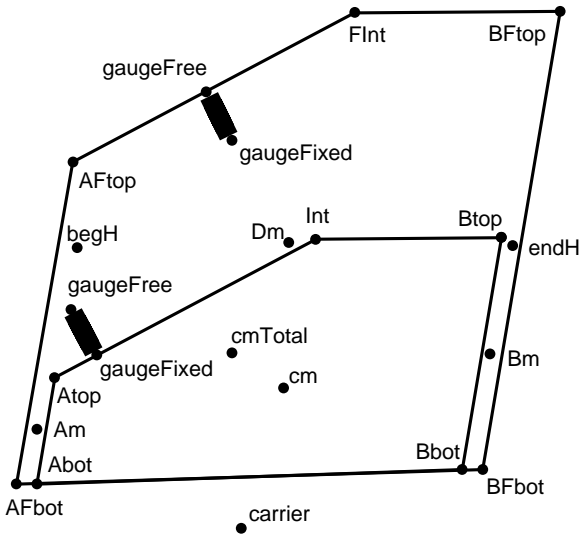


Figure 5: Minimalistic geometry set

able and clear from Figure 5. Second capital letter F in point name means the reference to frame. It can be seen the application point of the sealing friction - A_m, B_m, D_m . There are the two different possibilities for measurement of anti trap force. They are represented with points *gaugeFixed*, *gaugeFree*. More detailed it is shown also at Figure 6. The glass rotation point is replaced in point named *carrier*. The center of mass the window glass is denoted by point *cm* or at other Figures like C_{cm} . The meaning of point *cmTotal* will be explained later. Some part of points coordinates is input data from CAD like geometry of glass, other is calculated on the parametrization basis. A-pillar is built by part of frame coordinates $\{AFbot, AFtop\}$ and B-Pillar with $\{BFbot, BFtop\}$. The set of points can be divided into three groups: CAD-inputs, physical subset and parametrized derivatives. The geometry of window glass $Atop, Abot, Int, Btop, Bbot$ builds the first subset. The second one contains the points like Cm, Am, Bm, Dm .

The rest points like frame coordinates, position of measure gauge and so on belong to the last subset. The last group of coordinates can be calculated through coordinates of the first and second subsets. For example, it can be assumed that the position of the measure gauge is defined according to some distance from the left top point of glass with parameter *posFixedGauge*. The coordinates of gauge position in the reference system fixed to glass can

be represented as following

$$\begin{aligned} \xi_{gaugeFixed} &= \xi_{Atop} + \frac{posFixedGauge \cdot (\xi_{Int} - \xi_{Atop})}{\sqrt{(\eta_{Int} - \eta_{Atop})^2 + (\xi_{Int} - \xi_{Atop})^2}} \\ \eta_{gaugeFixed} &= \eta_{Atop} + \frac{posFixedGauge \cdot (\eta_{Int} - \eta_{Atop})}{\sqrt{(\eta_{Int} - \eta_{Atop})^2 + (\xi_{Int} - \xi_{Atop})^2}} \end{aligned} \quad (10)$$

If the measure gauge is fastened to the glass, these coordinates are dependent on $\alpha(t)$ and $z_0(t)$ in the coordinate system with reference to ground. The coordinates of the free end of the measure gauge are yet more complex as in 10. This example is yet another confirmation that correct preparation for calculation with this approach is the only possibility to make it with using a system of symbolic calculation.

From the same point of view, the input information can be divided into some similar groups like CAD-geometry, physical properties, parametrization group and simulation parameters.

The practical experience has shown that driving force is almost impossible to set as a force with a constant value. All acting forces are changing during the window movement. If the driving force is too small - the glass falls down. The other way round, if the force is greater than necessary - the glass is accelerated very fast, which by no means reflects reality. That is why the simple drive model was implemented.

The second reason for introducing the drive model was caused by a question about quality estimation of the dynamical model. Information from practical experience can be obtained easily for an electric driven system as for manual ones. In this way, the drive parameter group is included in the common input. Certain parameters for result visualization were included in the simulation parameter set.

Measuring gauge All mechatronic systems are potentially dangerous for their users. Electrical drives can deliver very fast and much energy. That is why systems with electric driven components like window regulator in a car should be designed for protected and controlled usage of them. The controlling of anti-trap forces is carried out by means of using a measuring gauge. Anti-trap is allowed but the trapping force is restricted. This restriction is reflected in the legislation of countries. The anti-pinch force is controlled by way of their measuring with devices like a gauge.

As already mentioned above, there are two differ-

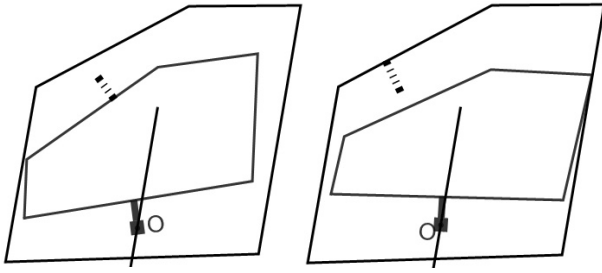


Figure 6: Two method for anti trap measuring

ent possibilities to make it. It is shown at Figure 6. There are small distinctions at this picture but they lead to great consequences for dynamical behaviour the system.

The first of all the center mass of system is changed if the gauge is fixed at glass. The moment of inertia is also another else without gauge or the if gauge is fixed at frame. The physical influences additional mass of measuring tool are very strong dependable on position the gauge at glass.

The model with gauge at glass can be reduced to model with gauge at frame by means of changing of coordinate the center mass position C_m into $C_t (= cmTotal)$ at Figure 5) and value the moment of inertia $J_{cm} \rightarrow J_t$ according to next rule

$$C_t = \frac{C_m \cdot M_{glass} + C_g \cdot M_{gauge}}{M_{glass} + M_{gauge}} \quad (11)$$

$$J_t = J_c + M_{glass} \cdot |C_t, C_m|^2 + M_{gauge} \cdot |C_t, C_g|^2$$

where C_m - center mass of glass, C_g - coordinates fixation of gauge, C_t - new center mass the system glass+gauge, J_t -moment of inertia the system glass+gauge, $|P_a, P_b|$ - is function for calculation a distance between point P_a and P_b , M_{glass} mass of glass and M_{gauge} of gauge. For left schema at the Figure 6 is valid $C_m \rightarrow C_t$, $J_c \rightarrow J_t$, for right schema: $C_m \rightarrow C_m$, $J_c \rightarrow J_c$, i.e the fixed gauge at frame does not change the dynamical behaviour the window regulator system.

3 Numerical solutions

In spite of desire to be minimalistic, the relative many parameters were involved in model. One reason is for it - dynamics, which means development process in time. For numerical calculation that means there is no result like number but rather like curve with time as abscissa.

The most important question dealing with dynam-

ical simulation is how good the simulation answers at given questions and where is physical limit of its the validity. In a general way the comparison with experiment is the right method to estimate the quality of simulation results. At the same way the dynamical measuring has the similar problem: every experimental curve is a little be differ from last one. The question remains the same - which measure result is more correct?

The second reason of difficulty the dynamics is lack of our human filling the processes which are running in time. That is why as a result of simulation, the values which are known for observer are chosen: $\alpha(t)$ $z_0(t)$. This is what can simple be observed.

The second set of results for comparison measure-simulation comes from drive: motor current,rotational velocity. For it there is enough data and already some filling what is happen in reality on the basic practical experience. What is also in list of interests: driving force and closing velocity($\dot{z}_0(t)$). This selected signal list of result for analysis is chosen as criteria for plausibility of model.

First simulation run The geometry of glass, physical values and dive parameters was taken from some real industrial project. The purpose of first simulation is to test plausibility of whole complex model. The val-

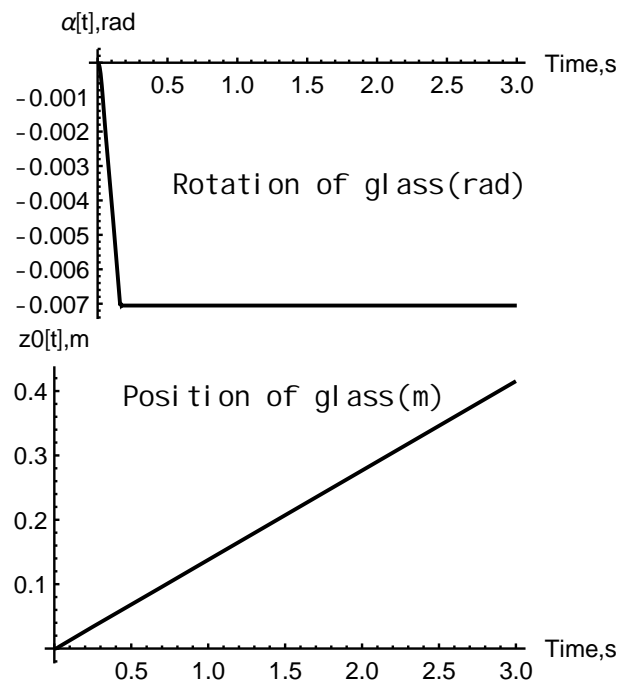


Figure 7: Rotational angle $\alpha(t)$, translational position $z_0(t)$

ues of comparison with real experiment are window velocity during closing, state of drive like motor armature rotational speed, current and so. The cable tension force is rough estimated on the base given friction and gravity. The mass of glass is $M_{glass} = 3.2 \text{ kg}$, the mass of gauge - $M_{gauge} = 0.75 \text{ kg}$. To observe the dynamical influence of gauge, it is set at glass with position $posFixedGauge = 0.1 \text{ m}$ from $Atop$. Friction is geometry dependent in accordance with data from seal manufacturer. In our case the friction is divided into three part: A-pillar, B-pillar and horizontal chamber. For this simulation it is set to $K_{fric} = 40 \text{ N/m}$. Coefficient of restitution for contact between glass and pillars is $K_{res} = 0.1$. It means almost plastic impact: the 90% energy during contact vanishes in pair pillar-seal. Running height in rail direction upto frame is about 0.45 m . The height of gauge was taken about 0.06 m . Both last values remain all time the same for all simulation.

One important peculiarity of the simulation run is to calculate contact event: gauge - door frame. The position and time of collision gauge with frame depends on whole dynamical process and it is unknown at the beginning of simulation. The initial position of glass is set in neutral state ($\alpha(0) = 0 \text{ rad}$). Assume that rotation stiffness is neglected, i.e. $Stiff_{rot}$ was set to small values. Very important values are the maximal possible angle for tilting due to geometrical constrains in form pillars: $\alpha_{critA} = 0.0136845 \text{ rad}$ and $\alpha_{critB} = -0.00705323 \text{ rad}$. Figure 7 shows only kinematic which is simple to understand, because it what we can daily see. The glass position and time give us right filling that the closing time of electrical window regulator is within real range ($3 \div 6 \text{ s}$).

To imagine simulation run better helps Figure 8. What can be measured in practical experiment - this are motor current and armature rotational speed. For it there is already proper filling what is representative of reality. They are shown at Figure 9. The drive values qualitative are in realistic range and quantitative behaviour show nothing unusual. The most interesting dynamics is hidden in forces. It is presented at Figure 10. Rough estimation of drive force which should be approximately sum of friction and gravity forces what correspond our belief. There are yet dynamical value as generalized force for variable $\alpha(t)$. It is friction torque. As simulation result in stationary range its value is about 3 Nm . For comparison with reality there is no reliable data from measures. From geometry and kinetic in simulation is detected the time for contact

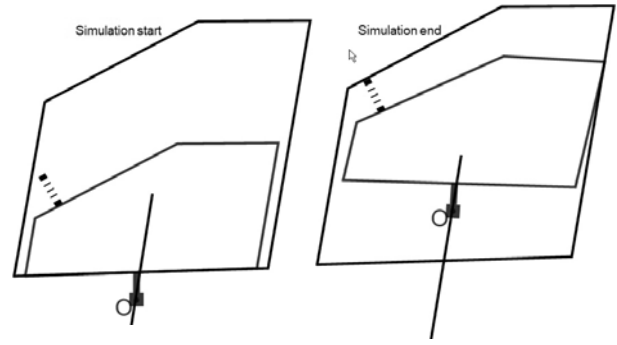


Figure 8: State for begin and end of simulation

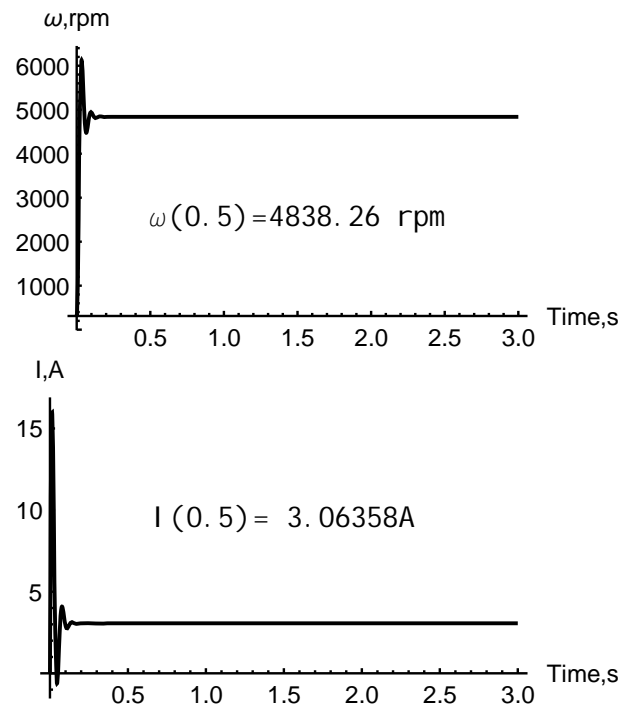


Figure 9: Rotational velocity of armature, motor current

gauge with frame. This time and collision position can be determined only during the simulation. In this case this time was $t_{contact} = 2.42381 \text{ s}$

Simulation variants The position glass supporting all time at B-pillar shows the influence of gravity and friction are greater as the acting of additional mass of gauge. If the friction force is getting less, for example, K_{fric} to set to $K_{fric} = 4 \text{ N/m}$ instead last value that was 40 N/m . Practically, it can be caused of tear and wear or seal with another property was used on the manu-

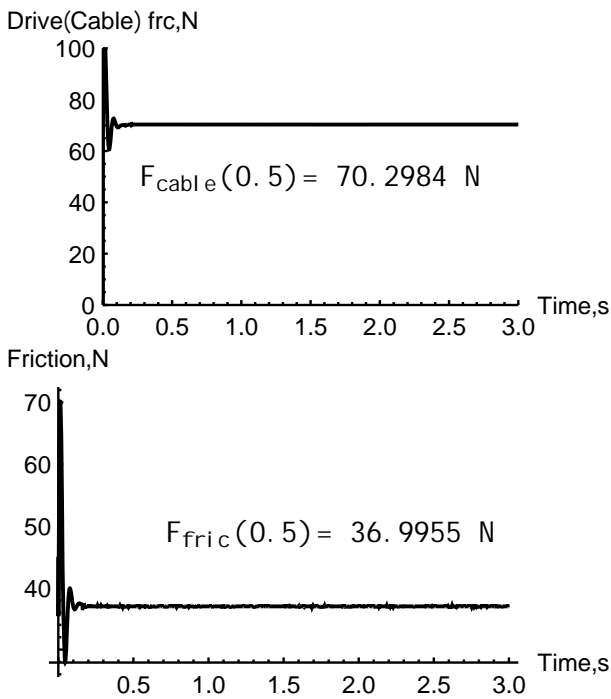


Figure 10: Cable force and resultant friction in rail direction

facturing. Rest of parameters remains the same. The simulation gives the results what will be waiting: the stationary cable force is going down from 70.29 N to 40.53 N due to decreasing of friction. More tiny oscillations on the B-pillar, drive is taken less current, contact time gauge with frame is reduced to $t_{contact} = 2.33051\text{ s}$ but qualitative there is no essential changing. It is very important to understand that the simulation is ended at moment the contact. The reason for it will be discussed later. Take a look if the coefficient of restitution is changed from $K_{res} = 0.1$ (almost plastic impact) to full elastic one, i.e. $K_{res} = 1.0$. Already at rotational angle $\alpha(t)$ can be seen the first fall down from initial start position in stationary state at Figure 11. More clear it is to observe at angular velocity, but nothing to notice at glass position. Next test is dealt with moving the gauge along the window edge. The position of measure gauge will be change to $posFixedGauge = 0.05\text{ m}$ from $Atop$, i.e. 5 cm nearer to A-pillar. There is not visible changing with previous simulation. The contact time is smaller $t_{contact} = 2.32789\text{ s}$. To get effect the tilting, mass of gauge is setting greater from $M_{gauge} = 0.75\text{ kg}$ to $M_{gauge} = 1.05\text{ kg}$. The mass of measuring equipment is getting heavier for 0.3 kg . The simulation shows the tilting has been begun. In can be clear seen at Figure 12.

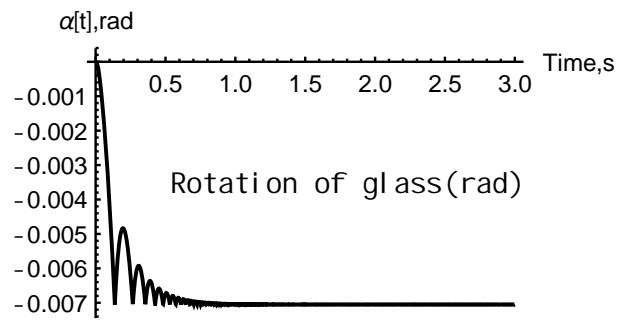


Figure 11: Rotational angle $\alpha(t)$ during elastic collision glass with pillar

To avoid this tilting there are two possibilities. First one:

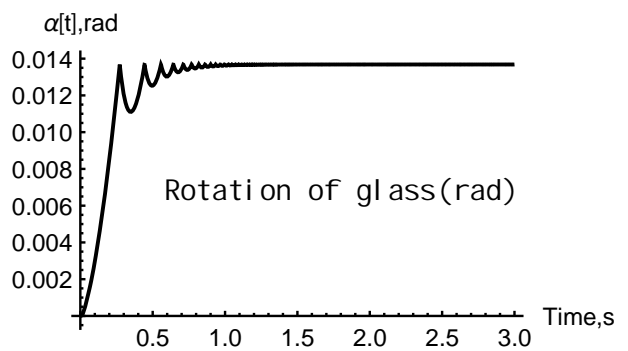


Figure 12: Tilting of window glass due to heavy measuring equipment

to give more friction, what was simulated with setting K_{fric} again to 40.0 N/m . The qualitative result is similar to angle $\alpha(t)$ at Figure 7 or moving near to the center of glass what means to change the position gauge at glass from last $posFixedGauge = 0.05$ to $posFixedGauge = 0.30\text{ m}$. The qualitative graphics is alike curve at Figure 11. To get some feeling of the influence the rotational stiffness of carrier, which was early set to small values, it is setting to $Stiff_{rot} = 0.25\text{ Nm/rad}$. The last state remains the same: $posFixedGauge = 0.30\text{ m}$, $M_{gauge} = 1.05\text{ kg}$, $K_{fric} = 4.0\text{ N/m}$. Its influence can be observed at Figure 13. Upper graphics shows the position near to center: $posFixedGauge = 0.30\text{ m}$, lower - for gauge position $posFixedGauge = 0.05\text{ m}$

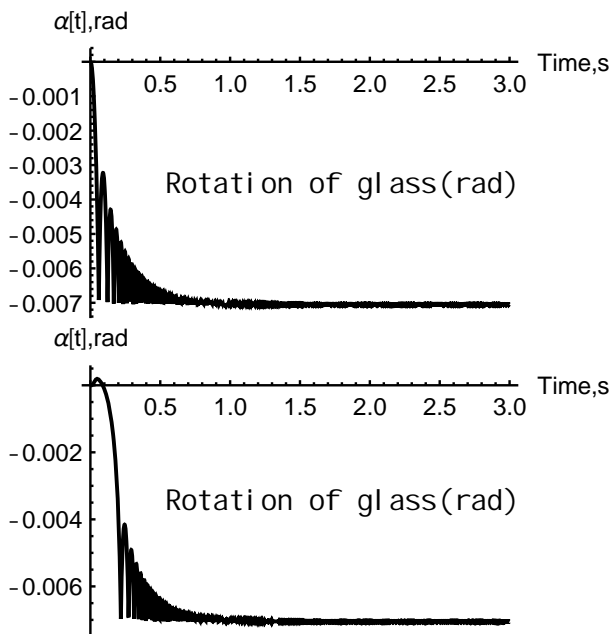


Figure 13: The gauge is near to center the glass (upper) and near to A-pillar (lower) with rotational stiffness

4 Short analysis of simulation results

The developed simulation is delivered the physically awaiting results. The simulation of selected cases convinces in plausibility of physical concept. The simulation shows clear the practical limits the mechatronics modelling. The small variations of some parameters give almost no visible alteration in dynamical situation, but in the reality it can be great acoustic, thermal or tear and wear changing. The current simulation is ended at the beginning of collision gauge frame. The number of model parameters is relative big in spite of many simplification for real process how reduction of real dimensionality from tree to two, replacing of distributed acting friction to concentrated one. But in principal, the simulation gives possibility to understand the most important influences in process the tilting of glass in the door during the movement.

5 Summary and outlook

The developed simulation has been shown clearly that dynamics needs more parameters than in statics case for description already a simple model. The method

of controlling the legislation the window regulator systems brings the big error by increasing the dynamics effects. The developers of window regulator should pay yet more attention on dynamical behaviour such products. The most simplest analyse of dynamical process is involved at least two-dimensional world : something in time. The attempts to find some reasonable ways to verification and validation the simulation model convince again that the dynamics is not yet enough understandable by modern mechatronics developers. In electronic engineering the dynamics is since long time usable in practice but the mechanics world is in principal no one-dimensional. It makes the understanding yet more difficult. It should be understood in mechanical engineering: the dynamics can no be described in numbers but rather in curves.

The next step in simulation development is to continue the movement window after collision the glass with measure gauge (or obstacle). For it is necessary to take the system state before direct collision and to set it dynamical state (position of glass, velocities, forces and also contact position) as initial conditions for new dynamics case : run against obstacle (or measure gauge).

It is possible to implement the main ideas of this simulation model in VHDL or Modelica languages. The basic for this already present in this languages in form translational and rotational branches. The both technologies like Kirchoff's laws for networks and Lagrange's formalisms based on the same - energy conservation law.

References

- [1] P.J. Ashenden, G.D. Peterson, D.A. Teegarden *The System Designer's Guide to VHDL-AMS* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA ©2002
- [2] P. Fritzson *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*, Wiley-IEEE Press. 2015
- [3] L. D. Landau and E. M. Lifshitz, *Mechanics*, Vol 1 of A Course of Theoretical Physics, Pergamon Press, Oxford, 1976.
- [4] Wolfram Research, Inc., *Mathematica*, Version 11.1, Champaign, IL (2018).
- [5] A.I. Lurie, *Analytical Mechanics*, Springer, Berlin, 2002, ISBN-10: 3540429824, ISBN-13: 978-3540429821

Optimization of operational parameters in biogas plants using the anaerobic digestion model number 1 (ADM1)

David Wagner^{1*}, Wolfgang Schlüter¹

¹Department of Engineering, Hochschule Ansbach, Residenzstraße 8, 91522 Ansbach, Germany;
**david.wagner@hs-ansbach.de*

Abstract. While the main objective in energy production is the reduction of fossil fuels, CO₂-production by fossil fuels increased over the last decade. Therefore the need for usage of regenerative energies is obvious. Biogas plants are advantageous because they can be used without spatial limitation and their substrate is abundant ubiquitously as it covers the whole range of produced organic matter from photosynthesis, municipal, industrial and animal waste. Although the need for optimization strategies is given, the fermentation process in biogas plants is complex and therefore traditional optimization approaches are cumbersome and carry the risk of complete plant failure. In this paper the optimization potential of the ADM1, which represents a detailed description of the anaerobic digestion process is analysed and compared to the standard ADM1 setup. Technical parameters like substrate composition and dilution rate are optimized to yield a high methane gas flow. It is shown that the optimization of substrate composition has a direct impact on the maximum applicable dilution rate. It is also shown that the feeding rate can be increased to yield higher productivities with optimized substrate compositions.

Introduction

In first half year of 2018 renewable energy sources replaced coal as the major energy resource with around 117,8 TWh [1]. From these energies, biomass is, due to its wide abundance, a perfect complement to wind and solar energy. Biomass is a prerequisite for biogas production. The fermentation process needed for biogas production is consisting of a multitude of consecutive steps with several bacterial species involved. The anaerobic digestion can be used to break down any given substrate besides wood. The need for rational optimization is given because the process complexity makes traditional optimizations difficult. Process optimization with the help of experimental design methods would be expensive and time-consuming. Although it is known that high protein amount yields higher theoretical methane production, the additionally produced ammonia is limiting the reaction rate. A condition for the usage of

optimization algorithms is a suitable anaerobic digestion model. Many models are available describing the process in differing detail. However, only two of them can be seen as state of the art, the Siegrist model [2] and the anaerobic digestion model No. 1 (ADM1) [3]. Both of them were developed in 2002. Compared to the ADM1 the Siegrist model is smaller, which means lower simulation times at the price of less details. Due to its versatility the ADM1 is used more often. In the case of ADM1, Parker points out, that the validation against different anaerobic digestion scenarios is only a matter of detailed feed characterization [4]. This is also one reason why the acceptance of the model in research is higher than in industry [5]. Further reasons are the complexity of the ODEs and the need for exactly characterizing feedstock as input feed [6]. However, in a recent paper this model is described as the most comprehensive one [7]. The proposed reaction scheme of the sequential steps in the ADM1 is shown in Figure 1.

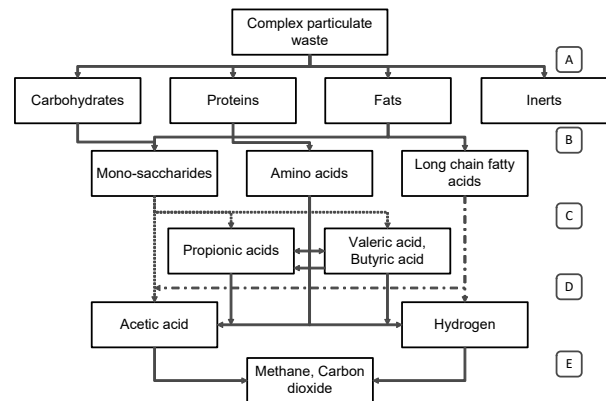


Figure 1: Biogas fermentation process depicted by ADM1 (derived and changed from [3]).

The fermentation is characterized by the following steps: A – Disintegration, B – Hydrolysis, C – Acidogenesis, D – Acetogenesis, E – Methanogenesis. While other models only include the fermentation pro-

cess the ADM1 also considers disintegration and mass-transfer as well as cell death, which also contributes to additional particulate and inert matter (not shown in Figure 1). Therefore the strength of the model is the detailed description of underlying phenomena in the process. Hence the ADM1 is a powerful tool to tune the biogas production by applying optimization strategies. To ensure a broader industrial application of this model, in this paper the optimization potential is analysed and compared to the standard values in the ADM1 by simulation.

1 Materials and Methods

1.1 ADM1 – Model description

The model used in this article is an extended variation of the ADM1 of Batstone 2002. The ADM1 originally consists of 32 ordinary differential equations, while the one used in this work includes 39 state variables. Compared to the original model, equations for lactate and calcium are added. Lactate is one of the fatty acids produced in the process, while calcium is used to describe precipitation, which was not included in former models. The model itself is highly nonlinear and includes all steps shown in Figure 1. Due to the complexity, it shows some numerical inconsistencies in specific ranges. It includes physicochemical processes like gas-liquid and liquid-liquid transfer as well as disintegration steps. The hydrolysis step is divided into carbohydrates, lipids and proteins which are all first order reactions with different rate parameters. Starting parameters are the standard values described in Batstone 2002 [3] with the initial composition of particulate matter as shown in Table 1.

	Carbohydrate	Fat	Protein	Inert
Standard	20 %	25 %	25 %	30 %

Table 1: Composition of the standard substrate

1.2 Simulation and Optimization

The initial values for the optimization approach are taken from the standard substrate in Table 1 and additional parameters can be found in [3]. Simulations are done using the ode15s-solver in Matlab 9.2. If not stated otherwise the optimization is done with the tool fmincon from the Matlab Optimization Toolbox. The

solver is gradient-based, fast and suited for large problems with certain constraints. A scheme of the optimization procedure is shown in Figure 2. The objective function is always the mean of maximum steady state gas flow of methane (q_{CH_4}). The optimized variables are the substrate composition (containing carbohydrates, proteins, fat and inert material) and the applied dilution rate. This leads to a five dimensional optimization problem with the additional constraint that the sum of substrate components is 100%.

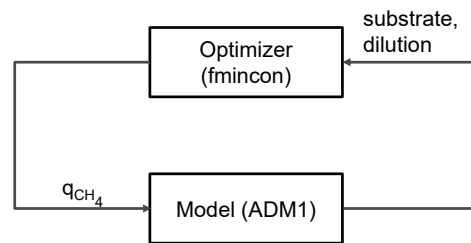


Figure 2: Scheme for optimization with fmincon.

2 Results and Discussion

Due to the diversity in the steps leading to methane the possibilities to optimize the process are manifold. From the available technical parameters the most important one is the dilution- or loading-rate which is directly depending on the substrate composition. Additional parameters are the initial pH-value and the temperature. Nevertheless various inhibitory effects hinder the prediction of optimal operation points. While high protein content is favored for methane production, the ammonia originating from proteins can result in process failure [8]. The high ammonia content is especially affecting methanogenic bacteria [9] in consequence VFAs (volatile fatty acids) are accumulated and the pH drop leads to declining ammonia concentrations. Besides the aforementioned substrate, the temperature plays a main role in all (bio-) chemical reactions and is therefore a valuable optimization parameter. However pH and temperature are not addressed in this study because in the ADM1 the temperature effects are described by the Arrhenius equation only. This makes a detailed analysis of the temperature effects impossible. Additionally it can be concluded that the initial pH-value has only a minor influence. While the other factors are influencing the complete time course of the process, the pH is changing rapidly due to the reactions in the reactor and

reaches equilibrium conditions very fast. Substrate composition and dilution rate have to be optimized simultaneously.

2.1 Substrate composition

The substrate composition is essential in the biogas production process. It is widely acknowledged in literature that it has a big influence on the methane yield [10]. Carbohydrates are degraded faster than fats and proteins indicating higher space-time yields of methane. However this is only the case if the acetogenic steps are equally fast as the methanogenic steps. Otherwise fast accumulation of VFA results in a pH drop inhibiting methane production. The optimization process uses the initial parameters of substrate composition which are given in Table 1. The result of this optimization process is shown in Table 2.

	Carbohydrate	Fat	Protein	Inert
Optimized	0.1 %	13.8 %	86.1 %	0 %

Table 2: Composition of optimized substrate.

It is obvious that 0 % of inert material is favoured. As mentioned earlier high protein contents are also desirable.

However the optimization results are astonishing, because the optimized substrate contains nearly no carbohydrates. This fact is explicable by the fast hydrolysis of carbohydrates leading to high VFA content and a massive pH drop. Another significant fact is the ratio of approximately 6:1 between fat and protein in the optimized substrate. This result matches with literature values, shown in Table 3, demonstrating that the methane content in the biogas is increased significantly by such a ratio [10].

Substrate	Protein/Fat ratio	Methane content
Optimized	6.2	56.8 %
Buckwheat	6.7	57.4 %
Fodder radish	5.7	55.1 %

Table 3: Comparison of the optimized substrate regarding protein/fat ratio with other used substrates [10].

2.2 Critical dilution rate (feeding rate)

The dilution rate describes the flow rate of feed in and out of the biogas fermentation tank. Due to its continuous operation the amount of inflow is equivalent to the outflow. The feed usually contains only substrate, while the outflow contains substrate, product and biomass. Therefore biomass productivity and dilution rate are directly proportional up to the critical dilution rate d_{crit} . The higher the biomass productivity the more methane is produced. Therefore it is critical for process yield. The critical dilution rate defines the point where an increase of inflow leads to wash-out conditions of biomass and a resulting productivity drop. It is obvious that the maximum applicable dilution rate is coupled with the maximum bacterial growth rate. The ADM1 contains seven bacterial species, two of them are producing methane, the first one via the acetoclastic and the second via the hydrogenotrophic pathway (see Figure 1). Complete process failure is only achieved when both of them are washed out, however it is advisable not to interrupt any pathway. With the optimized substrate composition given in Table 2 the nutritional supply is higher and the applicable dilution rate is larger. At a dilution rate of $\sim 0.216 d^{-1}$ complete wash-out condition is reached. An operation of $0.2 d^{-1}$ should be favoured to maintain high methane yields respecting process fluctuations. In Figure 3 the critical dilution rate d_{crit} is indicated by an abrupt productivity drop beyond a dilution rate of $0.216 d^{-1}$.

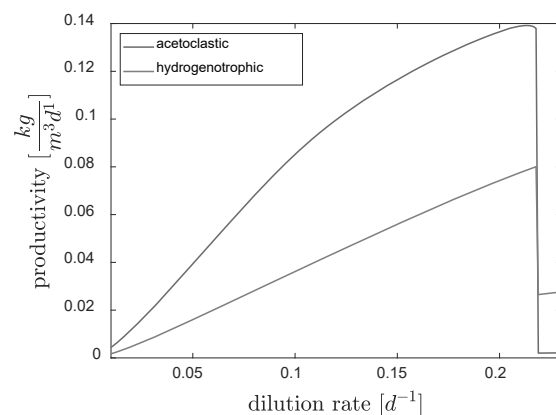


Figure 3: Critical dilution rate for acetoclastic and hydrogenotrophic biomass.

3 Conclusion & Outlook

Many experimental studies are available where different

variables are tested to give highest possible methane yields in biogas plant operation. This is done either by using neural networks [11] or by hand, adding supplements to the process [12]. However, detailed experimental investigations of optimization approaches are only feasible in lab scale. The substrate optimization with the help of the ADM1 is not carried out so far. In this article it is shown that it is possible to find optimal operational parameters with the ADM1 model. The steady state methane yield was increased by ~80% compared to standard values. It has to be mentioned that these are idealized values and it is doubtful that substrates are found fulfilling the optimized nutritional values. However in future applications the ADM1 could be used to define which substrate or substrate mixture should be fed in the biogas reactor. Moreover technical parameters like temperature and pH-value could be incorporated after extending the ADM1. By carrying out these procedures the ADM1 would be a useful tool to increase the biogas production and raise the rentability of biogas plants.

References

- [1] BDEW-Schnellstatistikerhebung Stat. Bundesamt 07/2018
- [2] Siegrist H., Vogt D., Garcia-Heras J.L. and Gujer W. Mathematical model for meso- and thermophilic anaerobic sewage sludge digestion. *Environ. Sci. Technol.* 2002; 36: 1113-1123.
- [3] Batstone D.J., Keller J., Angelidaki I., Kalyuzhnyi S.V., Pavlostathis S.G., Rozzi A., Sanders W.T.M., Siegrist H. and Vavilin V.A. The IWA Anaerobic Digestion Model No. 1 (ADM1). *Water Sci. Technol.* 2002; 45(10): 65-73.
- [4] Parker W.J. Application of the ADM1 model to advanced anaerobic digestion. *Biores. Technol.* 2005; 96(16): 1832-42.
- [5] Batstone D.J., Keller J. Industrial applications of the IWA anaerobic digestion model No. 1 (ADM1). *Water Sci Technol.* 2003; 47(12): 199-206.
- [6] Batstone D.J., Puyol D., Flores-Alsina X., Rodríguez J. Mathematical modelling of anaerobic digestion processes: applications and future needs. *Rev. Environ. Sci. Biotechnol.* 2015; 14(4): 595-613.
- [7] Poggio D., Walker M., Nimmo W., Ma L., Pourkashnian M. Modelling the anaerobic digestion of solid organic waste – Substrate characterization method for ADM1 using a combined biochemical and kinetic estimation approach. *Waste Manage.* 2016; 53: 40-54.
- [8] Chen Y., Cheng J.J., Creamer K.S. Inhibition of anaerobic digestion process: A review. *Biores. Technol.* 2008; 99(10): 4044-64.
- [9] Angelidaki I., Ellegaard L., Ahring B.K. A mathematical model for dynamic simulation of anaerobic digestion of complex substrates: focusing on ammonia inhibition. *Biotechnol. Bioeng.* 1993; 42(2): 159-166.
- [10] Herrmann C., Idler C., Heiermann M. Biogas crops grown in energy crop rotations: Linking chemical composition and methane production characteristics. *Biores. Technol.* 2016; 206: 23-35.
- [11] Gueguim Kana E.B., Oloke J.K., Lateef A., Adesiyun M.O. Modeling and optimization of biogas production on saw dust and other co-substrates using Artificial Neural network and Genetic Algorithm. *Renew. Energy.* 2012; 46: 276-281.
- [12] Liu L., Zhang T., Wan H., Chen Y., Wang X., Yang G., Ren G. Anaerobic co-digestion of animal manure and wheat straw for optimized biogas production by the addition of magnetite and zeolite. *Energy Conserve. Manag.* 2015; 97: 132-139.

Validierung von Näherungsverfahren zur Berechnung von langwelliger Austauschstrahlung im Rauminnen von Gebäuden

Katrin Reblinsky^{1*}, Habbo Heinze²

¹VWA Hochschule für berufsbegleitendes Studium, Wolframstr. 32, 70191 Stuttgart

*katrin.reblinsky@vwa-hochschule.de

²Hochschule Albstadt-Sigmaringen, Anton-Günther-Str. 51/3, 72488 Sigmaringen

Abstract. Zur Steigerung der Energieeffizienz von Gebäuden sind in der Energieeinsparverordnung methodische Bewertungen zwingend vorgegeben. Die aktuell gültigen Normen sowie die zukünftigen harmonisierten Vornormen verleihen der thermischen Simulation von Gebäuden ein stärkeres Gewicht. Aus Effizienz- und Konsistenzgründen werden dabei häufig harmonisierte bauphysikalische Näherungsverfahren eingesetzt, so auch zur Berechnung des langwelligigen thermischen Strahlungsaustausches im Rauminnen. In diesem Beitrag wird bei einem dieser normativ empfohlenen Verfahren, der *Mean Radiant Temperature (MRT)* mit flächengewichteten Sichtfaktoren, die Bezugsveröffentlichung von 1990 mit modernen Simulationswerkzeugen nachgestellt, erweitert und mit den Ergebnissen aus Referenzsimulationsprogrammen verglichen. Es zeigt sich, dass diese Näherung auch weiterhin bei rechenintensiven Simulationen wie *urban scale modelling* oder Simulation mit schlecht bestimmten Eingangsparametern (Behaglichkeitsmodelle) vorteilhaft eingesetzt werden kann.

Einleitung

Um die international vereinbarten Klimaschutzziele zu erreichen, hat die Europäische Union als ein bedeutendes Handlungsfeld die Steigerung der Energieeffizienz von Gebäuden definiert [1]. Zur Umsetzung bedarf es einer methodischen Bewertung, mit deren Erarbeitung die europäischen Normungsorganisationen beauftragt wurden [2]. Die Bundesrepublik Deutschland hat sich zum Ziel gesetzt, mit der Energieeinsparverordnung bis zum Jahr 2050 einen klimaneutralen Gebäudebestand zu erreichen. Die erforderlichen Nachweisverfahren in der DIN 4108 sind mittlerweile so streng formuliert, dass viele bisher übliche konstruktive architektonische Lösungen nicht mehr widerspruchlos umgesetzt werden können, wenn sie mit den schematisierten Verfahren energetisch bewertet werden. Als Alternative wird deshalb zunehmend die ebenfalls zugelas-

sene thermische Simulation von Gebäudebestandteilen genutzt. Allerdings gab es hier bis vor kurzem noch erhebliche Diskussionen darüber, in wieweit die Ergebnisse solcher Simulationen valide sind.

So wurden beispielsweise in der DIN EN ISO 13791 in der Fassung von 2012 konkrete Vorgaben zur Validierung der thermischen Simulationsverfahren gemacht [3]. Mittlerweile gilt es allerdings als akzeptiert, dass die dort zur Validierung vorgeschlagenen Referenzfälle sich teilweise nicht reproduzieren lassen [4]. Im Zuge der Neustrukturierung der Normen zur Reihe DIN EN 52000 [5] wurde deshalb in den Nachfolgeschriften DIN EN 52016 und DIN EN 52017 kein neues Validierungsverfahren benannt [4]. Statt dessen wird auf die Referenzsimulationen der American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) Bezug genommen [6].

Grundsätzlich ist die Fragestellung nach der Realitätstreue von thermischen Simulationen nicht neu. Ingenieurwissenschaftlich wurde die rechnerische Betrachtung gebäudeenergetischer Fragestellungen bereits seit mehreren Dekaden untersucht. Spätestens seit 1980 wurde bauphysikalische Grundlagenarbeit geleistet, um Verfahren zur energetischen Simulation von Gebäuden ingenieurtechnisch praktikabel zu machen [7].

In dieser Veröffentlichung wird ein elementarer aber einflussreicher Berechnungsbaustein der thermischen Simulation näher betrachtet, die Berechnung des energetischen Austauschs von langwelliger Wärmestrahlung im Rauminnen.

Aus Effizienzgründen wurden in den 1980er Jahren vereinfachte Verfahren zur näherungsweise Berechnung des thermischen Strahlungsaustausches von Bauteilen entwickelt. Diese werden gerade wegen ihrer langjährig etablierten Nutzung nach wie vor in komplexeren thermischen Simulationsprogrammen verwen-

det [8].

Dabei ist noch nicht detailliert geklärt, welche Abweichungen diese Näherungen bei der praktischen Berechnung aktueller, moderner Gebäude verursachen. Vielmehr wird nach wie vor auf die ursprünglich in den 1980er Jahren durchgeführten Validierungen der Einzelverfahren Bezug genommen und zusätzlich eine Validierung der Gesamtberechnungen gegen die Referenzsimulationen durchgeführt.

Im Folgenden werden die ursprünglichen Referenzberechnungen der heute vorzugsweise eingesetzten Einzelnäherungsverfahren noch einmal wiederholt und die ursprünglichen Prüffälle erweitert. Dabei werden auch Unklarheiten im Schrifttum diskutiert, die in der praktischen Anwendung zu Unsicherheiten führen. Außerdem werden die Ergebnisse mit Modellrechnungen aus professionellen bauphysikalischen Simulationsprogrammen verglichen, in denen dieselben Näherungsverfahren zum Einsatz kommen.

1 Normativer Hintergrund

Die DIN 4108 stellt Mindestforderungen an den Wärmeschutz, unter anderem gibt sie Grenzwerte für die jährlich akzeptierten Übertemperaturgradstunden vor. Für den Nachweis, dass dieser Grenzwert eingehalten wird, werden nur Randbedingungen für die thermische Simulation vorgegeben, aber keine Details für die Berechnungsmethodik.

Konkrete Vorgaben für die Berechnung wurden allerdings in der DIN EN 13791 gemacht. Im informativen Anhang E wurde die Berechnung der inneren Austauschvorgänge langwelliger Strahlung in Gebäuden anhand eines vereinfachten Verfahrens vorgestellt. Dieses Verfahren basiert auf dem ursprünglich von Walton [9] vorgestellten Algorithmus. Hierbei wird die effektive Gesamtstrahlung an einer Innenfläche eines Raumes dadurch ermittelt, dass jede Oberfläche auf eine fiktive gemittelte Oberfläche strahlt, die aus allen anderen Oberflächen abgeschätzt wird.

2 Vorgehensweise

Das in DIN EN 13791 genannte Näherungsverfahren wurde mit einigen anderen Verfahren systematisch durch Stefanizzi *et al.* [10, 11] bewertet. Hierbei wurden 4 Testfälle definiert, die einzeln berechnet wurden. Diese Veröffentlichung ist nach wie vor der maßgebliche Beleg für die grundsätzliche Anwendbarkeit der

Methode.

Diese Testfälle werden mit moderner Simulationstechnik nachgestellt und erweitert. Anschließend werden auf Basis der Testfälle Modellrechnungen mit einem professionellen Simulationsprogramm durchgeführt. Der langwellige Austausch von Strahlung ist ein häufig vorkommendes Berechnungselement, er wird bei der Gesamtberechnung jedoch hinsichtlich seines Beitrags bewertet. Dadurch soll die tatsächliche Auswirkung der Näherungsverfahren bewertet werden, welche sich einem normalen Anwender nicht ohne weiteres erschließt.

3 Langwelliger Strahlungsaustausch im Gebäudeinnerem

Thermische Simulationen von Gebäuden, die ausdrücklich die Wechselwirkung mit der äußeren Umgebung berücksichtigen, berechnen sehr häufig die Energieströme durch die Gebäudehülle. Es hat sich gezeigt, dass im Gebäudeinneren eine Aufteilung der Wärmeströme in einen konvektiven und einen Strahlungsanteil ein möglicher Ansatz ist [7]. Der Strahlungsanteil wird dabei durch einen rein langwelligeren Strahlungsaustausch zwischen den innenliegenden Oberflächen betrachtet, die als diffuse graue Lambert-Strahler angesehen werden. Es wird also angenommen, dass die Oberflächen in alle Raumrichtungen gleichmäßig abstrahlen. Weiterhin wird angenommen [10], dass jede Oberfläche eine eindeutige Temperatur annimmt und ihre Strahlungsgrößen Emissionsgrad ε , Absorptionsgrad α und Reflexionsgrad ρ unabhängig von Wellenlänge, Strahlungsrichtung und Temperatur sind und die Oberflächen für die betrachtete langwellige Strahlung undurchsichtig sind.

Unter diesen Voraussetzungen läßt sich der langwellige Strahlungsaustausch in einer geschlossenen Gebäudehülle bei bekannten Oberflächentemperaturen und Emissionsgraden exakt berechnen, wozu es mehrere Verfahren gibt [12]. Im Folgenden wird das bei Stefanizzi [10] verwendete Verfahren in der dort verwendeten Notation skizziert. Betrachtet werden N isotherme Innenflächen A_k einer geschlossenen Gebäudehülle mit absoluten Temperaturen T_k und Emissionsgraden ε_k . Für einen stationären isothermen Zustand muss es an jeder Oberfläche A_k einen Wärmestrom

$$Q_k = (q_{ka} - q_{ak})A_k \quad (1)$$

geben, der die zufließende und abfließende Strahlungsenergie beschreibt. Die Größe q_{ka} gibt die totale Einstrahlung in W/m^2 an, die auf der Oberfläche A_k auftrifft und q_{ak} die gleichmäßige totale Abstrahlung, die die Oberfläche verlässt. Damit ergibt sich mit Hilfe des Stefan-Boltzmann-Gesetzes der Zusammenhang

$$q_{ka} = \varepsilon_k \sigma T_k^4 + (1 - \varepsilon_k) q_{ak}. \quad (2)$$

Energieerhaltung liefert die Bilanzgleichung

$$A_k q_{ak} = \sum_{j=1}^N A_j q_{ja} F_{jk}. \quad (3)$$

Die auf A_k auftreffende Energie setzt sich zusammen aus den Anteilen der von allen anderen Flächen abgestrahlten Energie. Dieser Anteil wird durch die rein geometrisch bestimmten Sichtfaktoren F_{jk} wiedergegeben. Sind diese bekannt, so lassen sich aus allen drei Gleichungen (1), (2) und (3) die Zielgrößen Q_k bestimmen. Zweckmäßigerweise [10] geschieht dies durch folgende Bestimmungsgleichung in Matrixform:

$$Q = B^{-1} C. \quad (4)$$

Die Koeffizientenmatrix lautet

$$B_{kj} = \left(\frac{\delta_{kj}}{\varepsilon_j} - \frac{F_{kj}(1 - \varepsilon_j)}{\varepsilon_j} \right) \quad (5)$$

und der Spaltenvektor C hat die Komponenten

$$C_k = \sum_{j=1}^N (\delta_{kj} - F_{kj}) \sigma T_j^4. \quad (6)$$

4 Näherungsverfahren Mean Radiant Temperature Balance

Bereits 1980 führte Walton ein stark vereinfachtes Verfahren zur Bestimmung der Wärmeströme Q_k ein, die die explizite aufwändige Berechnung der geometrischen Sichtfaktoren vermeidet. In seinem Näherungsmodell strahlt jede Oberfläche A_k auf eine fiktive Oberfläche A_f , die die Summe von *allen anderen* Oberflächen darstellt

$$A_f = \sum_{j \neq k}^N A_j. \quad (7)$$

Analog definiert er die dazugehörigen fiktiven Emissionsgrade

$$\varepsilon_f = \left(\sum_{j \neq k}^N A_j \varepsilon_j \right) / A_f \quad (8)$$

und die fiktiven Temperaturen

$$T_f = \left(\sum_{j \neq k}^N A_j \varepsilon_j T_j \right) / (\varepsilon_f A_f). \quad (9)$$

Zu beachten ist [9], dass die betrachtete k -te Oberfläche selbst in den Summen nicht berücksichtigt werden darf, was im Schrifttum [10, 3] aus der Formeldarstellung nicht immer klar hervorgeht und zu deutlichen Abweichungen bei der Berechnung führen kann. Außerdem sind die Temperaturwerte immer als absolute Temperaturwerte in Kelvin zu verstehen.

Die auf der Oberfläche A_k auftreffende Strahlung pro Einheitsfläche wird dann näherungsweise durch folgenden Ausdruck beschrieben:

$$R_k = 4\sigma F_{kf} (T_f - T_k) [(T_f + T_k)/2]^3 \quad (10)$$

wobei der effektive Sichtfaktor sich alleine aus den Flächen und den Emissionsgraden berechnet:

$$F_{kf} = \left[1 + \frac{1 - \varepsilon_k}{\varepsilon_k} + \frac{A_k(1 - \varepsilon_f)}{A_f \varepsilon_f} \right]^{-1}. \quad (11)$$

Die realen Sichtfaktoren stellen per Definition sicher, dass die Strahlungsenergiebilanz stimmt. Durch die Näherung können einzelne Strahlungsanteile unter- oder überbewertet werden, was Walton in seiner MRT-Näherung durch einen Korrekturterm ausgleicht:

$$\tau_{\text{bal}} = \left(\sum_{k=1}^N R_k A_k \right) / \sum_{k=1}^N A_k \quad (12)$$

Aus der effektiven einfallenden Strahlung R_k und dem Korrekturfaktor τ_{bal} lässt sich dann der MRT-Näherungswert für Q_k berechnen:

$$(Q_{\text{MRT}})_k = (R_k - \tau_{\text{bal}}) A_k \quad (13)$$

Zu Gleichung (10) ist anzumerken, dass Walton in seiner Originalformulierung folgende weitere, allgemein bekannte Näherung verwendet, die unabhängig

von der MRT-Näherung ist:

$$T_f^4 - T_i^4 \approx 4 \left(\frac{T_f + T_i}{2} \right)^3 (T_f - T_i) + \Delta \quad (14)$$

Der Fehlerterm Δ hat bei den im Rahmen dieser Untersuchung durchgeführten Berechnungen keine signifikanten Auswirkungen auf die Ergebnisse gehabt (relative Fehler von unter 1%). In der Fachliteratur wird trotz Verweis auf die Originalveröffentlichung [9] diese Näherung teilweise nicht verwendet.

Die eigentliche Stärke in Waltons Variante der MRT-Näherung liegt in der effizienten Berechnung der Sichtfaktoren nach Gleichung (11). Diese gilt exakt für zwei unendlich ausgedehnte konzentrisch ineinander angeordnete Zylinder oder zwei konzentrisch ineinander angeordnete Kugeln [13], S. 589.

5 Testrechnungen

In den frühen Arbeiten zur MRT Näherung stand die effizientere Berechnung durch diese Methode im Vordergrund. Walton selbst definierte ausgehend von einem Kubus eine Reihe von einfachen Testfällen, die der Überprüfung seines Verfahren dienen sollten [9]. Stefanizzi verglich auf Basis dieser Testfälle die MRT Näherung mit den damals gängigen anderen Berechnungsverfahren [10]. Diese Testfälle bilden nach wie vor die Basis für jede Referenzimplementierung [7].

Pedersen, der maßgeblich das von der ASHRAE empfohlene Heat Balance Verfahren zur Thermischen Simulation mit entwickelt hat [7], stützte sich dabei ebenfalls auf die MRT Näherung [14]. Er verwendete einen quaderförmigen Referenzinnenraum mit Stirnflächen von 1m Kantenlänge, der in der Länge von 1m bis 10m variiert wird, siehe Abbildung 1. Hintergrund ist vermutlich [15], dass der Strahlungsaustausch nahe benachbarter Flächen durch den Emissionsgrad dominiert wird, der mit zunehmendem Abstand eine geringere Rolle spielt. Leider sind in [14] keine exemplarischen Zahlenwerte, sondern nur Kurvenverläufe zu den berechneten Wärmeströmen angegeben. Daher soll in dieser Arbeit einige repräsentative Ergebnisse für zukünftige Referenzrechnungen zur Verfügung gestellt werden.

Um den Einfluss von Abschattungen auf die Methode zu bewerten, wird ein Referenzmodell nach Shapiro [16] herangezogen, für das die Sichtfaktoren exakt berechenbar sind. Es handelt sich um zwei Flächen von jeweils 1m², die einen Abstand von ebenfalls 1m aufwei-

sen. Bei 3/4 des Abstands ist eine Abschattungsfläche der Kantenlänge 0,5 m positioniert, siehe Abbildung 2.

Der abschliessende Testfall basiert auf dem Treppenuntergeschoss eines Zweifamilienhauses mit Baujahr 1966, siehe Abbildung 3. Der Raum stellt keine geschlossene Hülle dar und entspricht von seiner Geometrie dem, was sich aus den noch vorhandenen Bauplänen aus der Bauphase ohne größeren Aufwand ablesen läßt. Die angenommene Temperaturverteilung wurde von unkalibrierten Temperatursensormessungen abgeleitet. Dieser Testfall soll die Situation einer Gebäudesimulation mit unvollständigen Informationen widerspiegeln.

Alle Berechnungen erfolgen mit Standardroutinen aus Matlab. Die für das exakte Verfahren notwendigen Sichtfaktoren wurden mit dem Programm von Lauzier [17] berechnet, mit dem auch die Abbildungen der Testgeometrien erstellt wurden, und anschließend mit Werten aus der Literatur verglichen.

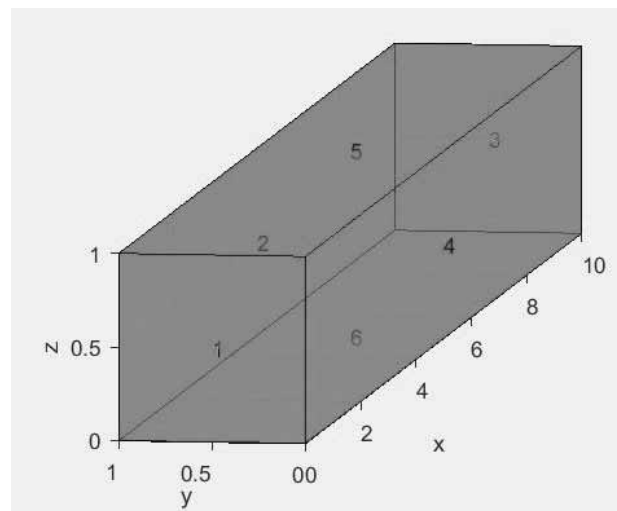


Abbildung 1: Referenzraum von Pedersen mit Stirnflächen von 1m² und Seitenlängen von 1m bis 10m. Der Emissionsgrad für alle Flächen liegt bei $\epsilon = 0,9$. Die Stirnfläche 1 strahlt mit 30°C ab, alle anderen Flächen liegen bei $T = 20^\circ\text{C}$.

6 Diskussion der Ergebnisse

In Tabelle 1 sind für jede der 6 Seiten des Referenzinnenraumes die nach dem MRT Näherungsverfahren (jeweils erste Zeile) und nach dem exakten Verfahren (jeweils zweite Zeile) errechneten Wärmeströme ange-

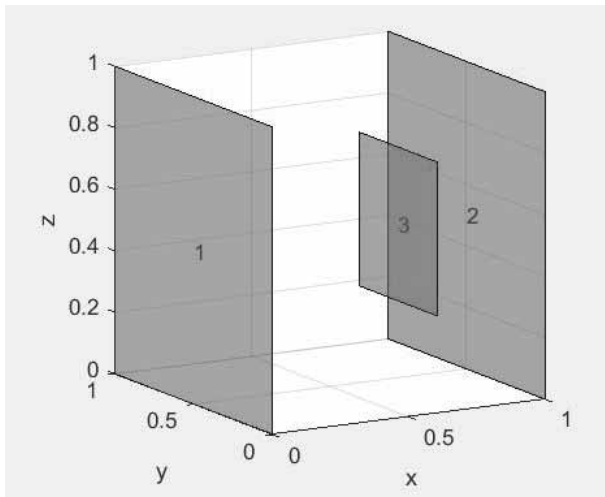


Abbildung 2: Testkonfiguration für sich blockierende quadratischen Sichtflächen nach [16]. Die Hauptflächen haben eine Fläche von 1 m^2 , die Hilfsfläche von $0,25 \text{ m}^2$.

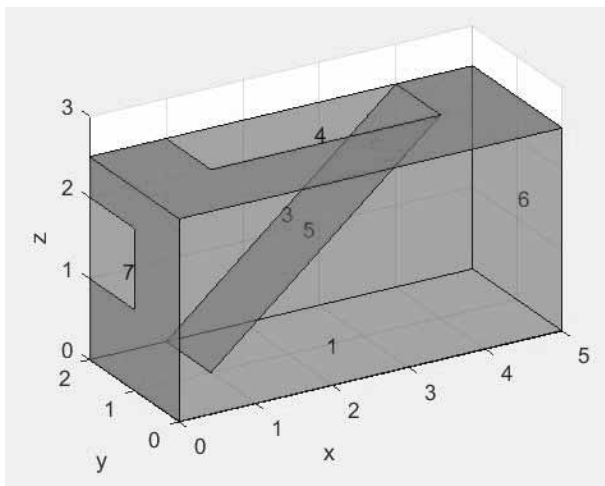


Abbildung 3: Treppenabgang eines Zweifamilienhauses mit Baujahr 1966. Die seitliche Wand (Fläche Nr. 2) ist ausgeblendet. Die Geschosshöhe beträgt $2,5 \text{ m}$. Die Breite der Treppe wurde auf 1 m genähert.

geben. Die Seitenlänge des Raumes wurde dabei zwischen 1 m und 5 m schrittweise um jeweils einen Meter erhöht. Zusätzlich wurde noch der Endpunkt von 10 m berechnet. Es zeigt sich, dass eine gute Übereinstimmung zwischen dem analytisch exakten Verfahren und dem Näherungsverfahren besteht. Pedersen stellte fest [14], dass andere MRT Näherungsverfahren bei erhöh-

tem Seitenverhältnis die Energieerhaltung (Strahlungsbilanz) deutlich verletzen. Das MRT Verfahren nach Walton verhindert dies methodisch durch eine Umverteilung per Korrekturfaktor. Trotzdem zeigt sich auch hier eine systematisch erhöhte Abweichung bei der entfernten Stirnfläche.

Länge in m	Q_{MRT} und Q für Flächen 1 bis 6 in W					
	1	2	3	4	5	6
1	-52,7	10,5	10,5	10,5	10,5	10,5
	-53,1	10,6	10,6	10,6	10,6	10,6
2	-53,8	12,1	5,4	12,1	12,1	12,1
	-53,5	12,4	3,9	12,4	12,4	12,4
3	-54,1	12,6	3,6	12,6	12,6	12,6
	-53,7	12,9	2,0	12,9	12,9	12,9
4	-54,2	12,9	2,7	12,9	12,9	12,9
	-53,8	13,1	1,3	13,1	13,1	13,1
5	-54,2	13,0	2,1	13,0	13,0	13,0
	-53,9	13,2	0,9	13,2	13,2	13,2
10	-54,2	13,3	1,0	13,3	13,3	13,3
	-54,0	13,4	0,3	13,4	13,4	13,4

Tabelle 1: Nettowärmestromdichten Q des Innenraumes aus Abb. 1 für variierte Seitenlängen von 1 m bis 10 m . Die oberen Werte entsprechen Q_{MRT} , die jeweils unteren dem analytisch exakten Vergleichswert Q .

Die Ergebnisse für den Testfall Nr. 2 mit sich blockierenden Sichtflächen finden sich in Tabelle 2. Die Stirnfläche 1 wurde mit einer höheren Temperatur als Abstrahler angenommen, die gegenüberliegende Stirnfläche 2 mit der niedrigsten Temperatur als Strahlungsempfänger. Die Zwischenplatte liegt von den Temperaturverhältnissen dazwischen. An dieser Konfiguration zeigt sich die große Schwäche der MRT-Methode nach Walton: Die von ihm verwendete Näherungsformel für die Sichtfaktoren 11 kann Sichtblockaden nicht darstellen. Die Ergebnisse stimmen nicht einmal qualitativ überein, sondern werden bei der Berechnung durch den Ausgleichsterm dominiert, der somit die Ergebnisse deutlich verzerrt.

Der betrachtete Testfall Nr. 3 ist mit einem höheren Praxisbezug ausgewählt worden. Aus der Literatur ist bekannt [7], dass für annähernd rechtwinklige Räume das MRT Verfahren nach Walton bei korrekter Anwendung ausreichend genaue Ergebnisse liefert. Die Ab-

T in °C	30	10	20
A in m ²	1	1	0,25
ϵ	0,9	0,9	0,9
Q_{MRT}	-85,81	85,62	0,19
Q	-26,57	31,25	-4,69

Tabelle 2: Eingabeparameter und errechnete
Nettowärmestromdichten für den Raum in Abb. 2.

weichung von Testfall Nr. 3 besteht in einer Schrägfläche, die teilweise die Sichtlinien der Flächen behindert. Die verwendeten Eingabeparameter sowie die Ergebnisse der Modellrechnung finden sich in Tabelle 3. Es zeigt sich eine überraschend gute Übereinstimmung für alle Seitenflächen. Bei der Schrägfläche 5 ist die größte Abweichung zu verzeichnen. Dies ist nach den Ergebnissen aus Testfall 2 verständlich, der Ausgleichsmechanismus zeigt hier seine größten Auswirkungen.

Nr.	A m ²	T °C	ϵ	Q_{MRT} W	Q W
1	10	22,5	0,92	242,08	257,3
2	12,5	26,5	0,9	-29,75	-21,93
3	12,5	26,5	0,9	-29,75	-30,19
4	7	29,5	0,95	-156	-164,12
5	3,90512	26,5	0,92	-8,11	-25,43
6	5	26,5	0,9	-10,34	-8,25
7	4	26,5	0,9	-8,13	-7,38

Tabelle 3: Eingabeparameter und errechnete
Nettowärmestromdichten für den Raum in Abb. 3.

7 Zusammenfassung und Ausblick

Thermische Simulationen von Wohn- und Zweckgebäuden haben in den letzten Jahren deutlich an Bedeutung zugenommen und werden mittlerweile routinemäßig praktisch eingesetzt, insbesondere bei der Bewertung von Heiz- und Kühllasten. Eine besondere Bedeutung hat dabei die Modellierung des langwelligeren Strahlungsaustausches in den Gebäudeinnenräumen, die in der Regel bauphysikalisch als Strahlungs-

austausch von grauen Lambertstrahlern erfolgt. In diesem Beitrag wurde ein etabliertes Näherungsverfahren MRT nach Walton verwendet, um an Modellgeometrien Referenzwerte für die Wärmeströme der einzelnen Oberflächen zu berechnen. Es wurden relevante Testfälle aus früheren Untersuchungen reproduziert und als Zahlenwerte bereit gestellt. Außerdem wurden die Grenzen des Verfahrens bei sichtbehindernden Konfigurationen demonstriert. Schließlich wurde an einem realistischen Praxisbeispiel die einfache Anwendbarkeit mit der Standardsoftware Matlab demonstriert.

Das ausbalancierte MRT Verfahren nach Walton zeichnet sich dadurch aus, dass es nicht zunächst anhand der Raumgeometrie die Sichtfaktoren der jeweiligen Flächen berechnen muss, sondern diese alleine anhand der Absolutflächenwerte selbst abschätzt. Dies ist besonders vorteilhaft, wenn rechenintensive Simulationen durchgeführt werden. Aktuelle Forschungsansätze zur Modellierung von ganzen Stadtvierteln zeigen, dass der langwellige interne Strahlungsaustausch einen relevanten Einfluss auf die Ergebnisgüte und die Rechenzeit hat [18], hier also Bedarf für ein Näherungsverfahren besteht.

Außerdem zeigt das Verfahren seine Stärke dann, wenn gerade keine genauen Informationen über die tatsächliche Geometrie vorliegen. Dies kann hilfreich bei der Berücksichtigung von Mobiliar und weiteren Einrichtungsgegenständen wie z.B. Unterhaltungselektronik sein, die als Strahlungspartner bei modernen Passivhausstrukturen zunehmend an Bedeutung gewinnen. Es ist daher durchaus denkbar, dass dieses Mittelungsverfahren zukünftig auch bei der Modellierung von Behaglichkeit vermehrt eingesetzt wird [19].

Literatur

- [1] Europäische Union. Richtlinie 2010/31/EU über die Gesamtenergieeffizienz von Gebäuden (Neufassung). EUR-Lex. 2010.
- [2] Europäische Kommission. Auftrag M/480 an CEN, CENELEC und ETSI zur Erarbeitung und Annahme von Normen für eine Methodik zur Berechnung der integrierten Gesamtenergieeffizienz von Gebäuden sowie zur Förderung der Energieeffizienz von Gebäuden gemäß der Neufassung der Richtlinie 2010/31/EU über die Gesamtenergieeffizienz von Gebäuden vom 14. Dezember 2010. EUR-Lex. 2010.
- [3] DIN. Wärmetechnisches Verhalten von Gebäuden – Sommerliche Raumtemperaturen bei Gebäuden ohne Anlagentechnik – Allgemeine Kriterien und

- Validierungsverfahren. *DIN EN ISO 13791:2012*, Deutsches Institut für Normung e. V., Berlin, Deutschland. 2012.
- [4] DIN. Energieeffizienz von Gebäuden – Fühlbare und latente Wärmelasten und Innentemperaturen – Teil 1: Allgemeine Berechnungsverfahren. *DIN EN ISO 52017-1*, Deutsches Institut für Normung e. V., Berlin, Deutschland. 2018.
- [5] DIN. Energieeffizienz von Gebäuden – Festlegungen zur Bewertung der Energieeffizienz von Gebäuden – Teil 1: Allgemeiner Rahmen und Verfahren. *DIN EN ISO 52000-1*, Deutsches Institut für Normung e. V., Berlin, Deutschland. 2017.
- [6] ASHRAE. Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs. *ANSI/ASHRAE Standard 140*, Atlanta, GA, USA. 2014.
- [7] Rees S, Spitler J, Davies M, Haves P. Qualitative Comparison of North American and U.K. Cooling Load Calculation Methods. *International Journal of Heating, Ventilating, Air-Conditioning and Refrigeration Research*. 2000;6(1):75–99.
- [8] Kieltyka M. Dynamische Kühllastsimulation nach ASHRAE. *Moderne Gebäudetechnik*. 2016;(9):26–29.
- [9] Walton G. A New Algorithm for Radiant Interchange in Room Loads Calculations. *ASHRAE Transactions*. 1980;86(2):190–208.
- [10] Stefanizzi P, Wilson A, Pinney A. Internal Long-Wave Radiation Exchange in Buildings: Comparison of Calculation Methods: I Review of Algorithms. *Building Services Engineering Research and Technology*. 1990; 11(3):81–85.
- [11] Stefanizzi P, Wilson A, Pinney A. Internal Long-Wave Radiation Exchange in Buildings: Comparison of Calculation Methods: II Testing of Algorithms. *Building Services Engineering Research and Technology*. 1990;11(3):87–96.
- [12] Baehr HD, Stephan K. *Wärme- und Stoffübertragung*. Berlin Heidelberg: Springer-Vieweg, 9th ed. 2016.
- [13] Kreith F, Manglik RM, Bohn MS. *Principles of Heat Transfer*. Cengage Learning, 7th ed. 2011.
- [14] Liesen R, Pedersen C. An evaluation of inside surface heat balance models for cooling load calculations. *ASHRAE Transactions*. 1997;103(2):485–502.
- [15] Davies MG. A review of some methods for calculating internal longwave radiation exchange in buildings. *Building Services Engineering Research and Technology*. 1992;13(2):95–99.
- [16] Shapiro A. Computer Implementation, Accuracy and Timing of Radiation View-Factor Algorithms. *Technical Report UCRL-89602*, ASME, Boston, MA, USA. 1983.
- [17] Lauzier N. View Factors and Radiative Heat Transfer. de.mathworks.com/matlabcentral/fileexchange/5664-view-factors. 2004.
- [18] Lauster M, Remmen P, Fuchs M, Teichmann J, Streblov R, Müller D. Modelling long-wave radiation heat exchange for thermal network building simulations at urban scale using Modelica. In: *Proceedings of the 10th International Modelica Conference*, edited by Tummescheit H, Arzen KE. 96th Linköping Electronic Conference, Lund; Sweden: Linköping University Electronic Press. 2014 Mar; pp. 125–133. Doi: 10.3384/ecp14096.
- [19] Stratbücker S, Park S, Bolineni S. Effiziente Strahlungssimulation für ein thermisches Behaglichkeitsmodell. In: *Proceedings of BauSIM 2012*. 4th German-Austrian IBPSA Conference, Berlin University of the Arts: IBPSA. 2012; pp. 445–452.

Towards an Open Simulation Scenario Infrastructure

Bikash C. Karmokar^{1*}, Bharvi Chhaya²,

Umut Durak^{3,4}, Shafagh Jafer², Sven Hartmann⁴

¹Leibniz Universität Hannover, Welfengarten 1, Hannover, 30167, Germany;

**bikash.chandra.karmokar@stud.uni-hannover.de*

²Embry-Riddle Aeronautical University, 600 S Clyde Morris Blvd, Daytona Beach, Florida 32114, United States

³German Aerospace Center (DLR), Institute of Flight Systems, Braunschweig, 38108, Germany

⁴Clausthal University of Technology, Clausthal-Zellerfeld, 38678, Germany

Abstract. Scenarios are crucial elements in a simulation study. Typical scenario management activities include scenario development, loading and parsing, distribution and role casting, event injection, and finally data collection and logging. Recently, significant effort is being spent towards standardization in simulation scenario development. A standard simulation scenario definition language requires an underlying metamodel. This paper presents a System Entity Structure (SES) based meta-modeling approach for achieving a standard scenario definition language as a significant step towards open scenario infrastructure.

1 Introduction

A model is a formal specification of a concept, and a simulation is a specific implementation or execution of a model [1]. The execution of any simulation requires a clearly-defined scenario. A simulation scenario is usually defined as the specification of a set of properties, which are (1) initial and terminal conditions, (2) significant events and their time line, (3) the environment, as well as (4) the major entities, their capabilities, behavior and interactions over time.

Scenarios are important artifacts in the simulation engineering process [2]. They not only define specification of a simulation run, but also provide an input for the design and evaluation of the simulation environment itself [3].

Scenario management is required due to the extensive need for scenarios in simulation. Scenario management includes all activities from the development to the execution of the scenario [4]. These are classified

into two categories, the activities that take place during (1) design and development of the scenario, and (2) the execution of the scenario [5].

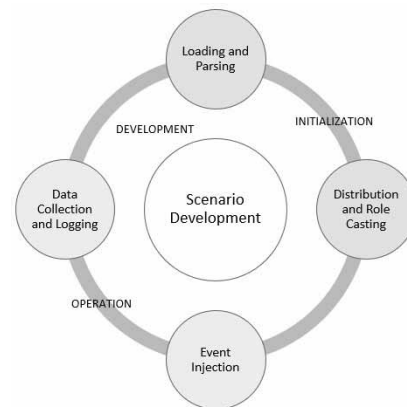


Figure 1: Scenario management activities.

Figure 1 shows the activities associated with scenario management. Scenario development is the only major activity included in the design and development category. During execution, the various activities involved are (1) scenario loading and parsing, (2) scenario distribution and role parsing, (3) event injection, and (4) scenario-related data collection and logging [5].

Scenario management is currently an extensive process due to the lack of common understanding, standardized practice and common tools. The need for standardization and an open infrastructure for scenario management is apparent. There are recent studies in aviation [6, 7] and in automotive [8] towards standardization in simulation scenario development. This paper presents a System Entity Structure (SES) based meta-

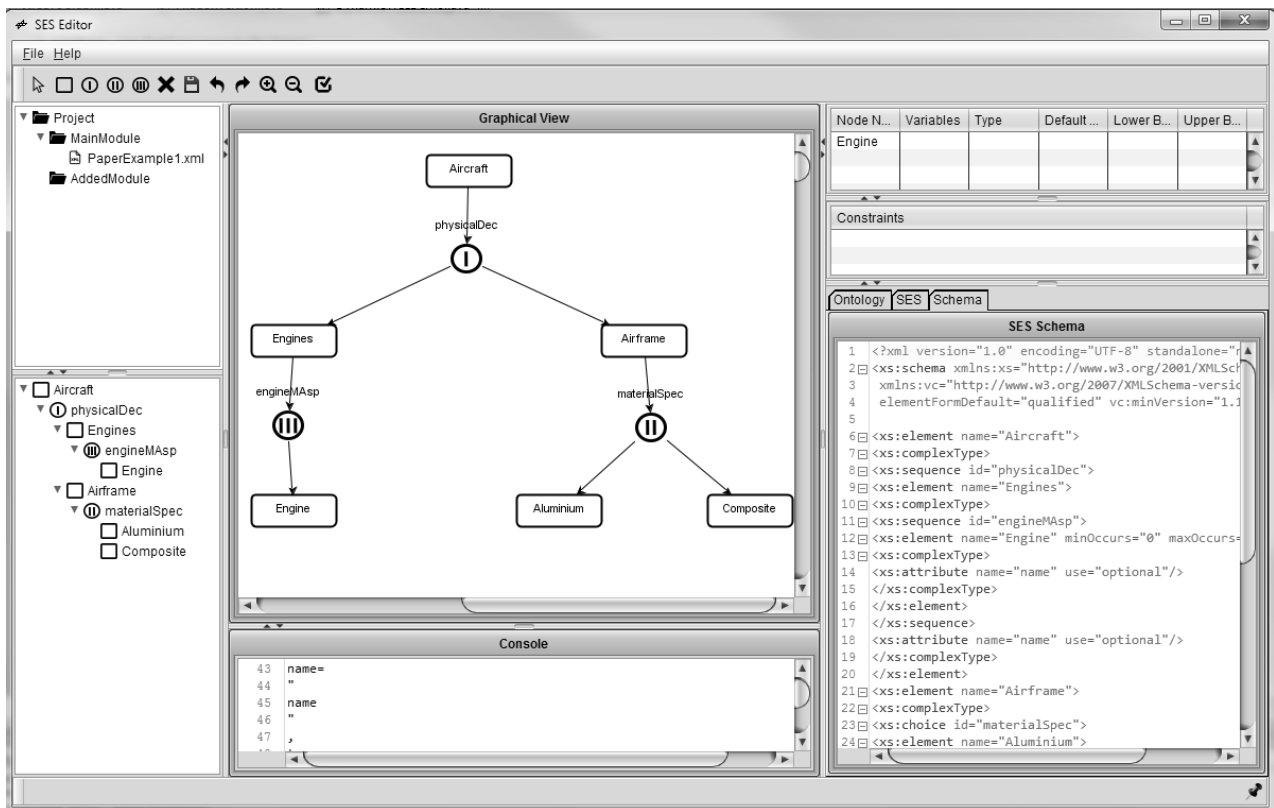


Figure 2: SES Meta-model in SESEditor.

modeling approach and supporting tooling for achieving a standard scenario definition language. The approach has first been proposed by Durak et.al [9] as a significant step towards open scenario infrastructure. The tooling is still under development.

2 Scenario Development

Scenario development is viewed as the transformation of operational scenarios (defined using natural language) to conceptual scenarios (conforming to a meta-model), and eventually to executable scenarios (specified using a scenario specification language). As the development process occurs, the scenario models are refined and transformed [10, 3]. Scenario development starts with operational scenarios, mostly in text form constructed in natural language by domain experts. They usually present the key information from the user's perspective and are often not complete and consistent. Operational scenarios are then used to develop conceptual scenarios, essentially to be based on a metamodel, either an explicit or an implicit one. The

result is then a complete and a consistent scenario specification, often compiled in a table. Finally conceptual scenarios are transformed into executable scenarios for target simulation environments using a set of rules specific to the target simulation environment.

3 Standardization in Scenario Development

In order to standardize the scenario development process, we propose standard metamodels which can be used as a basis for creating conceptual scenarios.

Recently, an effort has begun to develop a formal scenario definition language for aviation [11]. The Aviation Scenario Definition Language (ASDL)[6] is a domain-specific language (DSL) which aims to provide a simple, standardized method for aviation simulation scenario generation. ASDL aims to provide a common mechanism for verifying and executing aviation scenarios, allow for effective sharing of scenarios among various simulation environments, improve the consistency of different simulators and simulations, and enable the

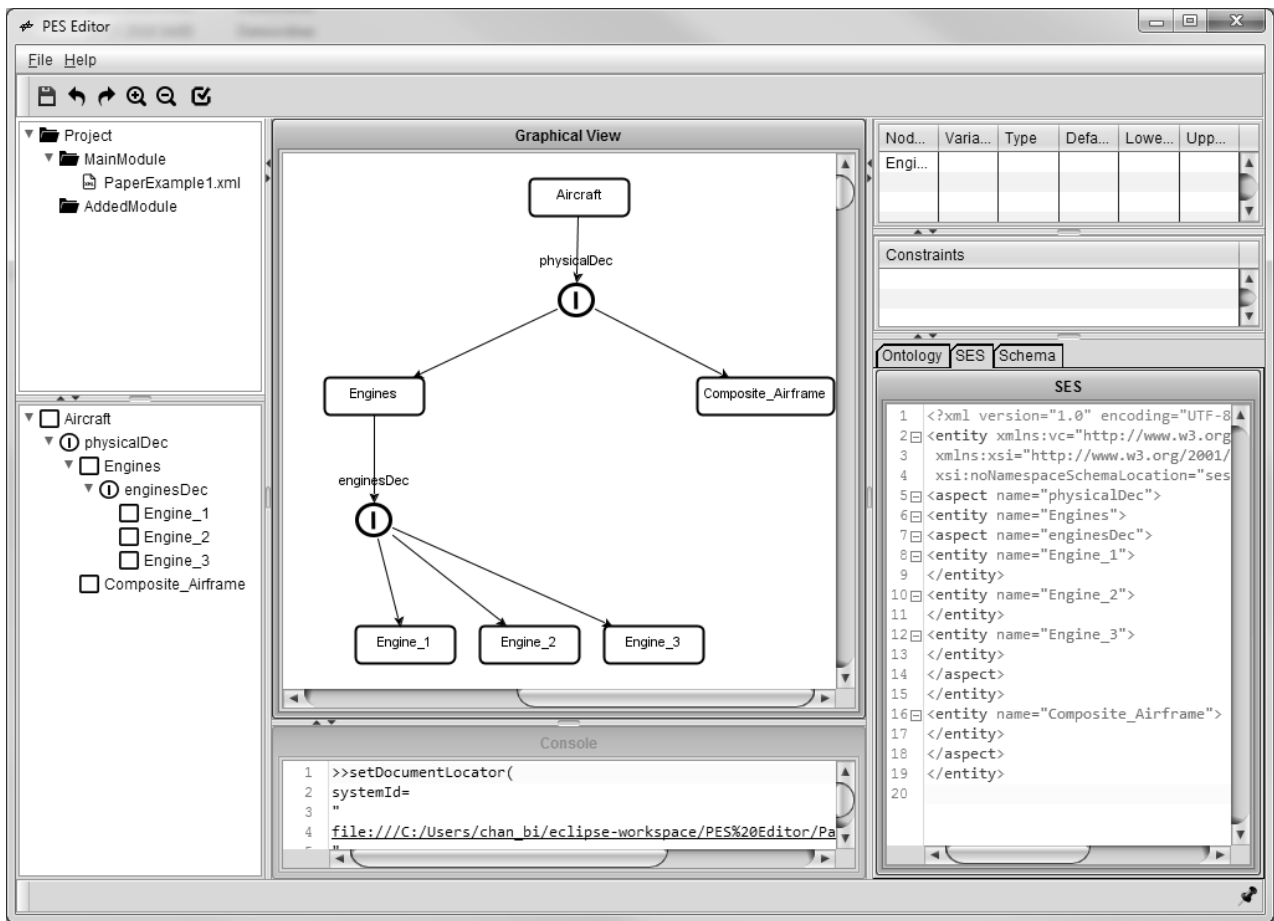


Figure 3: Pruning example in PESEditor.

reuse of scenario specifications.

In the automotive industry, OpenSCENARIO aims to establish a software-independent format of describing dynamic scenarios for virtual driving tests [12, 8]. This standardized scenario format would have potential applications in a wide range of automotive uses such as driver assistance, autonomous driving, test of sensors and algorithms and simulating driving dynamics.

4 System Entity Structure

Conceptual scenarios can be defined and described either graphically or textually using a scenario definition language. Scenario definition language can be introduced as a Domain Specific Language (DSL). Metamodels are used to define DSLs. One such method of describing metamodels is System Entity Structure [13]. SES is a high-level ontology which was introduced for

knowledge representation of decomposition, taxonomy and coupling of systems [14]. SES has a limited set of elements and axioms which makes it an accessible metamodeling approach.

There are four elements of SES [15], which are (1) Entity, (2) Aspect, (3) Specialization and (4) Multiple-Aspect. Entity is an object of interest, and can also have variables attached to it. An Aspect is used to denote a decomposition relationship. The taxonomy of an Entity can be represented by using Specialization. Multi-Aspect is a special kind of Aspect, which represents a multiplicity relationship that specifies the parent Entity as a composition of multiple Entities of the same type.

There are six axioms of SES [16], which are (1) uniformity, (2) strict hierarchy, (3) alternating mode, (4) valid brothers, (5) attached variables, and (6) inheritance. Uniformity confirms that any two nodes with the same label have isomorphic subtrees. Strict hierarchy is used to prevent a label from appearing more

than once down any path of the tree. Alternating mode recommends that if a node is an Entity, then the successor is either Aspect or Specialization, and vice versa. The valid brothers axiom disallows two brothers from having the same label. Attached variables specifies a constraint that variable types attached to the same item shall have distinct names. According to inheritance, it is indicated that a Specialization inherits all variables and Aspects.

Pruning is described as resolving the choices in Aspect, Multi-Aspect and Specialization relations and assigning values to the variables. It also specifies the cardinalities in Multi-Aspect relations. The outcome of pruning is called Pruned Entity Structure (PES), which is a selection-free tree.

Metamodels can be developed using SES following the approach proposed in [9]. The resulting Scenario SES captures all scenario elements and their relationships. SES specifies the simulation scenario definition language, and its pruning yield a Pruned Entity Structure (PES), which represents a particular scenario. The computational representation of SES and PES for scenario development has been proposed in [17]

5 Using SES Editor for Metamodeling

In SESEditor, SES is represented by a directed tree structure. Here, objects are represented by nodes which are connected using edges. Figure 2 shows an SES model created in SESEditor, where the root entity Aircraft is decomposed into Engines and Airframe using physicalDec aspect node. MultiAspect node enginesMAsp decomposes Engines to Engine, representing that Engines is made of multiple instances of Engine. Specialization node materialSpec is used to express that Airframe can be Aluminum or Composite.

6 Using PES Editor for Pruning

Figure 3 shows a PES of Aircraft Metamodel in PESEditor. During pruning, MultiAspect node enginesMAsp is pruned and aspect node enginesDec is added with children Engine_1, Engine_2 and Engine_3 of the same type as Engine. From the available options, specialization node is pruned and Composite_Airframe is added. Thus the completely pruned structure is created where there are no choices left.

7 Conclusions

The scenario management process consists of scenario development and several activities during scenario execution. A brief overview of the current effort towards standardizing scenario development in the aviation and automotive industries was presented here. The paper highlighted the use of System Entity Structure (SES) and its pruned counterpart PES for metamodeling scenario specification languages and for creating specific scenarios from it. The use of SESEditor and PESEditor was demonstrated to display a small metamodel of an aircraft entity. These editors allow graphical representation of metamodels and pruning for effective scenario generation.

References

- [1] Sargent RG. Verification and validation of simulation models. In: *Simulation Conference (WSC), Proceedings of the 2009 Winter*. IEEE. 2009; pp. 162–176.
- [2] IEEE. IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP). *IEEE Std 1730-2010*. 2011;.
- [3] Durak U, Topçu O, Siegfried R, Oguztuzun H. Scenario development: A model-driven engineering perspective. In: *2014 International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)*. IEEE. 2014; pp. 117–124.
- [4] Löfstrand B, Ericsson M, Johansson M, Strand J, Lepp H. Scenario management-Common design principles and data interchange formats. In: *European Simulation Interoperability Workshop (SIW), 04E-SIW-070*, vol. 46. 2004; .
- [5] Topçu O, Durak U, Oğuztüzün H, Yılmaz L. *Distributed simulation: A model driven engineering approach*. Springer. 2016.
- [6] Jafer S, Chhaya B, Durak U, Gerlach T. Formal scenario definition language for aviation: aircraft landing case study. In: *AIAA Modeling and Simulation Technologies Conference*. 2016; p. 3521.
- [7] Durak U, Jafer S, Beard SD, Reardon S, Murphy JR, Crider DA, Gerretsen A, Lenz H, Macchiarella ND, Rigby KT, et al. Towards a Standardization for Simulation Scenario Development in Aviation-Panel Discussion. In: *2018 AIAA Modeling and Simulation Technologies Conference*. 2018; p. 1395.
- [8] OpenSCENARIO bringing content to the road. www.openscenario.org/. Accessed: 2018-07-25.

- [9] Durak U, Pruter I, Gerlach T, Jafer S, Pawletta T, Hartmann S. Using System Entity Structures to model the elements of a scenario in a research flight simulator. In: *AIAA Modeling and Simulation Technologies Conference*. 2017; p. 1076.
- [10] Jafer S, Chhaya B, Updegrove J, Durak U. Schema-based Ontological Representations of a Domain-Specific Scenario Modeling Language. *Journal of Simulation Engineering*. 2018;1.
- [11] Reardon SE, Beard SD, Lewis E. Scenario Development at the Vertical Motion Simulator. 2017;.
- [12] Barsi A, Poto V, Tihanyi V. Creating OpenCRG Road Surface Model from Terrestrial Laser Scanning Data for Autonomous Vehicles. In: *Vehicle and Automotive Engineering*. Springer. 2018; pp. 361–369.
- [13] Durak U, Pawletta T, Oguztuzun H, Zeigler BP. System entity structure and model base framework in model based engineering of simulations for technical systems. In: *Proceedings of the Symposium on Model-driven Approaches for Simulation Engineering*. Society for Computer Simulation International. 2017; p. 1.
- [14] Kim TG, Lee C, Christensen ER, Zeigler BP. System entity structuring and model base management. *IEEE Transactions on Systems Man and Cybernetics*. 1990; 20(5):1013–1024.
- [15] Zeigler BP, Hammonds PE. *Modeling and simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange*. Elsevier. 2007.
- [16] Zeigler BP. Multifaceted modelling and discrete event simulation. 1984;.
- [17] Durak U, Jafer S, Wittman R, Mittal S, Hartmann S, Zeigler BP. Computational Representation for a Simulation Scenario Definition Language. In: *AIAA SciTech Forum*. 2018;. URL <https://doi.org/10.2514/6.2018-1398>

How to Define SES Trees for Variability Modeling

Christina Deatcu^{1*}, Hendrik Folkerts¹, Thorsten Pawletta¹, Umut Durak²

¹Hochschule Wismar - University of Applied Sciences, Research Group CEA, Philipp-Müller-Straße 14, 23966 Wismar, Germany; *christina.deatcu@hs-wismar.de

²Institute of Flight Systems, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany

Abstract. The System Entity Structure (SES) is a high level approach for variability modeling, particularly in simulation engineering, which is under continuous development. In this context, an enhanced framework is introduced that supports dynamic variability evolution using the SES approach. However, the main focus is to start a discussion about a set of design patterns, which were developed to analyze the tree design and computing aspects of System Entity Structures. As development of our MATLAB-based SES toolbox for construction and pruning of SES trees proceeded, the necessity to have some generalized examples for testing and verification came more and more into awareness. We propose a set of design patterns that, if completely representable and computable by a certain tool, support all aspects of SES theory. In addition, the patterns give users substantial support for developing SES models for other applications.

Introduction

This paper is a modified version of [1]. It details the specification of basic design patterns and introduces more advanced combined patterns.

Generally, variability modeling can be seen as an approach to describe more than one system configuration. According to Capilla, Bosch, and Kang [2], a software variability model has to describe the commonality and variability of a system at all stages of the software lifecycle. In software engineering, variability modeling is often closely associated with product lines. For software product lines the variability is described explicitly. Variation points are defined where different solutions can be derived. Such variability mechanisms can be specified at different levels of abstraction, ranging from requirements specification to source code implementation. A popular high level approach is feature modeling by means of feature models, which were introduced as part of Feature-Oriented Domain Analysis by Kang et al. [3] and subsequently extended and used in various ways. An important further development of variability modeling has been the notion of variability in time, known as binding time in product line engineering [4]. That means that variability can be realized from design time to runtime.

In simulation engineering, the problem of variability modeling is well known from the eighties. One of the first high level approaches for variability modeling in the design phase was introduced with the System Entity Structure (SES) by Zeigler [5]. The objective was to describe a set of system configurations for a family of systems. An SES is represented by a tree structure, which describes a set of modular, hierarchical system structures, defines references to basic models in a model base (MB) and specifies various parameter settings for the referenced basic models. In addition, the approach defines several abstract transformation methods for deriving a particular system configuration and for generating an executable simulation model [6]. The entire approach was continuously further developed by Zeigler and many other researchers, such as in [7], [8], and [9].

As another approach for describing system structures and their configurations, various XML based composition schemes, which distinguish between interfaces and the concrete implementations of the models were proposed, such as by Röhl and Uhrmacher [10] or by Wang and Wainer [11]. Although that research does not explicitly address model families as the SES/MB approach does, some of the basic ideas are similar. Moreover, the ideas in [10, 11] are important for the design of reusable model components and their organization in an MB. In this paper, the reusability of components in an MB is not discussed. It focuses on modeling system variability using the SES and the necessary software framework.

The problem of variability at runtime is known as variable or dynamic structure system modeling and simulation. Analogous to the approaches in software engineering, such a dynamic variability can be described on the level of a specific model, such as introduced by Barros in [12], or separated using a higher level model abstraction in conjunction with an appropriate software framework. Regarding this, a first theoretical approach for dynamic variability modeling using the SES/MB method was published in 1990 [13].

Based on this early idea, a prototype of a full SES/MB

based modeling and simulation infrastructure has been developed and implemented within MATLAB/Simulink by the authors [14, 15], and a Python implementation is in progress. In addition, the SES theory has been enhanced. In this context, the necessity to have some generalized SES patterns came more and more into awareness. On the one hand patterns are helpful during software development and on the other hand the patterns are expected to give users substantial support for developing SES models for their applications.

After a short overview to the enhanced SES ontology and the enhanced SES/MB based infrastructure, basic design patterns for variability modeling are introduced. The description is related to the modeling capabilities provided by feature models. Then, some combined patterns are discussed exemplarily to give an impression for advanced variability modeling possibilities using the SES and its extensions. Finally, the main results are summarized and an outlook to future work is given.

1 Background

According to Zeigler and Hammonds, the SES is an ontology, a language with syntax and semantics to represent declarative knowledge [8]. It is particularly suitable for describing system configurations for different application domains. An SES is represented by a directed tree structure. Objects are represented by nodes which are connected by edges. There are four node types with different properties describing the objects and their relations. Furthermore, there are axioms for defining the SES correctly. Since an SES describes a number of system configurations, the SES tree needs to be *pruned* to get one particular configuration, which is called Pruned Entity Structure (PES).

The classic SES theory was extended by several researchers over the last decades. In [14] and [16] the SES theory was extended with a procedural knowledge representation. Some of these extensions are used in this paper. A comprehensive example on how the pruning patterns proposed in this paper can be used, is demonstrated in [17].

1.1 Node Types

Among the four node types, there are two groups, the entity nodes and the descriptive nodes. Entity nodes describe objects of the real or the imaginary world. The root and the leaves of an SES tree are always entity nodes.

Relations between the entity nodes are specified by descriptive nodes.

Descriptive nodes are the genus for aspect nodes, specialization nodes and multi-aspect nodes. Aspect nodes (name suffix *DEC*) describe how entity nodes can be decomposed in partial entities whereas the taxonomy of an entity is described by specialization nodes (name suffix *SPEC*). Multi-aspect nodes (name suffix *MASP*) are a special case of an aspect node with all children being of the same kind.

Each node or edge can have attached variables, also called attributes. For entity nodes, the variables represent properties of the respective object whereas the variables at descriptive nodes specify relations between their parent node and children nodes or decisions for the pruning process. With the extended procedural knowledge representation, values of attached variables can be assigned dynamically.

1.2 Axioms

The semantics of the SES are defined by axioms. The types of the nodes have to follow the axiom *alternating mode*. Every entity node has to be followed by a descriptive node, and vice versa. A *strict hierarchy* is needed. In every path of the tree, a name of a node may occur only once. If nodes in different paths have the same name, they need to have the same variables and isomorphic partial trees. This is called *uniformity*. Nodes on the same level of hierarchy and having the same father, called sibling nodes, have to be *valid brothers*, meaning that sibling nodes must not have the same name. The axiom of *attached variables* implies that a node must not have variables of the same name. The axiom of *inheritance* implies, that during pruning, the parent and the child of a specialization combine their attributes. If parent and child have the same attributes, the parent's attributes are overwritten with the child's attributes and their values.

1.3 Extended SES/MB Infrastructure

The SES describing a set of system designs has been associated with the idea of model generation of modular, hierarchical systems from the very beginning [6] which led to the SES/MB approach. Each system design is defined by its system structure and parameter configuration in the SES. The core assets of all system variants are specified as a set of configurable basic models, which are organized in a Model Base (MB). The classic SES/MB framework defines a set of transformation methods for

generating executable simulation models, but automated model generation is not provided. To allow automated generation and execution of models, the SES/MB approach has been extended ([14], [15], [16]). These extensions make the SES/MB approach more pragmatic for implementation and to be used in a simulation infrastructure.

Figure 1 depicts the extended SES/MB infrastructure consisting of the SES/MB framework, an *Execution Unit*, and an *Experiment Control*. Although the SES/MB approach and its extensions are usually considered in connection with the generation of simulation models, they are generally applicable to modular-hierarchical structured software systems.

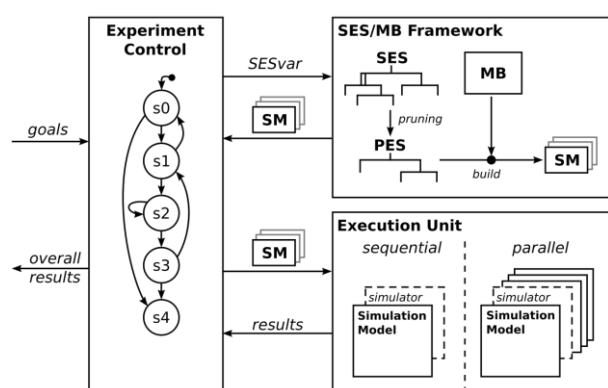


Figure 1: Extended SES/MB-based infrastructure.

Operations. On the SES, a *merge operation* is defined allowing two or more SES to be combined. This allows the quick reuse of a once defined SES. The essential operation on the SES is the *pruning method*. To extract one particular system structure and configuration, the SES needs to be trimmed to a PES. During the pruning process, decisions have to be taken at descriptive nodes. Therefore, rules need to be defined at aspect, multi-aspect and specialization nodes. The *specialization rule* (specrule) associated with a specialization node determines which child entity shall be selected. *Aspect rules* (aspectrule) associated with aspect or multi-aspect nodes on the same hierarchy level determine which of the siblings is to be chosen. Furthermore, cross-tree relations can be expressed by *selection constraints*. Selection constraints can be used to select a certain entity based on decisions taken anywhere else in the SES tree. Next to the pruning method, another transformation method is the *build method*. With the help of the build method, an executable model can be built from a PES and basic models

organized in an MB. The basic models are specific for a certain simulation software. Therefore, the build method needs to match to the simulator used.

Execution Unit and Experiment Control. For automated and reactive processing of SES models, an execution unit and an overall experiment control unit are added to the framework, as depicted in Figure 1. For automatic generation of different PES, leading to different simulation models, an interface to the SES is needed. This interface can be established by global variables of the SES, called *SES Variables* (SESvar), which can affect the decisions taken in descriptive nodes during pruning. Thus, a particular system configuration derived from an SES depends on the current settings of the SES variables. The value range of SES variables can be limited by defining *semantic conditions*, which are checked before pruning to exclude certain system configurations. By assigning values to the SES variables, the experiment control determines the order and system configurations of executable simulation models (SM) to generate from the SES with the pruning and build operations. Thereby different variants of the executable simulation models are generated. The experiment control then transmits the SM to the execution unit. The execution unit links the generated simulation model to the simulator, executes a simulation run and, finally, sends the results back to the experiment control. The results, in turn, can influence the decision of the experiment control on how to assign the SES variables next.

Special Attributes. Combining basic models from the MB leads to the creation of *coupled models*. In order to describe the structure of the executable model, some nodes need to define *couplings*. Couplings are properties of descriptive nodes of the type aspect and multi-aspect and consist of pairs of entity names and port names. Figure 2 gives an impression of what a definition of couplings may look like. Furthermore, for a multi-aspect node, a special variable, *numRep*, has to be defined representing the number of children to generate when pruning this node. To specify the basic model from the MB an entity node refers to, the *mb-attribute* is introduced. This special attribute is permitted just for leaf nodes. Finally, for some cases, it is necessary to define priorities for supporting decisions among descriptive nodes on the same level of hierarchy in the *priority attribute*. All values of attributes can be defined by constants or set via

SES variables or SES Functions.

SES Functions. The concept of *SES Functions* (SESfcn) has been introduced to specify complex variability within node attributes with minimal effort and to keep a lean SES tree. Typical examples include the definition of varying coupling relations, varying port numbers of systems or the definition of variable parameter configurations in attributes. During pruning, SES functions are evaluated, often with SES variables as input parameters. For effective coding of SES functions, the implicit attributes *parent* and *children* are introduced for each SES node. They encode the parent and children node names, respectively.

1.4 Software Tools for the Extended SES/MB Infrastructure

In the Research Group CEA, a prototype tool for the SES/MB infrastructure was developed, *The SES Toolbox for Matlab/Simulink* [15]. Currently, SES trees can be defined via a graphical user interface and a concrete variant can be extracted by pruning. The toolbox supports the modeler with plausibility test during SES construction, graphical representation of the SES, automatic generation of HTML documentation, and other features. The pruning process can be started from the graphical user interface and, in addition, is implemented to function automatically. Automatic pruning is necessary when using an SES constructed with the toolbox together with the experiment control. Furthermore, there is a prototype Matlab function implementing a build method for the simulation software Simulink, including SimEvents and Simscape, the MatlabDEVs toolbox [18], and for Modelica models. The SES is linked to the appropriate MB with the special mb-attribute of the leaf entity nodes.

Another software tool based on Python3/PyQt5 is under development. The aim is to be more independent from a computing environment and to support a greater number of simulators for building executable simulation models.

2 Basic Design Patterns

A few elementary design patterns that are required for design, implementation, and test of a pruning algorithm for an SES are presented in the following subsections. The proposed patterns for modeling of system structures are necessary, particularly if one aims to use the SES tool

for automated model generation in the context of the extended SES/MB approach. In analogy to the semantics of feature models ([2], [3]) and mathematical logical expressions, we try to classify the first patterns according to their purpose.

In the context of feature modeling, four kinds of features are used: (i) mandatory features (logical AND), (ii) alternative features (logical XOR), (iii) optional features and, (iv) OR-features (logical OR). At first sight, it seems to be obvious and simple to decide which SES constructs can be used to define tree sections satisfying the feature categories above, but there is often more than one way to express the logical relations among tree sections, components or entities. However, the simplest patterns consist of just one descriptive node with its parent and children.

Besides patterns fitting into the classification according to feature modeling, we identified some patterns useful to illustrate and test SES tools dealing with issues like inheritance, the special attributes, such as couplings and priorities, and evaluation order.

2.1 Mandatory Tree Sections

Mandatory are those sections which need to be existent in each system belonging to a family of systems. For an SES describing a family of systems, this means that all possible PES representing a certain system variant will also include those parts. The corresponding logical expression is the AND and we can call the linkage a ‘*has-a*’ relationship. Design patterns of this type are the aspect node itself, a multi-aspect node or specialization siblings.

Design Pattern #1 - Aspect Node. Figure 2 depicts the simplest case of a design pattern for mandatory sections, the aspect node itself.

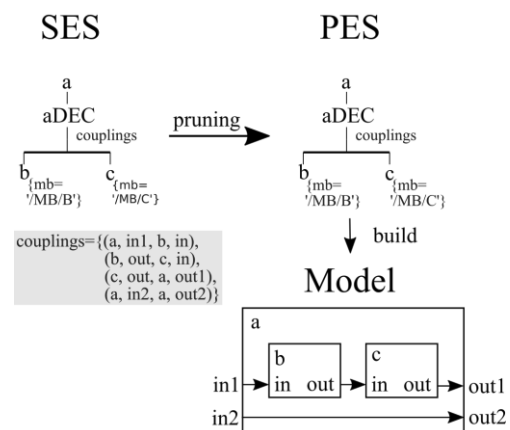


Figure 2: First design pattern for mandatory sections.

Each coupled system a consists of an entity b and an entity c . For model generation, aspect nodes need to define the special attribute for couplings, while the leaf nodes have the mb-attribute attached. Note that the derived PES is identical to the SES. The resulting abstract, simulator independent model on the right side is given to illustrate what kind of model structure would result from the PES. The following examples will not include detailed coupling definitions and models since couplings can be formulated analogously and concrete models depend on the chosen simulation environment.

Design Pattern #2 – Multi-Aspect Node. In Figure 3, a similar case to design pattern #1 is given. With this pattern we introduce the usage of SES variables, SES functions and the possibility to define multi-sets for setting values of attributes. Since the system as consists of a certain number of children of type b , multi-aspect nodes need to define two special attributes, the attributes numRep and couplings. When pruning a multi-aspect node, the numRep attribute is evaluated and an aspect node with children of the same type is created. In this example, the numRep attribute is defined by the current value of the SESvar VAR which is restricted to two or three by the semantic condition, but the value could also be hardcoded. Based on the number of children the couplings may vary, too. For an effective coding and to keep a lean SES tree, couplings are set via an SESfcn here. Based on the value of the SESvar VAR either cpl1 or cpl2 is chosen. The structure of cpl1 and cpl2 needs to be defined as it is presented for static couplings in Figure 2.

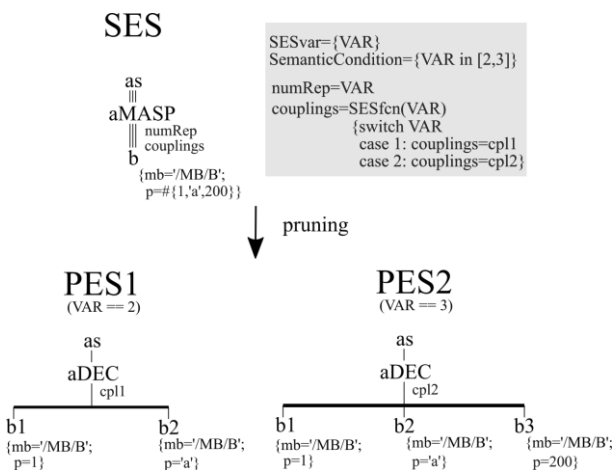


Figure 3: Design pattern for mandatory sections with a multi-aspect node.

The attribute p is set by dint of a multi-set variable. Although after pruning all children of as are of the same type b and are referencing the same basic model B in the MB, their parametrization can be different. Children's names are generated by appending a number to the entity name b to ensure that the resulting siblings fulfill the axiom of valid brothers.

Design Pattern #3 - Specialization Siblings. If two or more specialization nodes are on the same hierarchy level, they will all be evaluated. For the pattern depicted in Figure 4, this means that a will specialize into one of b or c AND into one of d or e . Which child of the specialization is taken depends on the values of the SES variables and the specialization rules. If, for example, VAR1 is set to 1 and VAR2 is set to 2, at specialization $a1SPEC$ the left child b is selected and, at specialization $a2SPEC$, the right child e is selected. During pruning, it depends on the pruning algorithm which of the two specializations is evaluated first. For this example, we assume that the left specialization node $a1SPEC$ is evaluated first. The evaluation order influences what is the resulting value of the mb-attribute since, according to the inheritance axiom, attributes with the same name are overwritten in the parent node.

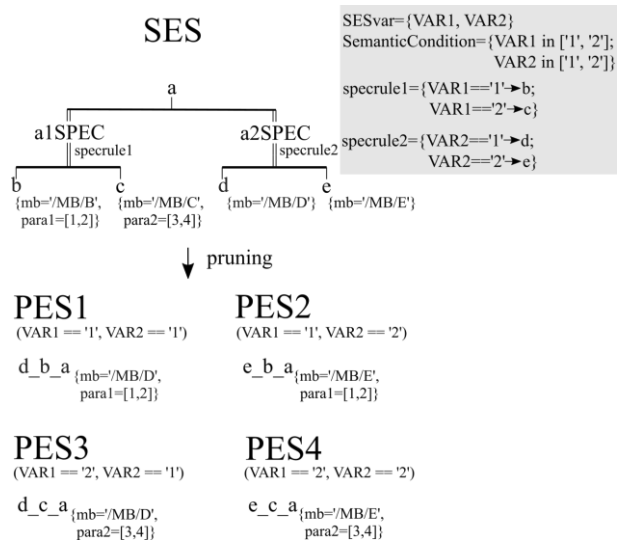


Figure 4: Design pattern for specialization siblings.

With this example, we also demonstrate the use of SES variables for defining specialization rules.

2.2 Alternative Tree Sections

Alternative are those sections from which exactly one part needs to be existent in each system belonging to a family of systems. A system described by a PES is of the type which is selected in a specialization or consists of the children of aspects chosen at aspect siblings. The corresponding logical expression is the XOR and we can call the linkage an 'is-a' relationship. Design patterns of this type are the specialization node itself, aspect siblings, multi-aspect siblings or siblings containing an aspect node and a multi-aspect node.

Design Pattern #4 - Specialization Node. The simplest pattern to specify alternatives is the specialization node shown in Figure 5. Based on the specialization rule, exactly one child needs to be selected to construct a valid variant. The specialization rule evaluates the SESvar VAR to allow a selection. After pruning, the resulting system can either be of type *b* OR of type *c*. Valid names are constructed by preceding the fathers name *a* with either *b* or *c* and attributes are inherited as defined by the SES axioms.

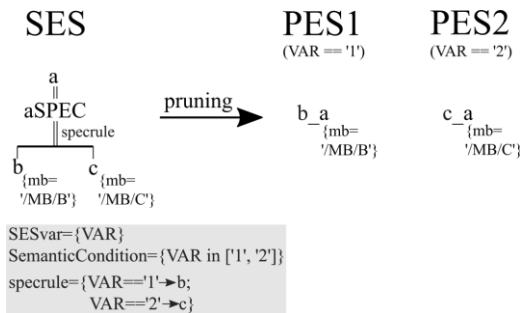


Figure 5: A simple specialization node.

Design Pattern #5 - Aspect Siblings. If two or more aspect nodes are on the same hierarchy level, exactly one of them has to be selected by evaluating the aspect rules of the aspect brothers. This is presented in Figure 6. The system *a* consists either of *b* and *c* or of *d* and *e*. The aspect rules one and two define which branch is selected during pruning.

Design Pattern #6 – Multi-Aspect Siblings. In the pattern shown in Figure 7, two multi-aspect nodes are on the same layer. In the first pruning step, the multi-aspect nodes are resolved leading to two aspect nodes. After this step, the resulting intermediate PES can be finally resolved using pattern #5 for aspect siblings, as previously described. The system *a* consists either of *b1*, *b2*, and *b3*

or of *c1* and *c2* depending on which child is selected based on the aspect rules.

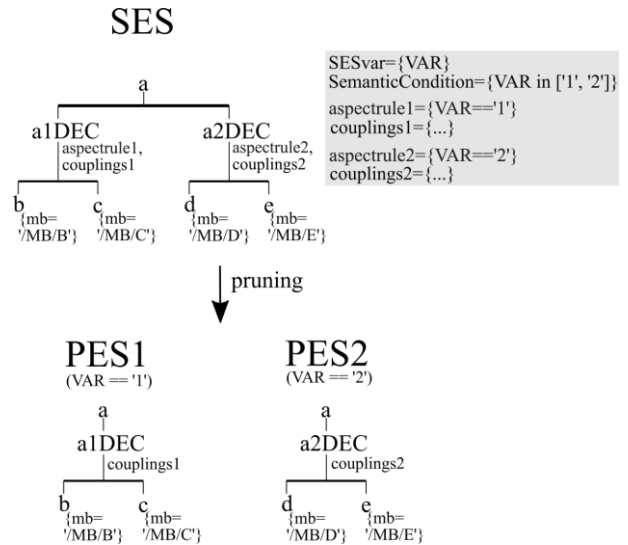


Figure 6: Aspect siblings result in an alternative selection.

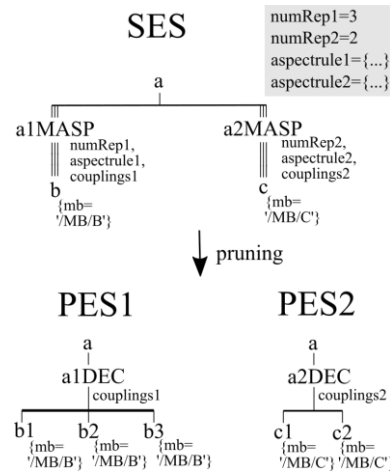


Figure 7: Multi-aspect siblings behave like aspect siblings.

Design Pattern #7 - Aspect and Multi-Aspect Siblings. If there are more than one aspect nodes and multi-aspect nodes on the same hierarchy level, the behavior is like aspect siblings after the multi-aspect node is resolved (see Figures 6 and 7).

2.3 Optional Tree Sections - Design Pattern #8

Optional sections can be contained in the resulting system structure, but they do not have to be. This can be done using an extension of the SES, the NONE element. A NONE element for a leaf entity node means that, if the NONE branch is selected, the entity is not included

at all. In the pattern shown in Figure 8, the specialization has a child which is a NONE element. Hence, this SES can evaluate to NONE during pruning based on the specialization rule. The system *a* can either be of type *b* or not exist at all.

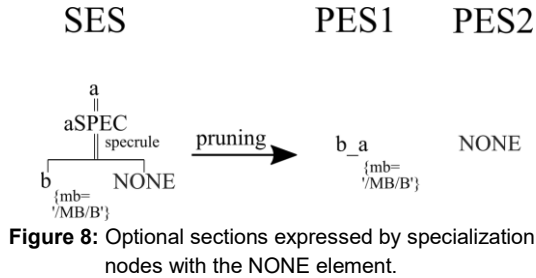


Figure 8: Optional sections expressed by specialization nodes with the NONE element.

2.4 OR Tree Sections - Design Pattern #9

Logical OR means that one or more entities or tree sections need to be included to get a valid variant. This can be expressed by an aspect node whose children are followed by specializations. Each specialization contains a NONE element as one child. The corresponding design pattern is shown in Figure 9. At least one specialization has to evaluate to a node not being NONE. By defining the specialization rules reasonably, the user has to ensure, that this is guaranteed. This pattern is composed by pattern #1 describing mandatory tree sections in combination with pattern #8 for optional elements. After pruning, the system *a* consists of *b* and *c*. System *b*, in turn, is of type *bs* or not existent while *c* is of type *cs* or not existent.

Couplings have to be adjusted when the tree changes by evaluating nodes. Since the couplings are defined at the aspect node *aDEC*, it is obvious, that there is a need for the possibility to define variable couplings. Variable couplings can be defined via SES Functions as introduced with design pattern #2.

3 Combined Design Patterns

In the previous section, the elementary design patterns were discussed in particular. During tool development, the necessity of testing combinations of basic patterns (combined patterns) was recognized. Issues addressed with these patterns are inheritance, evaluation order and priorities. In the next sections, a few combined patterns are introduced.

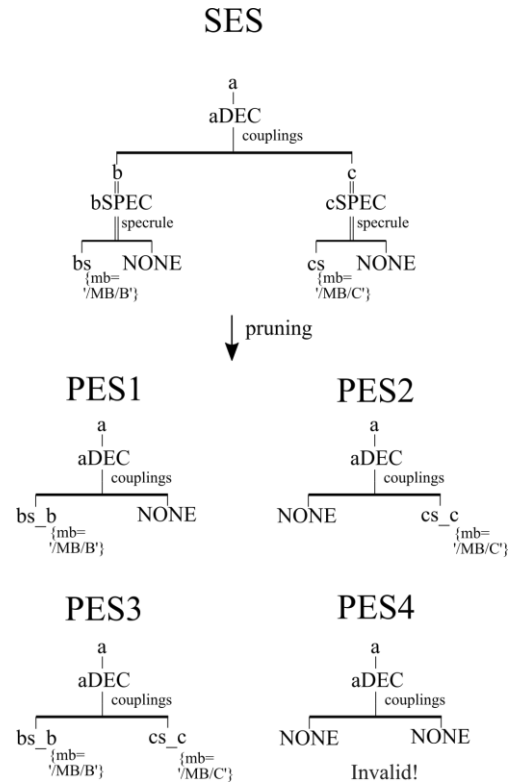
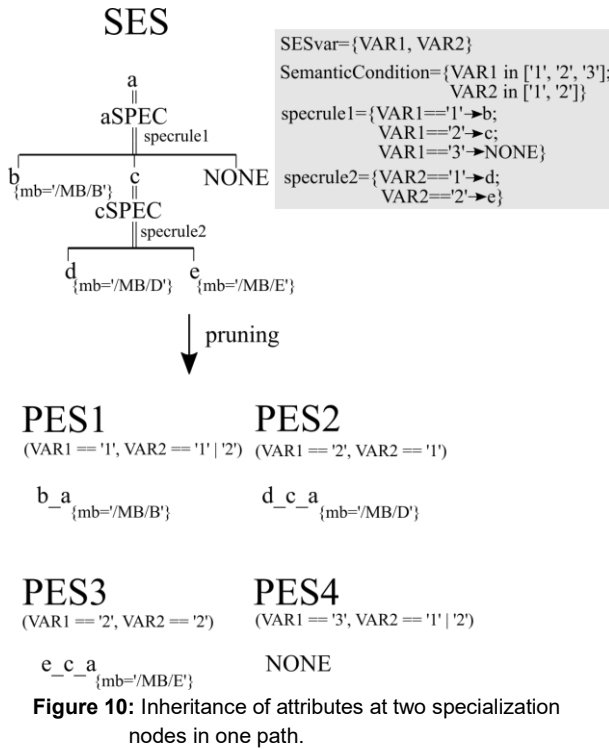


Figure 9: OR expressed by aspect nodes followed by specializations with NONE elements.

3.1 Design Pattern #10 - Two Specialization Nodes in One Path

Specialization nodes inherit the attributes of the selected child and append them to the father's attributes, as previously described. In Figure 10, there are two specialization nodes in one path. Additionally, the NONE element is used. Thus, this can be classified as an optional feature and shall clarify the axiom for attribute inheritance. During pruning, firstly *aSPEC* is evaluated. In case the child *c* is selected, another decision for child *d* or child *e* is taken. In that case, the attributes of the child node of *cSPEC* are inherited to the first node.



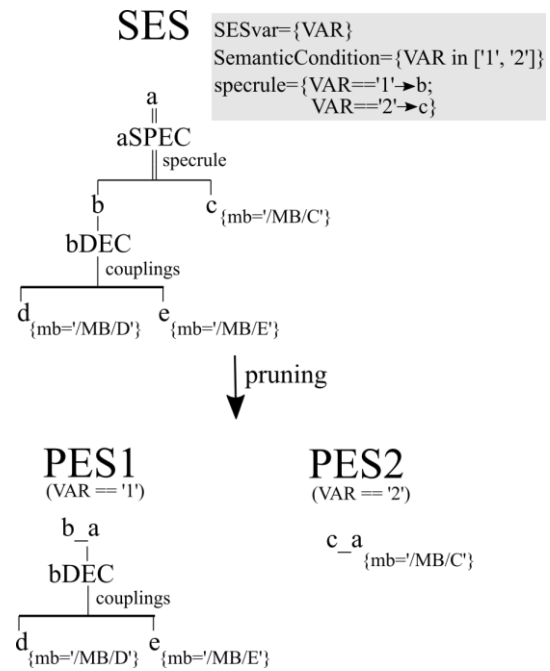
3.2 Design Pattern #11 – Specialization with Succeeding Aspect

Figure 11 depicts a specialization node with a succeeding aspect node. This pattern is a combination of a specialization node followed by a single aspect node. Therefore, it is a combination of alternative and mandatory tree sections. The system a can be of type b or c , the b can be decomposed in d and e .

3.3 Design Pattern #12 - Specialization and Aspect Siblings

When aspect nodes and specialization nodes are brothers, the specialization node has to be resolved first during pruning. If, additionally, an aspect node is below the specialization node, during pruning two aspect nodes will become siblings. Since, in this case, the occurrence of aspect siblings is not known until the first pruning step, aspect rules could not be formulated beforehand. In order to tackle this, the priority attribute for aspect and multi-aspect nodes was introduced. Throughout the whole SES, these nodes get a unique number. If, during pruning, aspect nodes become brothers, a decision about which to choose can be made. In Figure 12, an example is given. The system a consists of b and c , if it is of type d . If it is of type e , it can consist of b and c or of f and g . In case

the system is of type e , a decision as to which decomposition to take is made by interpreting the priority attribute.



3.4 Design Pattern #13 - Several Multi-Aspects in a Path

If a multi-aspect node is followed by a second or even more multi-aspects, complexity of resulting structures grows considerably. During pruning compliance with the SES axioms has to be ensured and therefore some renaming operations are necessary. Figure 13 depicts two successive multi-aspects where the numRep variable of $bMASP$ defines a multiset. Renaming of b to $b1$ and $b2$, as necessary for children of multi-aspects and explained in pattern #2, results in renaming of $bMASP$ to $b1MASP$ and $b2MASP$. Generally, one can say that renaming an entity node always calls for renaming the following descriptive node, too. The first pruning step resolves the multi-aspect $aMASP$ to an aspect $aDEC$ with two child nodes of type multi-aspect. In a second step the child nodes are resolved to aspect nodes, too. Since the numRep variable of $b1MASP$ and $b2MASP$ was set via the multi-set, variable couplings are necessary. Couplings are set with the help of an SESfcn which chooses cpl1 or cpl2 based on the current value of numRep.

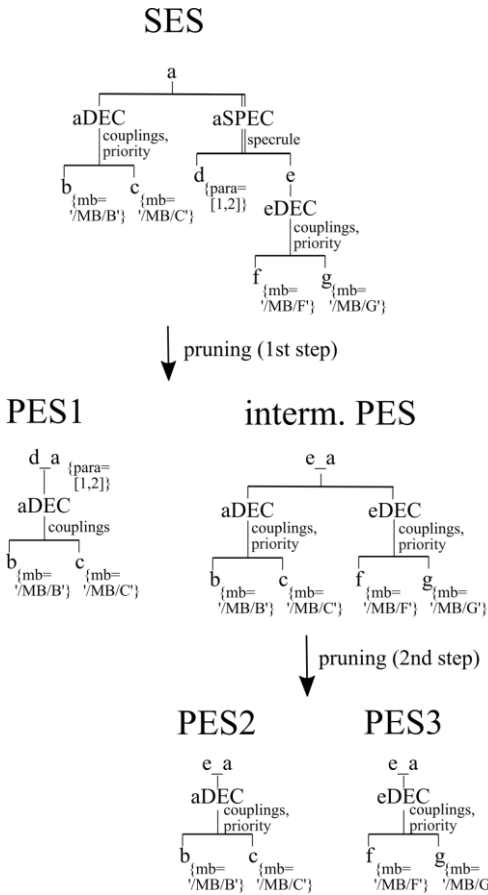


Figure 12: Aspects can become siblings by resolving a specialization.

3.5 Design Pattern #14 – Selection Constraints

A modification of pattern #3 (specialization siblings) is shown in Figure 14. With this pattern the use of selection constraints and semantic conditions to limit the possible valid structures is pointed out. The corresponding constructs in feature models are known as *require* and *exclude* [2].

The dotted line from leaf node *c* to node *d* in Figure 14 depicts a selection constraint which means that, if *c* is selected at *aISPEC*, *d* has to be chosen at *a2SPEC*. The resulting PES is PES3. Note, that current value of VAR2 does not matter for the selection, if VAR1 is set to 2.

Another way to control possible variants are the semantic conditions. The semantic conditions for value combinations of VAR1 and VAR2 interdict the selection of *d*, if *b* is already selected.

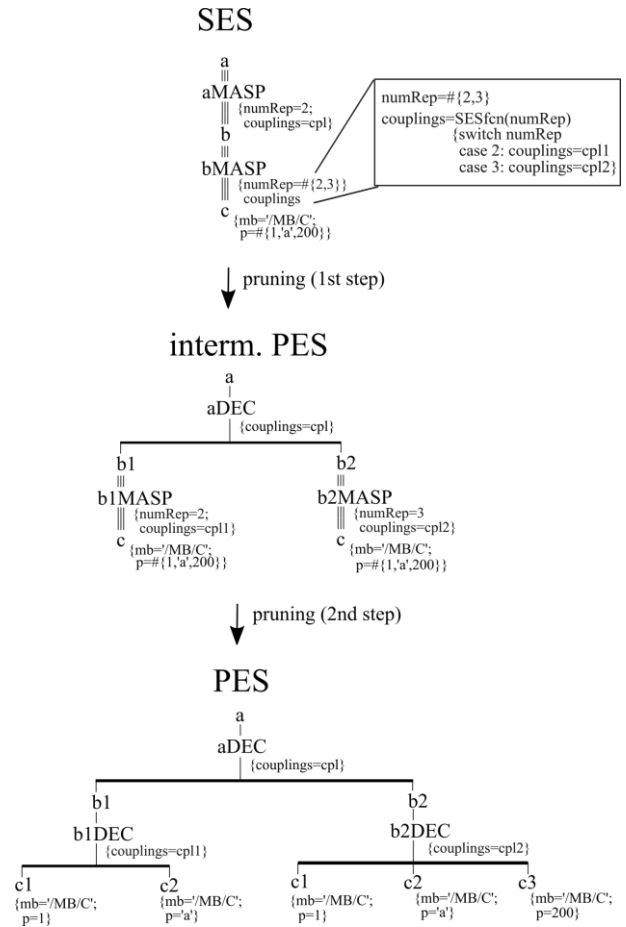


Figure 13: Successive multi-aspects with variable number of replications.

4 Conclusion and Future Work

This paper gives an overview of essential patterns needed to describe and test the pruning process of an SES. With the described patterns, the implementations and tests of the software tools for the extended SES/MB infrastructure developed in the Research Group CEA could be done. However, while developing and testing the software tools, more combined patterns were found and tested. Some pitfalls in the pruning algorithm were discovered. Therefore, additional attributes, such as the priority attribute, were introduced. However, e.g. for specialization siblings, there has to be found a way to decide which sibling is evaluated first. Formalized definitions of pruning algorithms are essential part of future work.

Future works regarding the SES software tools will especially cover enhancements of the model builder. For a full software support of the SES/MB-based infrastructure, it is necessary to have a fully functional model

builder, which supports the generation of hybrid models with basic systems that are modeled using different paradigms. In addition, the Python implementation will allow to support a set of model builders for different simulators.

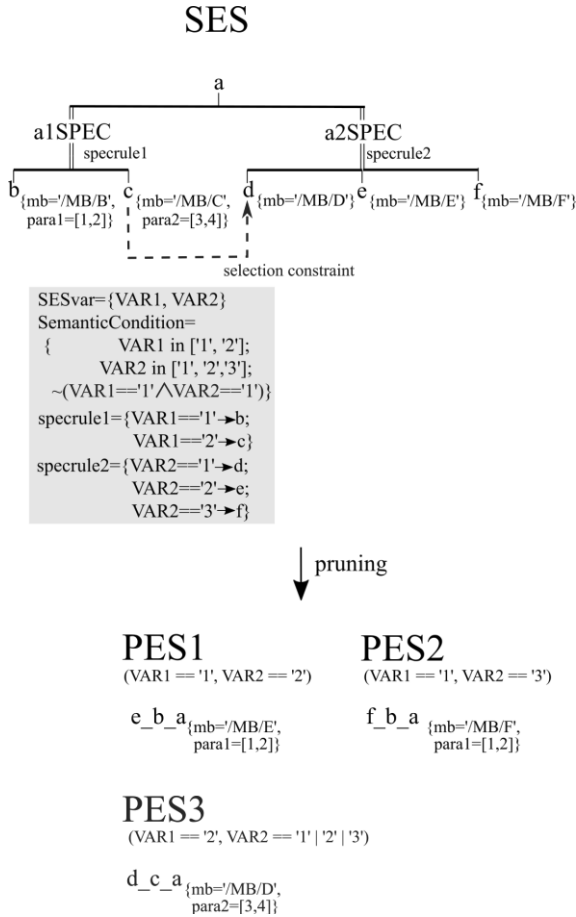


Figure 14: Variant restriction with selection constraints and semantic conditions.

References

- [1] Deatcu, C., Folkerts, H., Pawletta, T., and Durak, U.. *Design Patterns for Variability Modeling Using SES Ontology*. SpringSim-Mod4Sim 2018, April 15-18 2018, Baltimore, MD, USA, Society for Modeling & Simulation International (SCS), pp. 528-539. (DOI: 10.22360/SpringSim.2018.Mod4Sim.004)
- [2] Capilla, R., J. Bosch, and K. C. Kang (Editors). *Systems and Software Variability Management*. Springer, Berlin Heidelberg, Germany, 2013.
- [3] Kang, K. C., S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technical Report, Software Engineering Institute Carnegie Mellon University, Pittsburgh, PA, USA, 1990.
- [4] Capilla, R., and J. Bosch. *Binding Time and Evolution*. In *Systems and Software Variability Management*, edited by R. Capilla, J. Bosch, and K. C. Kang, pp. 57-73. Springer, Berlin Heidelberg, Germany, 2013.
- [5] Zeigler, B. P.. *Multifaceted Modelling and Discrete Event Simulation*. Cambridge, Academic Press, 1984.
- [6] Zeigler, B. P., T. G. Kim, and H. Praehofer. *Theory of Modeling and Simulation*. 2nd ed. San Diego, CA, USA, Academic Press, 2000.
- [7] Rozenblit, J. W., and B. P. Zeigler. *Representing and Constructing System Specifications Using the System Entity Structure Concepts*. Proceedings of the 1993 Winter Simulation Conference, pp. 604-611.
- [8] Zeigler, B. P., and P. E. Hammonds. *Modeling and Simulation-Based Data Engineering*. San Diego, CA, USA, Academic Press, 2007.
- [9] Zeigler, B. P., and H.S. Sarjoughian. *Guide to Modeling and Simulation of Systems of Systems*. Springer, London, UK, 2013.
- [10] Röhl, M., A. M. Uhrmacher. *Composing simulations from XML-specified model components*. Proceedings of the 2006 Winter Simulation Conference, pp. 1083-1090.
- [11] Wang, S., G. A. Wainer. *Semantic selection for model composition using SAMSaaS*. Proceedings of the 2015 Spring Simulation Conference - TMS-DEVS, pp. 25-32.
- [12] Barros, F. J.. *The Dynamic Structure Discrete Event System Specification Formalism*. SIMULATION: Transactions of The Society for Modeling and Simulation International. 13(1996)1, pp. 35-36.
- [13] Zeigler, B. P., and H. Praehofer. *System Theory Challenges in the Simulation of Variable Structure and Intelligent Systems*. Computer Aided Systems Theory — EUROCAST '89. EUROCAST 1989. Lecture Notes in Computer Science, edited by F. Pichler and R. Moreno-Diaz., vol 410, pp. 41-51. Springer, Berlin, Heidelberg, 1990.
- [14] Pawletta, T., A. Schmidt, B. P. Zeigler, and U. Durak. *Extended Variability Modeling Using System Entity Structure Ontology within MATLAB/Simulink*. Proceedings of the 2016 Spring Simulation Conference – ANSS, pp. 62–69.
- [15] RG CEA. *The SES Toolbox for MATLAB / Simulink Website*. http://www.cea-wismar.de/tbx/SES_Tbx/ses-ToolboxMain.html. Accessed Nov. 22, 2017.
- [16] Schmidt, A., U. Durak, and T. Pawletta. *Model-Based Testing Methodology Using System Entity Structures for MATLAB/Simulink Models*. SIMULATION: Transactions of The Society for Modeling and Simulation International. 92(8), pp. 729-746, 2016.
- [17] Folkerts, H., T. Pawletta, C. Deatcu, and U. Durak. *Variability Modeling for Engineering Applications*. Simulation Notes Europe vol. 27(4),2017, pp. 167–176.
- [18] RG CEA. *The PDEVs Toolbox for MATLAB Website*. http://www.cea-wismar.de/tbx/DEVS_Tbx/Matlab-DEVS_Tbx.html. Accessed Nov. 29, 2017.

***SimLack*: Simulation-based Optimization and Scheduling of Generic Powder Coating Lines**

Lukas Hollenstein^{1*}, Adrian Lötscher¹, Fabian Luccarini²

¹ZHAW Zurich University of Applied Sciences, IAS Institute of Applied Simulation, Grüental, Postfach, 8820 Wädenswil, Switzerland; *lukas.hollenstein@zhaw.ch

²e. Luterbach AG, Mülacher 10, 6024 Hildisrieden, Switzerland

Abstract. Powder coating and paint-spray lines are often complex production plants because of many dynamical dependencies, limited buffer space and sequence dependent changeover times. We have developed a generic simulation and optimization platform that enables the engineers to design more performant and energy efficient facilities and the production planners to increase productivity through simulation-based optimization. The simulation environment builds on a generic modelling library that captures all variations of such facilities. Executable models are generated automatically from annotated CAD layouts. As a result, the system smoothly integrates with the engineering process. Once the facility is in use, the fully specified virtual plant is used for simulation-based scheduling, employing a combination of a generic priority-based heuristic and a variant of simulated annealing. We discuss how these two aspects of the system render it an important innovation for the painting line industry and show first results from the scheduling system.

Introduction

Powder coating and paint-spray facilities are among the most important energy consumers in the industry, mainly due to their cure ovens staying in operation all day long. To keep operating costs and environmental impact down, machinery manufacturers, like the company e. Luterbach AG, are aiming to optimize the facility layouts and dimensions to their customers' needs. Once a powder coating line is installed and productive, the operating company intends to plan their production schedule in order to maximize productivity and minimize operation cost. Such facilities normally rely on a closed loop material handling system that transports the parts to be treated on hangers along chains. Failing to

feed the system with an efficient job sequence can lead to a significant drop in throughput or even to deadlocks.

Since a few years, Luterbach is already at the forefront of designing energy efficient facilities through simulating the energy and heat budget by means of pinch analysis [1]. The next step is to quantitatively optimize the design and the usage strategies of such systems with respect to the material flow and production performance. Even though manufacturing resource planning (MRP II) for generic production facilities may exist on the market the specific industry sector (powder coating and paint-spraying) has not embraced such solutions. Plant operators are often not employing enterprise resource planning (ERP) software throughout the production process or, if they do, struggle to enable it with sophisticated scheduling abilities that go beyond simple heuristics based on the planner's experience. In addition, when designing new facilities and retrofits, it appears that most machinery manufacturers still rely on static analysis and experience to plan the facility layouts. These approaches are bound to fail as facilities grow larger, more flexible and complex.

The scheduling problem for generic powder coating and paint-spray facilities can in principle be seen as hybrid flow shop scheduling with unrelated parallel machines [2, 3, 4]. Unfortunately, the rules that govern the dynamics of the facilities can be highly specific and complex, as can be the routes of parts through the system and the timing involved. As a result, the scheduling problem for such manufacturing systems is not straightforward to formulate in closed form in order to apply mathematical programming techniques or well-known heuristics. Therefore, a simulation-based approach is more promising [5, 6].

Apart from scheduling, there are two more reasons for a simulation approach in the context of plant engineering: first, the facility design phase highly profits from a generic simulation framework than can be used

to rapidly build models and compare different scenarios and configurations. Secondly, a dynamic visualization of the systems helps to understand it, to pinpoint bottlenecks and, importantly so, supports sales with a quantitative proof of concept. Wilson & Zettle report on an operative scheduling solution for powder coating lines that relies on existing simulation models developed in the planning phase of the facility [7]. However, it is unclear in which way the simulation is used in their heuristic scheduling algorithms and if the scheduling methodology is generically applicable to any facility configuration.

We extend this approach in two ways. Firstly, we specify a generic annotation scheme for computer aided design (CAD) layouts that enables automatic model generation for powder coating lines. Secondly, we discuss a scheduling methodology that combines a generic priority-based sorting heuristic with a variant of the simulated annealing meta-heuristic that improves the solution by means of many simulation runs. We have implemented these concepts as an integrated platform for simulation-based optimization and scheduling of powder coating and paint-spray facilities. The simulation environment is called *SimLack*, derived from *Simulation* and *Lackieranlagen*, the German word for paint shop.

1 Automatic model generation from CAD layouts

When planning new powder coating and paint-spray facilities or designing a retrofit, machinery manufacturers aim at optimizing the facility layout and number and dimensions of building blocks in order to minimize the energy consumption and maximize the total throughput. Modelling and simulations enables prototyping, iterative refinement and quantitative analysis of designs. However, firstly, building and analyzing simulations is time consuming and, secondly, the engineers are usually no simulation experts.

To overcome this issue, we have developed an integrated simulation environment that consists of a generic modelling library for painting lines with an associated domain specific language based on annotated CAD drawings to generate simulation models automatically. A discrete-event simulation engine runs the models while showing the model state as a schematic visualization auto-derived from the original CAD drawing.

1.1 The generic model

Painting line facilities all follow similar principles and consist of the same building blocks, but vary in their details. A hanging conveyor system of some kind (often a *Power and Free* chain) is carrying the parts to be treated in hangers through a series of processing steps including pre-treatment, powder application or paint-spraying, curing, and cooling. Other intermediate steps as well as buffering and inner closed loops can be present, which complicate the system. Due to space and cost restrictions, the lines typically cannot be kept lean such that material- and color-dependent changeovers have to be taken into account and dynamic dependencies appear at certain points in the facility.

The generic discrete-event model describes each job as a number of parts on a hanger. The hangers are the entities that are moved through the system. The system is defined by the *facility configuration* that consists of

- the *positions* at which the hangers can reside,
- the *segments* of the hanging conveyor network connecting the positions,
- the *decision points* where two segments join or separate, and
- the *processes* in which the hangers can change state, representing either a loading point (source), a generic process (pretreatment, powder application, curing, cooling, etc.), or a discharging point (sink).

Each position belongs to exactly one process. Each segment consists of a sequence of positions. The segments must form a closed loop globally.

Next to the facility configuration, the model is specified by the following additional data:

- the sequence of processes a job has to pass through (incl. processing time or minimal/maximal retention time),
- specialized prioritization rules per decision point (optional), and
- sequence dependent changeover times per process (optional),
- availability patterns of processes to represent shifts and planned outages (optional).

The entities (hangers) are moved from one position to the next along the segments and cannot overtake other

entities. Positions can have capacity larger than one to allow for quickly adding buffer space in experiments. The routing of entities through the system is derived from the job's sequence of processes and the prioritization rules at decision points that can depend on the job attributes and the state of processes.

The model is generic in the sense that it is agnostic about the actual meaning of the processes and even the attributes of the jobs that influence the processing time, the changeover times and the prioritization rules in decision points. The semantic behind the processes and the job attributes is only in the data that is provided to generate the simulation model. This means the model represents very generically any kind of closed-loop flow shop system. It is therefore widely applicable, not only to the partnering company and not even only to painting lines.

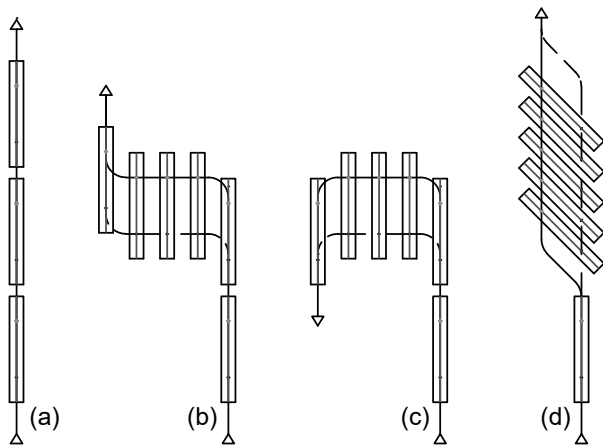


Figure 1: Example snippets from a CAD layout. Segment (a) has three positions in longitudinal direction; (b) is a transversal buffer where hangers exit in the same direction as they enter, i.e. no change of orientation; (c) is a transversal buffer with change of direction; and (d) is a diagonal buffer without change of direction.

1.2 The CAD-based domain specific language

In order to generate executable instances of the generic simulation model the complete facility configuration incl. its data needs to be provided. Machinery engineers use CAD drawings when designing new plant layouts or investigating changes to existing facilities. To integrate the simulation library seamlessly into the workflow of the engineers we devised an annotation

scheme for CAD layouts. The annotated CAD layout is imported into the simulation environment and turned into a specific simulation model automatically. The data needed to fully specify the simulation experiment (jobs, prioritization rules, changeover times, and availabilities) is not provided through the CAD-interface and needs to be imported separately. Here we focus on the CAD-based domain specific language, as it is the main innovation that enables automatic model generation.

CAD layouts of painting lines are complex and contain many elements, most of which do not provide information for generating the model. Fig. 1 shows snippets from a CAD layout exemplifying sequences of positions (i.e. segments). A drawing does not naturally contain semantic information, e.g. even though positions are always rectangular shapes it is not clear from the outset which process the positions belong to. Therefore, we require the elements to be placed in blocks (aka layers) that are named semantically and structured hierarchically according to the following scheme:

- **Block Processes:** contains all processes, e.g.
 - **Block LoadingPoint:** contains all its positions, e.g.
 - * Block P001 (with rectangle)
 - * Block P002 (...)
 - **Block Buffer1:** e.g.
 - * Block P003 (...)
 - * ...
 - * Block P010 (...)
 - ...
- **Block Segments:** contains all segments, e.g.
 - **Block P000–P002_0** (with lines and arcs)
 - **Block P002–P010_1** (...)
 - ...

There are optional naming conventions for the processes such that the type of process can be recognized (loading point, generic process, or discharging point). The positions can be named arbitrarily but uniquely. In the example above we applied the optional convention to name the positions P001 ... P010.

In order for the importer to uniquely identify the starting and ending positions of segments, we require the following convention: the names of segments must be made up of the *last position of the preceding segment* and the *last position of the current segment*. For

example the segment P000-P002_0 starts at the position P000, that is the end of the preceding segment, and ends at position P002. The importer algorithm identifies the intermediate positions along the segment geometrically.

Segments feature another complication, namely the possibility that the orientation of the hangers can change in the end of a transverse buffer, see Fig. 1. Because such situations are not straightforward to detect algorithmically we require the name of the segment to end either with _0 for “no change of orientation” or with _1 for “change of orientation.”

The combination of CAD drawing and annotation scheme (naming convention) can be seen as a domain specific language that enables data-driven model generation. Because the annotation with specifically named blocks is lightweight in its application, this approach integrates neatly into the workflow of the plant engineers. Other possibilities such as specifying the facility configuration in tables or using a graphical drag-and-drop editor would certainly lead to significant overhead in the design cycles. The specification is domain specific in the sense that it does rely on consistency relations that hold specifically in the powder coating domain, even though it may be applicable more generally to other closed-loop hybrid flow shop systems.

2 Simulation-based optimization

Production planners of painting lines currently use experience-based heuristics and simple static estimates to put together production schedules given a number of orders. Algorithmic assistance is highly asked for; however, acceptance of decision support systems strongly depends on the level of control and understanding the user can have.

With this requirement in mind, we have developed a hybrid scheduling scheme that builds on the user’s planning experience and knowledge of the facility and product characteristics. The method works in two phases that we describe briefly in the following.

2.1 Priority-based heuristic

- The user selects a number of *sorting priorities*. These are job attributes like material, paint color, due date, etc. and can be configured freely for each specific facility.

- All permutations of the sorting priorities are generated.
- Each permutation is then used to generate a job sequence by sorting hierarchically according to that sequence of priorities.
- For n selected attributes this procedure yields $n!$ priority-sorted job sequences.
- All sequences are evaluated using the simulation model of the facility, respecting all constraints like shift plans, changeover times, outages etc.

The result is twofold: A good estimate of the optimal schedule (according to some objective function like the makespan) and insight into which job attributes have a strong impact on the performance of the job sequence. This insight helps to improve acceptance by the user and the user can explore and learn about the facility.

2.2 Simulated annealing for refinement

Starting from the best job sequence found by the priority-based heuristic, we employ a variant of *simulated annealing*. This class of stochastic meta-heuristics is well-known to be able to find globally near-optimal solutions with certainty and has proven convergence properties [6]. The basic algorithm works as follows:

- In each iteration a random solution in the neighborhood of the current solution is generated.
- The objective function f is evaluated and if the new solution is better than the current one it is accepted directly as the next step.
- If the new solution is not better, it is accepted all the same with a probability proportional to $e^{-\Delta f/\beta_i}$. Here Δf is the positive difference between the new and the current objective value. The parameter β_i is a positive number that depends on the iteration i . It starts relatively large which means the probability of accepting poor solutions is large and therefore exploration of the solution space is made possible. With increasing number of iterations β_i is decreasing such that poor solutions are only seldom accepted, exploration ceases and exploitation (local refinement) is emphasized.
- As soon as the globally best solution does not improve anymore for some number of iterations or the maximal number of iterations is reached the algorithm stops.

The difficulty in applying simulated annealing is to define an efficient solution neighborhood, i.e. a procedure for generating new solutions nearby a given solution. A simple choice would be to swap a random single pair of adjacent jobs in the sequence, *single adjacent interchange*. We compared the performance of a number of different neighborhoods and found that the scheduling problem at hand asks for more exploration than single interchanges. Therefore, we make the neighborhood depend on the iteration: the number of adjacent interchanges starts at the length of the job sequence, i.e. large exploration, and subsequently reduces to one with increasing number of iteration. A similar effect could be achieved with single adjacent interchange and β_i reducing more slowly, but we found this variant to converge more quickly for a set of reference problems.

2.3 Preliminary results

Fig. 2 compares random sampling of the solution space with the proposed probability-based heuristic and simulated annealing. We show the distribution of objective values from the evaluated job sequences. In this case, the objective function is the makespan that is minimized. The distribution is multi-modal (three clusters appear) which is due to the discontinuity of the objective function. Observe that random sampling is not able to find the best-performing solutions with makespan below 160h. The priority-based heuristic generates a broad range of solutions incl. some that come close to the best ones found. The simulated annealing algorithm, however, is able to improve on the sorting heuristic and spends most of its time near the optimum trying to improve on it locally. These are preliminary explorations and more systematic performance analyses and comparisons are left for future work.

3 Summary and outlook

The integrated simulation and optimization environment *SimLack* essentially provides two important innovations to the industry sector of powder coating and paint-spray lines. Firstly, a generic modelling library and a domain specific language were developed to provide the possibility to generate simulation models automatically from CAD layouts. This approach integrates the simulation environment seamlessly into the workflow of the plant engineers. Secondly, a simulation-based scheduling methodology was devised, based on

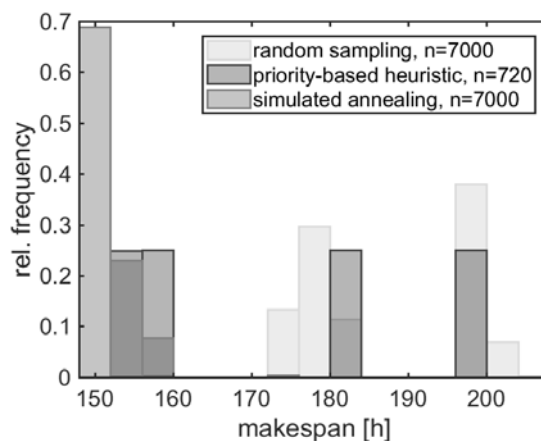


Figure 2: Comparison of different algorithms in terms of the distribution of the objective values (makespan) from evaluated job sequences in the course of the optimization runs.

a combination of a generic sorting heuristic and a variant of the simulated annealing meta-heuristic.

After importing a CAD layout into *SimLack*, a graphical user interface eases tweaking the facility configuration, data import and editing, the specification of detailed prioritization rules to apply at decision points, setting up changeover criteria and durations, and shift plans. The integrated platform enables the plant engineers to set up, carry out and organize simulation experiments to compare different facility layouts and various parameter settings.

A first use-case shows that e. Luterbach AG was able to acquire a new contract successfully through the ability to fine-tune the plant details and to convince the customer with quantitative estimates of its performance. In this case, the seamless integration of the *SimLack* platform proved crucial for its success, as the engineers went through several iterations to refine the layout to the point where requested performance could be achieved.

Preliminary performance tests of the scheduling system with small and medium sized problems show:

- Even though the sorting heuristic in the first stage can provide relatively good solutions the second stage can typically improve on it in a few thousand iterations (simulation runs), depending on problem size.
- Our simulated annealing variant with a neighborhood with decreasing number of adjacent inter-

changes is converging more efficiently than variants with a constant number of adjacent interchanges.

- Since simulations are run in parallel, a scheduling run for a large sequence of 790 jobs on a normal sized facility takes about 30 min (for 7000 simulations) on a quad-core laptop to run.

These first results are encouraging to refine details of the *SimLack* system and its algorithms. Next to more in depth performance analysis, we are planning to add alternative algorithms that the user can choose from and compare against each other. Further, a challenging improvement is to add support for the plant design process by providing means of automatically optimize certain parameters and aspects of a facility configuration.

Acknowledgement

We thank Florian Eggenberger and Ramon Kramis from e. Luterbach AG and Ivo Berchtold and Othmar Winiger from wirth+co AG for their constructive collaboration and the two companies for their support. This work is supported by the Commission for Technology and Innovation (CTI) of the Swiss Confederation through grant number 17655.1 PFEN-ES.

References

- [1] ke. A quantum leap. *IST International Surface Technology*. 2017;10(1):16–21.
- [2] Pinedo ML. *Planning and Scheduling in Manufacturing and Services*. New York: Springer-Verlag, 2nd ed. 2009.
- [3] Gupta JND, Stafford EF. Flowshop scheduling research after five decades. *European Journal of Operational Research*. 2006;169(3):699–711.
- [4] Ribas I, Leisten R, Framiñan JM. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*. 2010; 37(8):1439–1454.
- [5] Kiran AS. Simulation and Scheduling. In: *Handbook of Simulation*, edited by Banks J, This chapter contains sections titled: Introduction, Definitions and Background, Scheduling Approaches, Simulation of Job Shops, Simulation-Based Scheduling: Application, Implementation Issues, Concluding Remarks, References, pp. 677–717. Wiley-Blackwell. 2007;.
- [6] Gosavi A. *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Operations Research/Computer Science Interfaces Series. Springer US, 2nd ed. 2015.
- [7] Wilson H, Zettl R. Simulation and Optimization Based Flexible Job Scheduling of Powder Coating Lines. In: *Digital Product and Process Development Systems*, edited by Kovács GL, Kochan D, IFIP Advances in Information and Communication Technology. Springer, Berlin, Heidelberg. 2013; pp. 369–375.

oHMint: An Online Mathematics Course and Learning Platform for MINT Students

Helena Barbas¹, Franz Konieczny², Alexander Lohse¹, Thomas Schramm^{3*}

¹Universität Hamburg, Fachbereich Mathematik, Bundesstraße 55, 20146 Hamburg, Germany.

²integral-learning GmbH, Clausewitzstraße 2, 10629 Berlin, Germany.

³HafenCity Universität Hamburg, Überseeallee 16, 20457 Hamburg, Germany.

**thomas.schramm@hcu-hamburg.de*

Abstract. The oHMint project is an initiative with the goal of providing an online course and learning platform for STEM students of higher mathematics. Its flexible design allows it to be used for self-study as well as blended learning scenarios including flipped classrooms. In a pilot project in 2017/18 the chapter *Differential Calculus* is being produced as a prototype of an oHMint unit. This pilot is organized and funded through the Hamburg Open Online University, with technical support and implementation by integral-learning GmbH. It is intended as a catalyst for the future development of the full oHMint course spanning four semesters. A unique characteristic of oHMint is its broad supporting base among a variety of German institutions in the form of the OMB+ consortium.

Introduction

Engineering mathematics is the main tool to describe the world and to simulate its processes. A profound mathematical education is therefore essential for future engineers and scientists. Digitalization is becoming increasingly important in teaching and studying at university level. More and more online learning tools are being developed to address the current need for educational resources that go well beyond the classical textbook by providing high temporal and spatial flexibility for users while at the same time offering an interactive learning environment. In this spirit, the oHMint project was initiated through the OMB+ consortium, a group of 14 German institutions of higher education. The acronym stands for **online Higher Mathematics (for m)int**. It is the aim of oHMint to provide university students of STEM degree programmes with a modern way of transitioning smoothly from a high school level of mathematics to a bachelor degree level. The OMB+ consortium has already developed the online mathematics bridging course OMB+, which is widely used and recommended as a

preparation for those interested in a STEM university degree programme. This experience is a valuable asset for oHMint.

Currently, in a pilot project with a lifespan of 14 months one base unit of oHMint is being created. This serves as an opportunity to test new didactical approaches for mathematics online courses as well as innovative types of exercises addressing the specific needs of the current student generation.

1 Background of oHMint

1.1 OMB+ mathematics bridging course

The OMB+ consortium has developed and maintains the well-accepted “Online Mathematics Bridging Course Plus” OMB+ [1] under the auspices of the TU9, a group of nine important German technical universities. The course is based on the recommended cosh catalogue [2], which sets a standard of mathematical requirements for freshmen in STEM studies, and should also give an orientation for school teachers in mathematics.

The OMB+ is text oriented, but includes a large amount of questions, interactive elements, videos and examples with standard solutions, which can be uncovered systematically. As of today, around 50 German institutions use and recommend the OMB+, which is available in German and English and currently being translated into Chinese. Supplementary chapters covering e.g. stochastics, complex numbers and formal logic have been added and further educational videos are in the making. The oHMint project uses the same enriched text oriented style and shares similar features. For more information on OMB+ see [1] and [3].

1.2 The Hamburg Open Online University

The acronym HOOU stands for Hamburg Open Online University (www.hoou.de), a joint initiative of six institutions of higher education in Hamburg: Universität

Hamburg (UHH), Hamburg University of Applied Sciences (HAW), Hamburg University of Technology (TUHH), HafenCity University Hamburg (HCU), Hochschule für bildende Künste (HFBK) and the Hochschule für Musik und Theater (HFMT). The Hamburger Senat (Senate of Hamburg) founded it in 2015 as the educational branch of “Strategie Digitale Stadt” (Strategy Digital City), an initiative to bundle processes of digitalization within the city.

The HOOU focuses on learners and collaboration for problem solving and develops study material at academic level. Its target groups are not only the custom university students but also interested non-academic persons. Thus, all study material is open as Open Educational Resource and is distributed under a CC license. The production of the first chapter of oHMint is funded by the HOOU and running from November 2017 through December 2018.

2 Modularisation and scope

2.1 Course structure

The OMB+ consortium designed the preliminary structure of the curriculum for the full oHMint course. It spans four semesters and aims to cover all the mathematics typically taught in STEM studies at German institutions of higher education. The curriculum was carefully structured, keeping in mind that different (types of) universities have different needs for their respective degree programmes. The course will be available in English and German. With oHMint we aim to create a course that consists of units which can be combined in various ways to cover these needs. The fact that a broad spectrum of institutions is represented in the OMB+ consortium justifies our hope that the final course will be widely accepted within the relevant community. This clearly distinguishes oHMint from existing higher mathematics online courses in Germany.

There are three types of units in oHMint: base units cover standard content and are expected to be included in every higher mathematics course, regardless of the institution it is taught at. Supplementary units go deeper into the (theoretical) background of a base unit, and include more abstract views on base unit concepts as well as more involved parts. Finally, optional units exist for subjects that are not necessarily always part of a higher mathematics course, depending in particular on the study programme and/or institution. The idea is that lecturers will be able to choose and combine the units that are relevant for their classes to provide students with everything they need (but not more).

Parallel to the development of the content of oHMint, a data base is being created that keeps track of all the interconnections between mathematical concepts and results in the units and which will help to ensure the desired

modularity of the course.

The units themselves are text-based and consist of lectures, exercises, trainings and final tests. They are supplemented by videos where important concepts or ideas are also shown. Each chapter starts with a motivation for the new mathematical concepts taught such that students develop a sense for the importance of the content they are supposed to learn. Moreover, many interactive elements are being implemented in oHMint, some are discussed in more detail below.

2.2 Content of the first-semester course

The learning material from which a lecturer can choose for a first-semester oHMint course consists of the following units:

- Three base units: numbers and functions, differential calculus, integral calculus.
- Four supplementary units: differential calculus, integral calculus, sequences, series.
- Eight optional units: continuity, determination of zeros, L’Hospital’s rule, partial fraction decomposition, approximate integration, sequences, series, Fourier analysis.

The scope of the content of each unit has been roughly assessed in terms of the European Credit Transfer and Accumulation System (ECTS) which is commonly used at European institutions of higher education. ECTS points measure the volume of learning based on workload and desired learning outcome. In Germany, one ECTS point is equivalent to 25-30 hours of studying.

The oHMint working group has made an assignment between the content of the chapters of the first-semester course and ECTS points. The base units are assigned 1 ECTS point each, the supplementary units and optional units are assigned 0.5 ECTS points. The workload for the corresponding units is estimated to be within the given range.

These values are used as an orientation for the lecturers who use the material for their courses. The actual assignment of credit points to the course is made by the lecturer or their institution.

3 Didactical concepts of oHMint

3.1 Didactical methods

The oHMint materials are designed to be used for a wide range of blended learning environments up to fully web-based self-learning courses. We selected our didactical methods to specifically suit these environments.

Our guiding principles are a spiral curriculum (after J. Bruner, see [4]), the development of intuitive understanding, awareness, and knowledge of mathematical thinking

(see e.g. [5], [6], [7]), and in general an application oriented approach. We follow the ideas of cognitive load theory [8] which focuses on the limited capacity of working memory of students. New ideas and concepts are introduced incrementally, always keeping the cognitive load small and focused. The formally exact representation is the result of the teaching process and not as often a starting point, making the constructive process of mathematical knowledge visible to the students and fostering the awareness between intuitive thinking and correct formulation of mathematics.

Our database of mathematical interconnections (see 2.1) supports the development of the spiral curriculum as examples and exercises are shared between the units and viewed from different perspectives.

The revised version of Bloom's taxonomy [6] (see also [15]) is used for a careful evaluation of the learning goals for each unit and exercises and quizzes are designed to reflect this information. This allows the student to assess their learning achievements and the growth of their knowledge continuously. Automatically corrected problems on different levels of content provide them with a differentiated feedback to support the learning process. Different types of exercises support the comprehension of ideas, the stabilization of acquired knowledge or the development of routine skills. Application-oriented exercises from science and engineering impart an understanding of the importance of the methods to the relevant application domains.

3.2 Serious games and gamification

Serious games are games that are designed in our context for educational purposes. Gamification can then be defined as the use of typical game design elements and scenarios in non-game contexts, see [9]. We develop a variety of such components for oHMint which create a new incentive for active participation. It is the aim of the current pilot project to test several approaches and choose the most successful ones for implementation in the entire course which will subsequently be developed. Some of our components are:

Badges: We are developing a system of badges that participants receive for completing certain tasks. The positive effect of badges has been documented, see [10]. This will however be an optional element since not everyone likes to be “distracted” by such matters.

Exercises in game form (competitive and non-competitive): For skills that require routine we are developing a gaming approach. The player is given a task such as “what is the derivative of the function $f(x)$ ”. If answered correctly, the player receives points and proceeds. After finishing the game, players can enter their score into a high score list. We are also investigating the option of enabling players to compete against each other in a duel or small groups and the possibility of a “Who wants to be

a millionaire?”-styled game. Developing and realizing these ideas and more general the whole course we benefit from close interaction with the e-learning group at Universität Hamburg as well as with student helpers also from the HafenCity University who provide valuable feedback from the target group of oHMint.

3.3 The advantage of the oHMint concept

Because of the modularisation (see 2.1) lecturers can design their own course by selecting units depending on their programme-level.

Innovative types of exercises are implemented: audio files train the transition between written and spoken mathematics; in “reversed” exercises the students shall find flaws in a given line of arguments; in “drawing” exercises they are asked to sketch the graph of a function with certain properties. Moreover, there is a large amount of quick checks, i.e. short questions or problems interspersed in the course where students can immediately check if they understood some new content or not.

In the available group instructing mode professors can not only compile a course suitable for their needs but also follow the learning progress of their students in a gradebook overview. Also, a flipped classroom version will be available (see 4). And last, a call centre run by integral-learning GmbH will be integrated to oHMint as it is already in the OMB+. Thus, participants of oHMint can contact a call centre for mathematical questions from 10 am till 8 pm via internal chat, telephone, skype (telephone and chat) or internal forum as a free service.

For more details on our didactical methods see [11].

4 Flipped Classroom

At many universities a typical mathematics lecture for engineers suffers from low intrinsic motivation of the students, their low competence in school mathematics, (which we hope to enhance through OMB+) and the procrastinating study style of many students.

Flipped classroom is an interactive teaching model that addresses this problem. The information and content instruction takes place outside of class time and class time is used for practice oriented activities. The concept has been successfully introduced in university mathematics and evaluated in recent years [12, 13, 14]. It has been shown that the average performance of students in exams is at least equal to the one of students instructed with traditional methods. Moreover, through flipped classroom students tend to have a better understanding of underlying concepts. The gap between better and weaker students tends to get smaller while the well performing students perform equally well as or better than with traditional classroom lectures.

The students get preparatory material focusing on the

basic concepts with a set of key questions guiding their studies and a quiz of control questions that have to be answered before class time. The quizzes are automatically corrected and give students helpful feedback and encourage them to prepare. In-class time is used for interactivity with more challenging problems that are followed up with homework.

In winter term 2018/19 the first prototype module for differential calculus will be used in a flipped classroom context for a freshmen class of Mathematics in a Geodesy and Geoinformation course at the HCU. The class time will be split. The first half will be an interactive session to handle feedback questions from students and presentations from students. The second half will be used for group work on special exercises to solidify and deepen the gained new mathematical competencies.

Implementing the first base unit of oHMint into a real teaching scenario at this early stage will result in valuable experience and feedback for the further development of the oHMint course.

5 Conclusions

As explained above, the production of the first oHMint base unit *Differential Calculus* is funded by the HOUU. It is meant as a sample for the remaining units of the course, testing new approaches in digital teaching and innovative types of exercises. The oHMint working group, currently a subgroup of the OMB+ consortium, intends to create attention for the project and raise further funding through this sample unit. Other working groups are encouraged to join the oHMint project and/or produce an oHMint unit themselves.

The main assets of oHMint are its flexibility in terms of teaching style, ranging from self-study to flipped classroom settings. Each unit will be translated into English once it is completed. A preliminary version of *Differential Calculus* will be put to the test in the winter term 2018/19 at HCU Hamburg as a flipped classroom version, feedback from HCU students and lecturers will be incorporated into the future development of oHMint.

Overall, we believe that oHMint poses an important step in the digitalization of STEM education in Germany. The broad supporting base guiding its development ensures that it is tailor-made for the needs of lecturers and students alike, and its modular structure makes it suitable for all institutions.

Acknowledgements. We gratefully acknowledge the financial support of the HOUU for the current project, the guidance and input of the OMB+ consortium (especially Volker Bach) in the development of oHMint and support beyond the technical lead by integral-learning GmbH.

References

- [1] OMB+ www.ombplus.de
- [2] Braun I, Schröder J et al. *cosh cooperation Schule:Hochschule – Mindestanforderungskatalog Mathematik (Version 2.0)*, 2014. https://lehrerfortbildung-bw.de/u_matnatech/mathematik/bs/bk/cosh/katalog/index.html
- [3] Barbas H, Schramm T. The Hamburg Online Math Test MINTFIT for Prospective Students of STEM Degree Programmes, *MSOR Connections*, 2018, 16(3): 43-51.
- [4] Bruner J. *The Process of Education*, Harvard University Press. Cambridge, Mass., USA, 1960.
- [5] Pintrich PR. The Role of Metacognitive Knowledge in Learning, Teaching and Assessing, *Theory into Practice*, 2002, 41(4): Revising Bloom's Taxonomy, 219-225.
- [6] Krathwohl DR. *A Revision of Bloom's Taxonomy: An Overview*, 2002. <https://www.depauw.edu/files/resources/krathwohl.pdf>
- [7] Zech F. *Grundkurs Mathematikdidaktik*, Beltz Verlag, Weinheim und Basel, 1996.
- [8] de Jong T. Cognitive load theory, educational research, and instructional design: some food for thought, *Instructional Science*, 2010, 38: 105-134.
- [9] Deterding S, Dixon D, Khaled R, Nacke LE. Gamification: Toward a definition, *CHI 2011*, 2011, Vancouver, Canada.
- [10] Denny P. The Effect of Virtual Achievements on Student Engagement, *CHI 2013: Changing Perspectives*, 2013, Paris, France, pp. 763-772.
- [11] Barbas H, Gasser I, Konieczny F, Lohse A, Seiler R. oHMint: Online Higher Mathematics for MINT Students - An Online Mathematics Course and Learning Platform, Preprint, to appear in the Proceedings of the 46th SEFI Annual Conference, 2018, Copenhagen, Denmark.
- [12] Epstein, J; *The Calculus Concept Inventory -- Measurement of the Effect of Teaching Methodology in Mathematics*, Notices of the AMS 2013, 60(8), <https://www.ams.org/journals/notices/201308/rnoti-p1018.pdf>
- [13] Maciejewski, W.; *Flipping the calculus classroom: an evaluative study*, Teaching Mathematics and its Applications: An International Journal of the IMA, Volume 35, Issue 4, 1 December 2016, Pages 187–201, <https://doi.org/10.1093/teamat/hrv019>
- [14] Cronhjort, M., Filipsson, L., Weurlander, M.; *Improved engagement and learning in flipped-classroom calculus*, Teaching Mathematics and its Applications: An International Journal of the IMA, , hrx007, <https://doi.org/10.1093/teamat/hrx007>
- [15] Anderson, LW., Krathwohl, D. (eds.); *A Taxonomy for Learning, Teaching, and Assessing: a Revision of Bloom's Taxonomy of Educational Objectives*. New York Longman, 2001.

An Object-Oriented Approach to Modelling Technical Systems

Dirk Eisenbiegler¹, Tilo Emmmler²

¹University of Furtwangen, Furtwangen, Germany, dirk.eisenbiegler@hs-furtwangen.de

²tilo-emmler@t-online.de

Abstract. This paper presents an object-oriented approach to programming technical systems in a modular fashion. Each technical system includes a physical model, a user interface and a controller. It is the intention of this approach to build up a set of technical components, user interfaces and controller programs in a modular fashion. The paper explains, how this approach can be applied for teaching in the computer science and engineering domain.

Introduction

Programming simulations for complex technical systems is challenging [6,7]. There are already different concepts for building such models in a modular manner [9]. This paper describes a systematic approach towards modelling physical systems using basic object-oriented programming techniques such as classes, instances, inheritance, inner classes, etc.. It is the intention of this approach to develop a toolbox of reusable technical components.

The approach presented in this paper is very well suited for teaching students in the computer science and engineering domain. The approach is based on object oriented techniques. Computer science students learn how to apply object oriented modelling techniques by the example of physical components, controllers and user interfaces. Besides, engineering students can learn how to build physical models, how to build control units and how to create user interfaces. It would overburden students to teach object oriented modelling, physical modelling, controller programming and user interface design all at the same time. Instead, it makes perfect sense to break down the complexity of “implementing a technical system” into its subtasks and address the different subtasks exercise by exercise.

This can only achieved if a software architecture is

used, where each subtask is represented by a different component and where the different subtasks can be implemented independently.

The approach presented in this paper is based on the Physolator simulation framework. Different kinds of tools are available for modelling and simulating physical systems. The Physolator framework is one of them [1,2,4,10]. Physolator has been designed to support an object-oriented developments style. In the Physolator framework, modern object-oriented techniques can be applied when implementing physical systems with their variables, differential equations and events.

Physolator is well designed for teaching. Other than well-established frameworks like Modelica or Matlab, you do not have to teach a proprietary programming language before you can get started. Physolator provides all basics you need for simulating physical systems, but the well-established simulation frameworks provide quite some advanced features, that Physolator does not have, such as algebraic equation solving. Different kinds of powerful libraries for different fields of applications have already developed for these frameworks.

1 Technical Components

A technical component contains a physical component. Besides the physical component, the technical component may also contain a controller and a user interface (see figure 1). The physical component describes the physical behaviour of the mechanical and electrical devices of the technical system. Every physical component shall be equipped with sensors and actuators. Sensors and actuators are technical devices. They are used to read physical values from the physical component and to write physical values to the physical component, respectively.

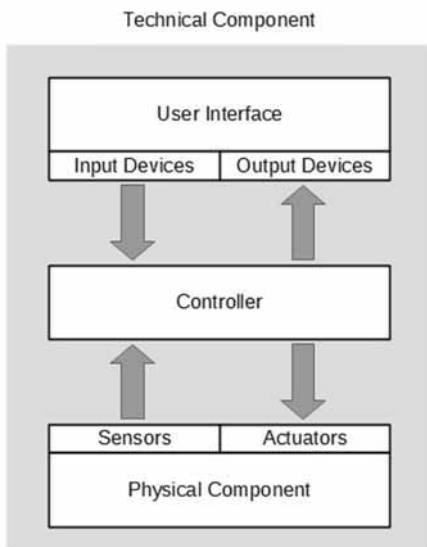


Figure 1: Technical Component.

User interfaces are devices attached to the physical components. The input data provided by the user specifies how the physical system is supposed to behave. This input data is forwarded to the controller. It is the controller's responsibility to make the physical component act according to the behaviour that is described by the user's input data. To achieve this, the controller interacts with the sensors and actuators of the physical component. From the perspective of a controller, sensor values are physical variables, that the controller can read and actuator values are physical variables that the controller is supposed to write.

It is also the controller's responsibility to inform the user about the current state of the technical system. There are various ways to achieve this. The controller may simply just write some sensor data to the user interface. The controller may also write some values to the user interface, that it has derived from the sensor data. The controller may also write some data to the user interface that represents the internal state of the controller.

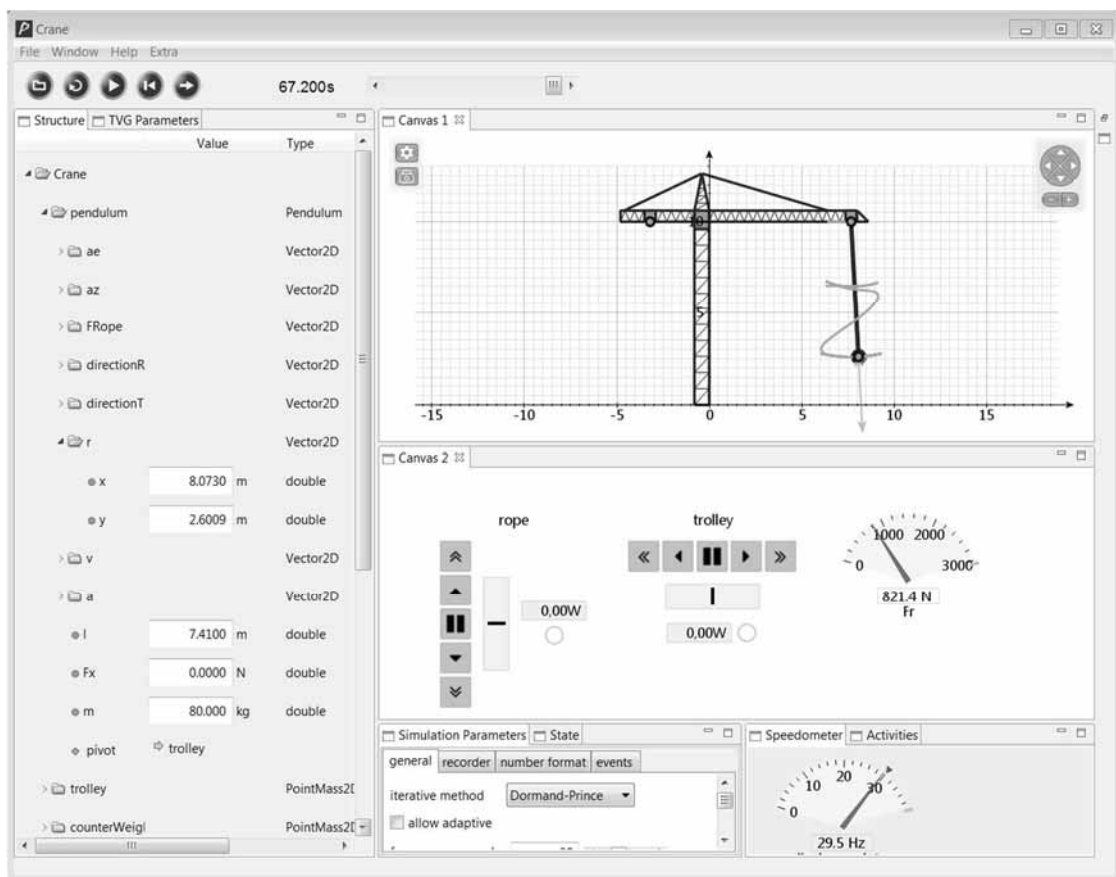


Figure 2: Physolator.

2 The Physolator Framework

Figure 2 shows the Physolator framework with a physical that has already been loaded. After loading a physical system to the Physolator, the simulation can be started. The physical system in figure 2 represents a crane. On the left, you can see the structure of the physical system with its components and subcomponents. The leaf elements of this hierarchy are scalar physical variables. For every element in this hierarchy, you can see its name, its type, its physical unit and its derivation relationship to other elements. Scalars also have an actual value that changes during simulation

Besides the physical system itself, the user can also program graphics components and attach them to the physical system. On the upper right hand side there is a crane canvas visually representing the current state of the physical system. Below there is a user interface. Both the visual representation of the crane and its user interface are graphics components that have been attached to the physical. Whenever the crane is loaded to Physolator, they are loaded as well.

The Physolator provides means for starting and stopping physical simulations. Simulation parameters define, how the simulation is carried out. You can choose from different kinds of ODE solvers, you can define the step width, the number of frames per seconds (graphics updates) etc. A built-in recorder stores the time course of a simulation run. After you have stopped a simulation, you can navigate forth and back in the time course of the simulation. By default, simulations are run in real time mode. Alternatively, you can also execute simulations as fast as possible. Built-in performance monitoring tools inform the user about the actual load and help the user to locate the performance bottlenecks.

3 The Crane

The physical system to be considered in this paper is a tower crane. The crane has a jib on the right hand side and a counterjib on the left hand side. A trolley is attached to the jib in such a way that it can be moved horizontally along the jib. A rope is attached to the trolley. At the lower end of the rope, a load can be attached. The crane also has a counterweight. It is attached at a fixed position of the counterjib. The counterweight cannot be moved.

The crane has two motors. One is attached to the trolley and one is attached to the rope. The trolley motor moves the trolley left and right. The rope motor is located

inside the trolley. It moves the rope up and down. The two motors are the only actuators of the crane. The electrical power that is applied to the motors determines the behaviour of the crane.

Cranes are used to transport a certain load from one location to another. A user interface has to be provided to operate the crane. With the help of the user interface, a person defines the behaviour of the two motors and thus the movement of the load.

4 Physical Components

The crane is built using three kinds of physical components: point masses, pendulums and motors. Therefore, we first define the mechanical behavior of a point mass, a pendulum and a motor. Each of these three physical devices will be represented by a Java class. The Java classes are named *PointMass2D*, *Pendulum* and *Motor*. The crane consists of two motors (one for the trolley and one for the rope) two point masses (one for the load and one for the trolley) and one pendulum with the load. Therefore, every *Crane* object consists of two instances of *PointMass2D*, one instance of class *Pendulum* and two instances of type *Motor*.

Class	Instances
<i>PointMass2D</i>	<i>Trolley</i> <i>counterWeight</i>
<i>Pendulum</i>	<i>Pendulum</i>
<i>Motor</i>	<i>trolleyMotor</i> <i>ropeMotor</i>

Figure 3: Physical components and their instances.

In our technical system, physical components shall be controlled by some selected physical variables. Such variables shall be called input variables. These variables are effected from outside the physical component. Input variables have an impact on the behaviour of the physical component, i.e. they define how the internal state changes.

Point masses (*PointMass2D*) are pretty simple physical components. The state of an instance of *PointMass2D* is represented by its position r and its velocity v . From outside, the state is controlled by setting the acceleration a . The acceleration a is the physical input

variable of this component.

A pendulum is similar to a *PointMass2D*. Also a pendulum has a point mass with an actual position r and an actual velocity v . Unlike the *PointMass2D*, the acceleration of the pendulum's mass shall not directly be provided. Instead, this acceleration is internally computed using the centripetal force. It takes several physical input variables to compute the centripetal force. First of all, a pendulum necessarily needs a link to some pivot point. In our crane example there is a pendulum with the load. The trolley is the pivot point of this pendulum and this pivot point is moving. The pivot point of our *Pendulum* class is an instance of type *PointMass2D*. Due to the link to the pivot, the behaviour of the pendulum depends on the behaviour of the pivot. The pendulum depends on the pivot's position, on its velocity and on its acceleration. Besides the pivot, there are two physical input variables also effecting the behaviour of the pendulum: ae and F_x . ae is the external acceleration. On the surface of the earth, the gravitational acceleration $(0, -9.81) \text{ m/s}^2$ applies to every mass and it therefore also applies to the mass of the pendulum downwards. In other scenarios, other external accelerations may apply such as accelerations due to electrical fields, magnetic fields or wind. The external acceleration need not necessarily be constant. This is why the external acceleration ae was designed as a physical input variable rather than a constant.

In this paper, motors describe a linear and one dimensional movement. The internal state of the motor is represented by a one dimensional position r and a one dimensional velocity v . Motors are to be attached to external physical devices, that the motor is supposed to move. This is why from the perspective of the motor the velocity v is a physical input variable. Its value has to be provided from the device the motor is attached to. F_0 is another physical input variable of the motor. It describes the force, that has to be applied to the motor to keep the its velocity v unchanged. Both v and F_0 depend on the state of the device the motor is attached to. The physical input variable F_a is used to control the behaviour movement from outside. F_a is used to change the velocity. F_a is the force that accelerates the component. As long as F_a is zero, the speed remains at its actual value. Both force F_0 and force F_a apply to the motor. The actual mechanical power is $(F_0 + F_a) \cdot v$.

F_a is directly applied to the device the motor is attached to. This is why this input variable is directly

forwarded to the output. The acceleration applied to the device the motor is attached to depends on F_a only. This acceleration is $m \cdot F_a$ where m is the mass of the device.

Let us assume, that the motor is an electrical device. In order to work the way it has been described, the motor needs access to the velocity v and the force F_0 from the device, the motor has been attached to. It takes sensors to measure these values. In the case of the crane it takes sensors to measure the velocity of the trolley, the radial velocity of the rope of the pendulum and the force F_{rope} that the rope applies to the pivot.

5 Physical components Implementation

Differential equations are used to describe the behaviour of physical components. The following programming code shows the implementation of the point mass. The class is named *PointMass2D*. An object of type *PointMass2D* consists of a mass m , a position r , a velocity v and an acceleration a . The annotations in the code define the derivation relationships between these physical variables and also the variable's physical units.

```
public class PointMass2D {
    @V(unit = "kg")
    public double m;
    @V(unit = "m", derivative = "v")
    public Vector2D r;
    @V(unit = "m/s", derivative = "a")
    public Vector2D v;
    @V(unit = "m/s^2")
    public Vector2D a;

    public PointMass2D(Vector2D r, Vector2D v,
        double m) {
        this.r = r;
        this.v = v;
        this.m = m;
        a = new Vector2D(0, 0);
    }

    public PointMass2D(double rx, double ry, double vx,
        double vy, double m) {
        this.r = new Vector2D(rx, ry);
        this.v = new Vector2D(vx, vy);
        this.m = m;
        a = new Vector2D(0, 0);
    }
}
```

In a *PointMass2D* object, the acceleration a is to be used

as the physical input variable. You simply have to assign an appropriate value to a in order to give the object the desired behaviour. The internal state of a point mass is represented by its position r and its velocity v . The acceleration defines the behaviour of the point mass, i.e. it defines how this state is changed.

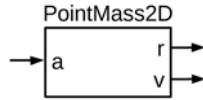


Figure 4: PointMass2D.

The *Pendulum* class inherits from *PointMass2D*. All physical variables and also their derivation relationships are inherited. Class *Pendulum* adds some more physical variables and a reference to a pivot point. The pivot point is a point mass.

```
public class Pendulum extends PointMass2D {

    public Pendulum(double rx, double ry,
        double vx, double vy, double m) {
        super(rx, ry, vx, vy, m);
    }

    @Crossref
    public PointMass2D pivot;

    @V(unit = "m")
    public double l = 1;
    @V(unit = "m/s^2")
    public Vector2D ae = new Vector2D(0, -9.81);
    @V(unit = "N")
    public double Fx = 0;
    @V(unit = "m/s^2")
    public Vector2D az = new Vector2D();

    @V(unit = "N")
    public Vector2D FRope = new Vector2D();
    public Vector2D directionR = new Vector2D();
    public Vector2D directionT = new Vector2D();

    public double radialPart(Vector2D p) {
        return footOfPerpendicularScalar(p, directionR);
    }

    public double tangentialPart(Vector2D p) {
        return footOfPerpendicularScalar(p, directionT);
    }

    public double getVr() {
        return radialPart(sub(v,pivot.v));
    }

    public void apply(Vector2D ae, double Fx) {
        this.ae.set(ae);
        this.Fx = Fx;
        l = dist(r, pivot.r);
    }
}
```

```
directionR.set(normalize(sub(r, pivot.r)));
directionT.set(-directionR.y, directionR.x);
az.set(mult(-pow(abs(tangentialPart(sub(v,
    pivot.v))), 2) / l, directionR));
Vector2D ax = mult(-Fx / m, directionR);
Vector2D ag = add(ae, invert(pivot.a));
FRope.set(mult(m, add(invert(ax), invert(az),
    footOfPerpendicular(ag, directionR))));
a.set(add(ax, az, footOfPerpendicular(ag,
    directionT), pivot.a);
}
}
```

The method *apply* inside the class *Pendulum* provides a means to apply input variable values to pendulum objects: a_e and F_x . By invoking the apply method, these two physical values are assigned to the pendulum and the internal variables of this component are computed in an appropriate manner.

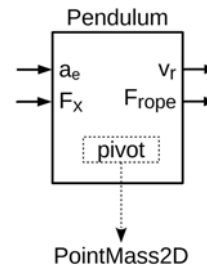


Figure 5: Pendulum.

The classes *Motor* is implemented in a similar manner. The state of a motor is represented by its actual position r and by its actual power P .

```
public class Motor {

    @V(unit = "m", derivative="v")
    public double s;
    @V(unit = "m/s")
    public double v;
    @V(unit = "N")
    public double F0;
    @V(unit = "N")
    public double Fa;
    @V(unit = "N")
    public double F;
    @V(unit = "J", derivative = "P")
    public double W = 0;
    @V(unit = "W")
    public double P;

    public void apply(double v, double F0) {
        this.v = v;
        this.F0 = F0;
        F = F0 + Fa;
        P = v * F;
    }
}
```

```

    }
}

```

Also the *Motor* class provides an *apply* method for assigning values to its physical input variables. The *apply* method assigns the actual velocity v and the actual external force F_0 . These values originate from the mechanical system the motor is attached to. Besides that, the motor provides another physical input variable F_a . This variable is used to control the motor's behaviour.

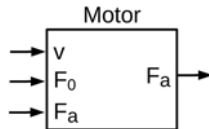


Figure 6: Motor.

6 Controllers

It is the controller's task to compute appropriate F_a values and apply them to the motors. A user interface could provide two simple rotary controls to control these two values. The F_a values applied to the motors result in accelerating them. However, with such a user interface it would not be not at all easy bring the device that is attached to the motor to hold, i.e. achieve, that the velocity of the device is exactly zero.

This is why a controller is added to the motor. The following class *MotorControlled* represents a controlled motor. A controlled motor is a motor with some extra control features. *MotorControlled* inherits from *Motor*. Therefore, *MotorControlled* inherits all the physical variables and features from the *Motor* class and adds two extra methods for controlling the motor's F_a value: *accelerate* and *breakDown*. The *accelerate* method is pretty simple. It is nothing but a setter for F_a . The *breakDown* method is more sophisticated. A controlled motor has an internal boolean variable named *breakingDown*. Initially, this value is *false*. When invoking the *breakDown* method the variable *breakingDown* is set to *true* and the breaking process starts. Breaking means applying a breaking acceleration *breakFa* in a direction that is opposite to the actual movement direction. The breaking shall automatically stop, when the velocity v has reached zero. Whenever the motor is in the *breakingDown* mode and the velocity v reaches zero, then the *stopBreakingHandler* is to be

invoked. In the *MotorControlled* class, it is the responsibility of method *g* to trigger the *stopBreakingHandler* at the right time. Physical event programming is used to achieve this. See [2,4] for details on this issue. As soon as the *stopBreakingHandler* is invoked, it stops the breaking process by setting *breakDown* back to *false* and by setting the acceleration F_a to zero.

```

public class MotorControlled
    extends Motor implements G {

    public boolean breakingDown = false;

    public void accelerate(double accelerateFa) {
        Fa = accelerateFa;
    }

    public void breakDown(double breakFa) {
        breakingDown = true;
        Fa = (v > 0) ? breakFa : -breakFa;
    }

    private Runnable stopBreakingHandler = () -> {
        Fa = 0;
        breakingDown = false;
    };

    @Override
    public void g(double t, AfterEventDescription aed) {
        if (!breakingDown)
            return;
        if ((Fa > 0) != (v > 0)) {
            aed.reportEvent(stopBreakingHandler,
                "motor has been stopped");
        }
    }
}

```

Figure 7 shows time diagram with the electric power of the two crane motors during a simulation run. The motors have been controlled by invoking the *accelerate* and *breakDown* methods. Approximately at time $t=2s$ the *accelerate* method of the rope motor is invoked and at about $t=2,5s$ the *breakDown* method is invoked. This results in the power peak according to the diagram. At about $t=3s$ the breaking process is finished, the power is back to zero and as a consequence, the motor does not move any more. Afterwards the same is applied to the trolley motor: first start the acceleration, then break down. Finally, the rope motor is accelerated again – this time to the opposite direction.

During acceleration and breaking the speed of the motors is constantly increased and decreased, respectively. If the force was constant, then this would result in a linear rise and decline of the power. However,

due to the oscillating pendulum, the forces applied to the motors are not constant. Therefore, the increase and decrease of the power is not linear.

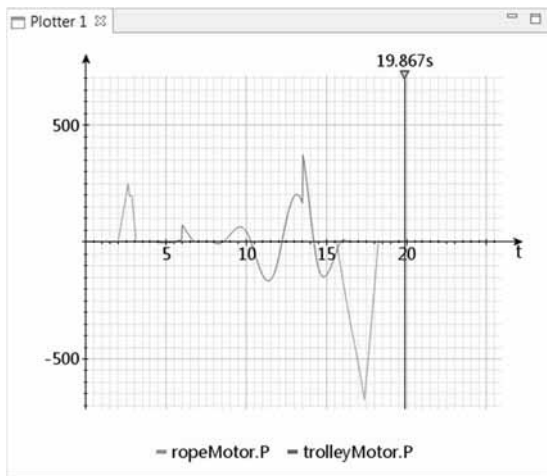


Figure 7: Controlled acceleration and breaking.

7 User Interfaces

The user interface must provide appropriate means for controlling and monitoring the state of the technical systems. A user interface should be well designed. A human user should easily be able to understand the relationship between the widgets and their meaning. Technical systems are to be used by humans for specific tasks. User interfaces have to be designed in such a way that the user can accomplish his tasks as quickly and efficiently as possible.

Figure 8 shows the user interface for the crane. It consists of three parts: one control element for the rope motor, one for the trolley motor and a pointer instrument for the total force that applies to the rope. A crane is used to transport a load from one point to another. The user shall solve this task as fast as possible without hitting an obstacle and without overloading the motors or breaking the rope.

The user interface of the crane consists of two motor control widgets and a pointer instrument. Each instance of the motor control widgets is linked to one of the two motors. The elements of the motor control widget can be arranged either vertically or horizontally. For sake of a better usability, the rope elements have been arranged vertically and the trolley elements have been arranged horizontally. The pointer instrument is linked to a physical value: the actual force that applies to the rope.

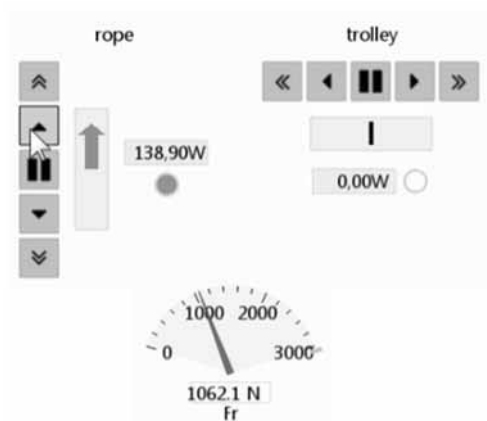


Figure 8: User Interface.

Both motor control widgets work the same way. Let us have a look at the rope motor. Here the elements are arranged vertically. The buttons \uparrow \downarrow $\|\|$ \downarrow \downarrow are used to change the acceleration. As long as you press \uparrow or \downarrow a constant accelerating force F_a is applied. When pressing \uparrow this force is positive and results in continuous increment of the speed. The actual speed is represented by the red arrow on the right hand side of the buttons. The length of the arrow is proportional to the motors velocity. If the motor is stopped, then the arrow disappears and there is a black line instead. In figure 8, the rope motor is currently moving upwards and the trolley motor is currently stopped. As soon as the user releases the acceleration buttons \uparrow \downarrow the motor starts an automatic break process by applying a fixed breaking acceleration. A short time later, the rope stops moving. The buttons \uparrow \downarrow are similar to \uparrow \downarrow , except that the motor does not start breaking when the buttons are released. At the moment of the button release the accelerating force F_a is set to zero instead. As a result, the motor continues running with a constant velocity – the velocity at the moment when the button has been released. As long as there is no further user interaction, the rope will constantly move with this speed. In this situation the user can press the button $\|\|$ to manually break the motor down.

On the right hand side of the velocity arrow there is a label with the motor's actual power. The power is displayed as a number. For sake of a better readability, this number is updated only two times per second. The circular label underneath indicates whether the motor is

overloaded, i.e. if the electrical power exceeds a predefined level. A filled circular label ● indicates that the motor is overloaded whereas a non-filled circular label ○ indicates that it is not.

8 COMPOSING THE CRANE

As you can see in figure 2, two graphics components are attached to the crane. Canvas 1 in figure 2 shows a two dimensional picture of the crane in its current state. Canvas 2 in figure 2 is an interactive graphics component with the user interface. These two classes are named *CraneTVG* and *CraneUserInterfaceTVG*. Both graphics have been tailored to the needs of the crane. For sake of space, the program code of the graphics components of the crane is not listed in this paper. Physolator supports reuse concepts for programming graphics that are attached to physical systems. Have a look at [2,4] for details on this topic.

The following program code shows the *Crane* class. *Crane* represents the complete physical system. Attached are two graphics components. The method *initGraphicsComponents* makes sure, that whenever the physical system *Crane* is loaded to the Physolator, the two graphics components *CraneTVG* and *CraneUserInterfaceTVG* are loaded as well.

In the upper part of the *Crane* program there are some constants *jibWidth*, *counterjibWidth*,...*g*. In the next section the technical components of the crane are defined: *pendulum*, *trolley*, *counterWeight*, *ropeMotor* and *trolleyMotor*. The technical components are initialized right at the point where they are declared. The constructor of class *Crane* puts the crane in a specific initial state. It defines the initial positions of the trolley, the counterweight and the pendulum. Furthermore, it defines the relationship between the trolley and the pivot: the trolley is the pendulum's pivot.

Physical systems run inside the Physolator framework must provide a method named *f*. Method *f* is used to compute the depend physical variables. Usually, this is the place where you would find physical formulas. In this case, however, the entire class *Crane* does not contain any physical formula. The crane is a composition of several physical components and the physical formulas are located inside the technical components. The program code of class *Crane* simply composes these components. It is the responsibility of method *f* to interconnect the technical components. Connecting

technical components means transferring physical values from one technical component to another.

```
public class Crane extends PhysicalSystem {

    public double jibWidth = 8;
    public double counterjibWidth = 4;
    public double jibHeight = 0.6;
    public double towerHeight = 10;
    public double towerWidth = 0.8;
    public double turntableHeight = 0.4;
    public double peakHeight = 2;
    public Vector2D g = new Vector2D(0,-9.81);

    public Pendulum pendulum =
        new Pendulum(0, 0, 0, 0, 80);
    public PointMass2D trolley =
        new PointMass2D(0, 0, 0, 0, 120);
    public PointMass2D counterWeight =
        new PointMass2D(0, 0, 0, 0, 800);
    public MotorControlled ropeMotor =
        new MotorControlled();
    public MotorControlled trolleyMotor =
        new MotorControlled();

    public Crane() {
        trolley.r.set(0.8 * jibWidth, towerHeight);
        counterWeight.r.set(
            -towerWidth - 0.6 * counterjibWidth,
            towerHeight);
        pendulum.r.set(
            0.8 * jibWidth, 0.65 * towerHeight);
        pendulum.pivot = trolley;
    }

    public void f(double t, double h) {
        trolley.a.set(trolleyMotor.Fa / trolley.m, 0);
        pendulum.apply(g, ropeMotor.Fa);
        ropeMotor.apply(pendulum.getVr(),
            -pendulum.FRope.abs());
        trolleyMotor.apply(-trolley.v.x, pendulum.FRope.x);
    }

    public void initGraphicsComponents(
        GraphicsComponents g, Structure structure,
        Recorder recorder, SimulationParameters sp) {
        g.addTVG(new CraneTVG(
            this, structure, recorder));
        g.addTVG(
            new CraneUserInterfaceTVG(this, structure));
    }
}
```

Figure 9 graphically describes how the above Java program builds up a crane from the technical components. The dashed arrow indicates, that the pendulum refers to the trolley. This indicates that the trolley position is the pivot point of the pendulum.

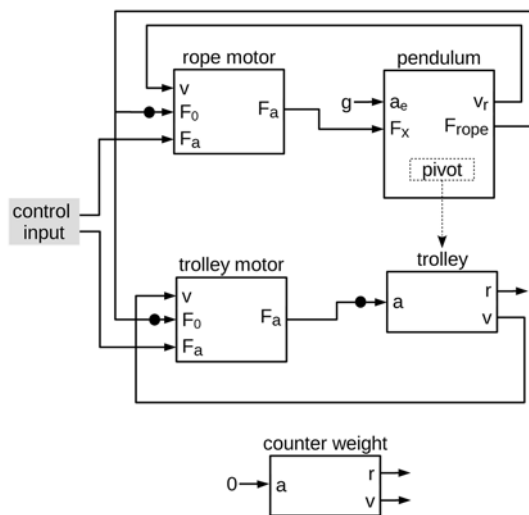


Figure 9: Crane.

On the left hand side of figure 9 there are two control inputs: the F_a values of the two motors. As explained earlier, the user interface does not directly write values to these physical input variables, but instead uses the control methods from class *MotorControlled*.

The solid line arrows in figure 9 indicate, that physical variable values are forwarded from one component to the next. These arrows correspond to the program code inside method f . In many cases, the physical output variables are one by one forwarded to an input variable of some other technical component. In some cases, however, some simple calculations have to be performed. In figure 9, the three filled black circles on the arrows indicate, that such a simple calculation is performed. Simple means, that mathematical expression are applied where the expression only depends on the output value and some constants. Let us for example have a look at the arrow pointing from the trolley motor to the trolley. It forwards the trolley motor's F_a value to value a of the trolley. This physical value cannot be forwarded as is. The trolley's input expects an acceleration rather than a force. Also the trolley expects a two dimensional value and the force F_a provided by the motor is one dimensional. F_a shall accelerate the trolley horizontally. The trolley shall not be moved vertically. Therefore, the acceleration to be applied to the trolley consists of an x -part and a y -part, where the y -part is zero and the x part is calculated by dividing the motor's F_a value by the mass of the trolley. In the program code from of class *Crane*,

you can find the assignment corresponding to this arrow:

```
trolley.a.set(trolleyMotor.Fa / trolley.m, 0);
```

9 Conclusions

The approach presented in this paper is based on the Physolator framework. Object-oriented techniques have been applied to build models for complex physical systems that have been built by composing basic technical components. A controller and user interface have been added to the technical system.

So far, only some few components have been implemented and only a single physical system has been built. It is the intention of the authors to extend this component library and to build other physical systems by composing the components of the library. Furthermore, a set of exercises for teaching is to be derived from this approach.

References

- [1] Eisenbiegler, D., 2015. The Software Architecture of the Physolator – a Physical Simulation Framework. In MSAM 2015, Conference on Modelling, Simulation and Applied Mathematics, Atlantis Press, pp. 61-64.
- [2] Eisenbiegler, D., 2016. Object Oriented Modeling and Simulation with the Physolator – Getting Started, Available at: <https://opus.hs-furtwangen.de/frontdoor/index/index/docId/614>.
- [3] Eisenbiegler, D., 2016. A Generic Particle Modeling Library for Fluid Simulation. In AMSM 2016, Conference on Applied Mathematics, Simulation and Modelling, Atlantis Press.
- [4] Eisenbiegler, D., 2015. Objektorientierte Modellierung und Simulation physikalischer Systeme mit dem Physolator, BoD Norderstedt.
- [5] Eisenbiegler, D., 2016. An Object Oriented Library for Acoustics Simulation Based on the Physolator Simulation Framework, In CMSAM 2016, Conference on Modeling, Simulation and Applied Mathematics, DEStech Publications.
- [6] Bala, B. Ashad, F., Noh K., 2017. System Dynamics, Springer.
- [7] Karl, C., Ibbs, W., 2016. Developing Modular-Oriented Simulation Models Using System Dynamics Libraries, Springer.
- [8] Rose, W., Eisenbiegler, D., 2017, Mixed Fluid and Rigid Body Simulations - An Object Oriented Component Library Based on the Physolator Framework, in Simultech 2017, SciTePress.
- [9] Disselkamp, J., Junglas, P. et. al., 2018, Implementing the Argesim C21 Benchmark with Modelica Components, in ASIM STS 2018, ARGESIM Verlag Wien.

- [10] Michael Tiller, Introduction to Physical Modeling with Modelica. Kluwer Academic Publishers, Dordrecht 2001, ISBN 0-7923-7367-7.

Online Komponenten der Lehre an der TU Wien

Andreas Körner^{1*}, Stefanie Winkler¹, Ruth Leskovar¹, Franziska Gorgas¹

¹Institut für Analysis und Scientific Computing, Technische Universität Wien, Wiedner Hauptstraße 8-10, 1040 Wien, Österreich; *andreas.koerner@tuwien.ac.at

Abstrakt. Dieser Beitrag gibt einen Überblick über die online Tools, die an der Technischen Universität Wien in der Lehre eingesetzt werden. Dabei werden die Merkmale der einzelnen Komponenten vorgestellt, sowie ihre Vor- und Nachteile im Einsatz gegenübergestellt. Dabei wird erläutert, wie diese Tools in den verschiedenen Lehrveranstaltungen eingesetzt werden. Die Anwendung dieser Elemente reicht von freiem Üben bis hin zum Assessment im Rahmen von Hausübungen, Übungstests und Vorlesungsprüfungen. Schlussendlich wird eine Evaluation der Prüfungsergebnisse der letzten Jahre vorgestellt und dabei auch das Feedback der Studierenden diskutiert.

Einleitung

Wenn man heute ein öffentliches Verkehrsmittel betritt, sieht man viele junge Leute angestrengt in ihr Mobilgerät starren. Diese Entwicklung, aber auch die Fortschritte in anderen technischen Bereichen führen dazu, dass man von der heutigen Jugend als „digital natives“ [2] spricht. Natürlich ist diese Formulierung nicht zu verwechseln mit Personen, die alle technischen Details der verwendeten Geräte wirklich verstehen. Die Jugend von heute weist eine sehr hohe Akzeptanz für technische Hilfswerkzeuge auf und hat daher auch keine Probleme sich mit neuen Medien, technischen Geräten oder modernen Applikationen auseinanderzusetzen. Diese Entwicklung beeinflusst auch die Unterrichtspraxis, beginnend mit der Volksschule, bis hin zur Universität, siehe [1]. Jedoch haben viele Universitäten nicht die Mittel diesen Fortschritt im Lehrbereich im nötigen Maß zu unterstützen und umzusetzen.

An der TU Wien lässt sich folgendes Bild zeichnen: Die Infrastruktur in den Hörsälen und Seminarräumen unterstützt die Verwendung von Notebooks durch die Verfügbarkeit von WLAN am gesamten Universitätsgelände. Das Aufladen der Akkus dieser elektronischen Geräte stellt schon eine Herausforderung dar, weil die Elektroinstallation aus den 70er Jahren des letzten Jahrhunderts stammen. Gerade in den älteren Gebäudeteilen sind Steckdosen eher Mangelware und unvorteilhaft verteilt. Allerdings wird die Möglichkeit des Streamings von

Lehrveranstaltungen an der Universität durch entsprechende Infrastruktur unterstützt. Die Nachverarbeitung der aufgenommenen Materialien ist anschließend von den Vortragenden bei Bedarf selbst zu erledigen. Ein weiterer Diskussionspunkt ist die Verfügbarkeit von EDV-Räumen, welche leider tendenziell immer weniger werden. Argumentiert wird diese Entwicklung dadurch, dass die meisten Studierenden über eigene Notebooks bzw. Tablets verfügen, welche bei Bedarf mitzubringen sind. Dies ist grundsätzlich richtig, bringt aber einige andere Probleme mit sich. Erstens ist der Besitz eines Notebooks oder Tablets an der Universität kein Muss. Zweitens ist natürlich auch bei der Verwendung der eigenen Geräte nicht sichergestellt, dass die Software auf dem neuesten Stand ist und eine reibungslose Verwendung in den Lehrveranstaltungen gewährleistet werden kann. Es gibt leider auch nicht die finanziellen Möglichkeiten, Geräte anzuschaffen und Studierenden zur Verfügung zu stellen, wie es an anderen Universitäten der Fall ist [3]. Problematisch wird dieser Umstand erst recht, wenn Lehrende Tests oder Prüfungen auf einem Online-System durchführen wollen. Nicht selten passiert es, dass Studierende mit ihrem eigenen Notebook zur Prüfung erscheinen, jedoch nach der Hälfte der Zeit technische Probleme mit dem Gerät auftreten. In solchen Fällen wäre es dann notwendig, schnellstmöglich einen Ersatz zu organisieren. Einerseits kann dies mittels zur Verfügung gestellten Notebooks abgewickelt werden, oder aber der oder die Betroffene wechselt in einen nahegelegenen EDV-Raum und beendet so die Prüfung regulär.

In der Forschungsgruppe Mathematische Modellbildung und Simulation (MMS) werden seit 2006 unterschiedliche online Tools in der Lehre, sowohl in Vorlesungen als auch Übungen, eingesetzt. Diese werden zusätzlich zu Erklärungs- und Übungszwecken auch für Tests und Prüfungen verwendet. Die unterschiedlichen Tools und deren Verwendung werden im Folgenden vorgestellt. Abschließend werden die Einsatzmöglichkeiten von online Plattformen in der Lehre evaluiert und in Hinblick auf die Entwicklung der Infrastruktur diskutiert.

1 Online Tools

Dieser Abschnitt beschreibt unterschiedliche online Tools, welche an der TU Wien eingesetzt werden. Im ersten Teil werden Hilfsmittel vorgestellt, die von der Technischen Universität Wien direkt angeboten und betreut werden und von den Lehrenden eingesetzt werden können. Der zweite Abschnitt dieses Kapitels beschäftigt sich mit den von der Forschungsgruppe MMS betreuten online Plattformen.

1.1 Diverse Plattformen an der TU Wien

An der TU Wien werden alle Lehrveranstaltungen über ein Content Management System (CMS) verwaltet. In diesem System werden die Studierendendaten erhoben und gespeichert. Zusätzlich zur Verwaltung der Studierendendaten werden die Anmeldungen zu Lehrveranstaltungen (LVA) und Prüfungen über dieses CMS abgewickelt. Daher werden alle organisatorischen Tätigkeiten, wie Raumreservierungen und Lehrveranstaltungsdetails auf dieser Seite administriert. Dieses System wurde eigens an der TU Wien konzipiert und wird stetig weiterentwickelt um sich den wachsenden Ansprüchen der Administration anzupassen. Jede LVA an der TU Wien besitzt eine Beschreibung, außerdem werden hier auch Informationen über Uhrzeit und Ort angekündigt. Nach der Anmeldung zu einer LVA können Studierende gegebenenfalls auch zusätzlich hochgeladene Informationen oder Materialien einsehen. Falls es sich um eine Übung handelt, ist es auch möglich Gruppenanmeldungen zu verwalten. Für einen durchschnittlichen Lehrenden an der TU Wien erfüllt diese Plattform also alle notwendigen Anforderungen, um eine Lehrveranstaltung organisatorisch abwickeln zu können.

Jene Lehrende, die nach dem Erstellen der LVA im CMS ein E-Learning System verwenden wollen, bedienen sich einer weiteren Plattform namens TUWEL, kurz für TU Wien E-Learning. Dies ist eine auf Moodle basierte Plattform, die direkt mit der Authentifizierung des CMS verbunden ist. Daher erlaubt ein einmaliges Anmelden im System auch die Benutzung von TUWEL. Nur ca. 30% der Lehrveranstaltungen an der Technischen Universität Wien sind mit einem Kurs ebenfalls in TUWEL vertreten. Ein Großteil dieser TUWEL-Kurse werden zum Hochladen und Strukturieren der Vorlesungsunterlagen verwendet, da Moodle eine Gliederung in unterschiedliche Module ermöglicht. Diese Module können einerseits datumsorientiert, aber auch inhaltsorientiert aufgeschlüsselt werden. Anwendung findet TUWEL vor

allem in der Administration von Übungen, vorzugsweise bei LVAs mit großen Teilnehmerzahlen. In diesem Fall bietet diese Plattform verschiedene Funktionen an, die diesen Prozess vereinfachen. Beispielsweise kann man gruppenabhängig unterschiedliche Materialien zur Verfügung stellen. Weiters wird es in Übungen vor allem dafür verwendet sogenannte Kreuzerlisten abzubilden. In früheren Zeiten wurde ein Zettel an die Seminarraumtür gehängt, bei der man eintragen musste, welche der Hausübungsbeispiele man gerechnet hatte und an der Tafel präsentieren könnte. Heutzutage kann dies online in TUWEL für alle zeitgleich abgewickelt werden. Studierende markieren im Kurs direkt, welche der Beispiele sie vorbereitet haben. Auch für selbstständige Gruppenanmeldungen für Kleinprojekte oder die Anmeldung zu Klausuren und mündlichen Prüfungen bietet TUWEL passende Werkzeuge.

In den Lehrveranstaltungen der Forschungsgruppe MMS werden darüber hinaus auch noch andere Funktionen von TUWEL genutzt. Hierzu zählt zum Beispiel die Formulierung eines Feedbackbogens, um die Administration und Inhalte der LVA besser evaluieren zu können. Ebenso werden häufig Abstimmungen durchgeführt, um ein Bild des Wissensstands in den Vorlesungen zu erheben. In den Lehrveranstaltungen des ersten Semesters wird auch ein Glossar angelegt, um die wichtigsten Inhalte wie in einem kleinen „Wiki“ zusammenzufassen. Dies ermöglicht es den Studierenden am Beginn des Studiums größere Unterstützung und mehr Anleitung zu geben.

1.2 MMT – Mathematics, Modelling & Tools

Die Motivation der Forschungsgruppe neue Methoden und innovative Techniken in der Lehre einzusetzen, war immer schon hoch. Schon 2006 wurden online Beispiele für Lehrveranstaltungen im Bereich der Modellbildung und Simulation entwickelt und auf einem externen Server, genannt MMT, administriert. Diese Beispiele veranschaulichen das Zusammenspiel und die Wirkung von Parametern und illustrieren Systemverhalten von einfachen Modellen. In der Vorlesung werden diese Beispiele im Vortrag verwendet, um erklärte Methoden und Praktiken anhand eines Anwendungsbeispiels zu visualisieren. Studierende können diese Beispiele dann zuhause verwenden um mit den Parametern experimentieren und unterschiedliche Verhalten des Systems zu analysieren. Zusätzlich haben Sie die Möglichkeit den originalen Programmcode des Beispiels herunterzuladen und ihn eigens

zu adaptieren bzw. den Code zu analysieren. Abbildung 1 zeigt den schematischen Aufbau anhand eines Beispiels.

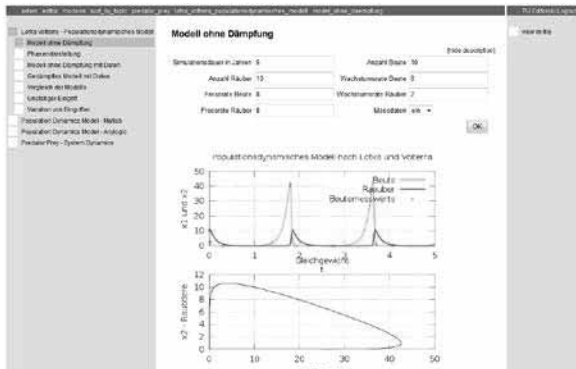


Abbildung 1: Dieses Bild stellt den schematischen Aufbau, die Beispielsammlung sowie den Parameter und Ergebnisbereich der online Plattform MMT dar.

TUWEL bietet zusätzlich zu den bereits erwähnten Funktionalitäten auch noch die Möglichkeit, eine Testumgebung zu gestalten, die sowohl Multiple Choice als auch aus numerischen Fragen unterstützt. Es können hier mehrere statische Fragen erstellt werden, aus denen dann in einem Testfall zufällig gezogen werden kann. Dieses Feature wurde genutzt, um mit den Simulationsbeispielen am MMT System Testfragen zu konstruieren. Die Studierenden erhalten die Aufgabenstellungen sowie einen Link des zugehörigen Beispiels in TUWEL. Dort müssen Studierende die verfügbaren Parameter mittels direkter Eingabe anpassen bis das gewünschte, in der Aufgabenstellung beschriebene, Verhalten des Systems eintritt und die Frage in TUWEL durch Auslesen der passenden Werte beantwortet werden kann. Dieses online Tool wurde in allen Modellbildung- und Simulationslehrveranstaltungen der Forschungsgruppe eingesetzt, sowohl für Präsentationen als auch in Tests.

1.3 Maple T.A.

In den Lehrveranstaltungen, welche sich mit der Mathematik Grundlehre auseinandersetzen, wird ein kommerzielles online Tool eingesetzt. Maple T.A. steht für Maple Testing and Assessment und ist eine Plattform basierend auf dem Computeralgebra-System Maple. Dieses System ermöglicht die Erstellung diverser mathematischer Fragestellungen mit der Möglichkeit, Antworten von Studierenden algebraisch auf Richtigkeit zu überprüfen. Durch Randomisierung von Parametern, Termen und Funktionen können die erzeugten Beispiele öfters

hintereinander geübt werden, ohne das gleiche Beispiel wieder und wieder zu erhalten. Zusätzlich zu der Möglichkeit Eingaben von Studierenden nicht nur auf numerische, sondern auch auf algebraische Richtigkeit zu untersuchen, kann die Antwort nicht nur direkt verglichen, sondern auf unterschiedliche gewünschte Eigenschaften untersucht werden. Diese Möglichkeit eröffnet eine andere Art der Fragestellung und erweitert somit die Möglichkeit an Übungsbeispiele und die erlernbaren Fähigkeiten.

Dieses online Tool wird auf unterschiedliche Arten eingesetzt. Einerseits können die entwickelten Aufgaben den Studierenden über TUWEL direkt und ohne zusätzliche Login-Daten in thematisch strukturierten Übungssammlungen zur Verfügung gestellt werden. Die Aufgabensammlungen werden dabei thematisch an die Inhalte der Vorlesung und Übung angepasst und in strukturierten Abschnitten zur Verfügung gestellt, wie in Abbildung 2 dargestellt. Dieselben Aufgaben werden dann genutzt, um Übungstests während des Semesters zusammenzustellen und diese dann über TUWEL abzuwickeln. Im Gegensatz zur verpflichteten Übung, in der Studierende ihre vorbereiteten Aufgaben präsentieren und die mathematischen Hintergründe erläutern müssen, stellen diese Tests eine rein handwerkliche Herausforderung dar. Hier soll überprüft werden, ob Studierende die mathematischen Methoden in einer Prüfungssituation abrufen und richtig einsetzen können. Es wird nicht verlangt, theoretische Beweise oder ähnliches zu führen, denn hier steht das praktische Rechnen im Vordergrund.

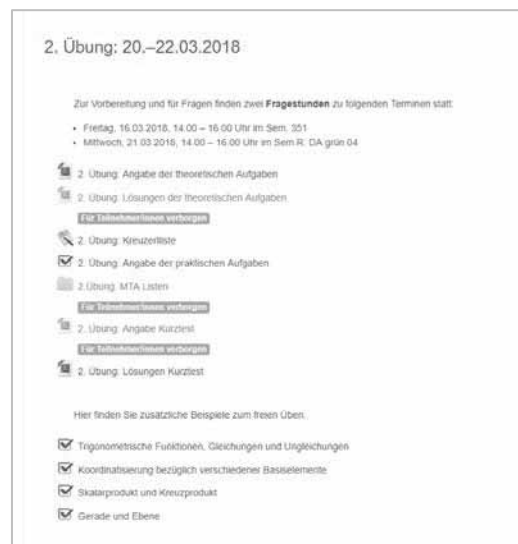


Abbildung 2: Hier sieht man den strukturellen Aufbau der online Beispiele im der Moodle Umgebung.

Abseits von freiwilligem Üben und den Tests wird Maple T.A. seit 2012 auch für die Durchführung von Vorlesungsprüfungen genutzt. Im Gegensatz zu den Übungstests bestehen die Aufgaben nicht nur aus Rechenbeispielen, sondern enthalten auch Fragen zum theoretischen Hintergrund der Beispiele. Die Gesamtpunkte der Prüfung entfallen zu ca. 15-25% an theoretische Fragestellungen und zu 75-85% an praktische Rechenbeispiele. Hierbei sei hervorgehoben, dass die theoretischen Fragestellungen immer in Bezug mit einer praktischen Rechenaufgabe gestellt werden. Da es sich bei den LVAs um Mathematik für Ingenieure handelt, sind langwierige Beweise nicht Teil der gewünschten Kompetenzen. Diese Prüfungen werden teilweise für die Benotung der Lehrveranstaltung direkt herangezogen oder aber dienen als Grundlage für die darauffolgende mündliche Prüfung.

2 Praktische Umsetzung

Die in Kapitel 1 erwähnten online Tools werden in den unterschiedlichen Lehrveranstaltungen kombiniert eingesetzt. In welcher Form dies stattfindet, wird im Folgenden dargestellt.

2.1 Einsatz in Lehrveranstaltungen der Modellbildung und Simulation

In den LVAs der Mathematischen Modellbildung werden die Beispiele am MMT-Server zur Erklärung in der Vorlesung herangezogen und dienen zur Übung sowie zur Prüfungsvorbereitung. Die Prüfung an sich, findet am eigenen Laptop im Hörsaal statt. Sie ist eine Mischung aus theoretischen und praktischen Fragen der Themen aus der Vorlesung und dauert eine Stunde. Bei dieser Art von Prüfung gibt es zwischen 3-300 Teilnehmer, abhängig von der Studienrichtung. Bei der Lehrveranstaltung für Maschinenbau und Wirtschaftsingenieurwesen Maschinenbau sind es jährlich 200-300 Studierende. Hier kam es auch schon zu technischen Problemen aufgrund der großen Anzahl an Studierenden. In den Anfängen waren die WLAN-Router mit der Zahl der Studierenden überfordert. Dies hat sich über die Zeit verbessert, da die Infrastruktur dem Fortschritt angepasst und die Anzahl an Router merklich erhöht wurde. Andererseits waren auch stellenweise mit der Serverauslastung Probleme aufgetreten. Dies wurde gelöst, indem die Prüfungen in zwei bis drei, sich überschneidende Zeitslots aufgeteilt wurden.

2.2 Einsatz in den Grundvorlesungen Mathematik

Auch die Testumgebung Maple T.A. läuft nach einem ähnlichen Schema ab. Studierende müssen sich im Vorfeld über TUWEL zu einem der angebotenen Zeitslots anmelden. Diese Zeitslots finden dann leicht überschneidend in diversen Seminarräumen, Hörsälen und EDV-Räumen statt. Abhängig von der Lehrveranstaltung und Prüfung kann dieser Zeitslot 30 oder 60 Minuten betragen. Die Prüfungen der Vorlesungen verlaufen grundsätzlich analog. Es gibt nur zwei wesentliche Unterschiede, erstens dauert diese Prüfung zwei volle Zeitstunden. Andererseits müssen Studierende ihre Mitschrift und Ausarbeitung abgeben, da diese zur Benotung und Nachvollziehbarkeit herangezogen werden. Bei den Tests ist dies nicht notwendig, da es hier lediglich um eine Überprüfung der Rechenfertigkeiten geht.

3 Besonderheiten von online Prüfungssituationen

In den letzten Jahren wurden die Lehrveranstaltungsstrukturen und Prüfungskonzepte immer wieder überarbeitet, um den Lernprozess bestmöglich zu unterstützen.

3.1 Vor- und Nachteile

Der Einsatz von online Tools hat seine Vor- und Nachteile. Eine online Lernplattform ermöglicht individuelles Lernen und dabei aber auch ein sofortiges Feedback. Ob Studierende vormittags, nachmittags oder in der Nacht lernen ist ganz ihnen überlassen. Sie bekommen vom online System zu jeder Zeit ein Feedback über die Richtigkeit ihrer Eingaben und können damit ihren Lernprozess vorantreiben. Ein weiterer Vorteil ist natürlich, dass die randomisierten Zahlen und Funktionen in den gestellten Aufgaben Abschreiben vom Nachbarn und Schummeln fast unmöglich machen. Dies stellt auch eine Erleichterung für Lehrende dar. Wenn man mehrere Prüfungen im Semester zusammenstellen muss, kann man ohne Probleme Beispiele immer wieder einsetzen, da sie generell nicht genau gleich sein werden, sondern nur die gleiche Struktur aufweisen. Dafür haben Studierende immer Zugang zu potenziellen Prüfungsbeispielen, da sie zum Üben zur Verfügung stehen.

Ein Nachteil der automatischen Bewertung solcher Systeme liegt wohl ebenfalls auf der Hand. Computergenerierte Beispiele und deren Bewertungsalgorithmen be-

werten nur so gut, wie die Software dahinter es ermöglicht, bzw. so gut wie der Algorithmus für die Bewertung ist. Beschäftigt man sich nur oberflächlich mit einem so mächtigen Programm wie Maple T.A. kann es natürlich schwer sein, die Beispiele auf das gewünschte Niveau zu bringen. Standardmäßig ist die algebraische Auswertung der Differenz des richtigen Ergebnisses und der Antwort des Studierenden als Bewertung voreingestellt. Falls diese Differenz Null ist, bekommt der Studierende alle zu erreichenden Punkte auf das Beispiel. Sollte man hier einen Fehler gemacht haben, verliert man die Punkte dieser Eingabe. Vergleicht man dieses Bewertungssystem mit dem der handschriftlichen Prüfung, gibt es viele Studierende und auch Lehrende, die der Meinung sind, dass unter Umständen ähnlich gemachte Fehler in schriftlichen Prüfungen Teilpunkte eingebracht hätten. Abhängig vom Ziel der Lehrveranstaltung kann es Fehler geben, die toleriert werden sollten und andere, die nicht tolerierbar sind. Diese unterschiedlichen Sichtweisen sind aber nicht nur vom Fach, sondern auch von der Institution abhängig. Mit dem richtigen Bewertungsalgorithmus können unterschiedlichste Ansprüche an die Bewertung einzelner Beispiele erfüllt werden.

3.2 Einsatz der online Ressourcen

Für die hier erwähnten Kurse wurde das Bewerten auf unterschiedliche Art abgewickelt. In den kurzen Tests, die im Laufe des Semesters abgehalten werden, geht es in erster Linie darum, dass die Studierenden mindestens die Hälfte der Aufgaben in der verfügbaren Zeit fehlerfrei berechnen können und damit ihre Rechenfertigkeiten unter Beweis stellen müssen. Die Punkte dieser Tests gehen nicht weiter in die Beurteilung der Lehrveranstaltung ein, sondern stellen lediglich eine notwendige Bedingung für das positive Abschließen der Lehrveranstaltung dar. Bei den Vorlesungsprüfungen hingegen ist dies natürlich nicht das Ziel. Studierende haben hier zwei Stunden Zeit, Beispiele zum gesamten Semesterstoff zu bearbeiten. Daher wurde entschieden, bei diesen Prüfungen die Ausarbeitungen der Studierenden abzusammeln und für die Beurteilung heranzuziehen. Einerseits kann die Mitschrift als Nachweis herangezogen werden, dass Studierende die Aufgaben eigenständig gelöst haben. Auf der anderen Seite ermöglichen diese Mitschriften die Bewertung von eingebetteten Theorieaufgaben sowie das Nachvollziehen von Fehlkonzepten bzw. kleinen Rechenfehlern. Das System unterstützt hier auch eine Nachbearbeitung der automatischen Bepunktung der Prüfung, um den

Anforderungen individuell nachkommen zu können.

Nun stellt sich die Frage, wie gut diese online Beispiele die erforderlichen Kompetenzen und Fertigkeiten der Vorlesung und Übung abbilden und abprüfen können. Grundsätzlich ist es wichtig, die online Beispiele so einzubinden, dass sie nicht nur eine freiwillige Zusatzleistung darstellen, sondern auch einen Mehrwert für Studierende im Laufe der Lehrveranstaltung mit sich bringen. Dies ist auch der Grund, warum die Beispiele in den mathematischen Kursen für Ingenieure, aber auch die MMT Beispiele in den Simulationslehrveranstaltungen, nicht nur einfach zur Verfügung gestellt werden, sondern sowohl in die Vorlesung als auch in Übungen und Prüfungen eingebettet bzw. eingesetzt werden.

Im Hinblick auf das „constructive alignment“ bezogen auf die Vorlesungsprüfung [4] wurden im Sommersemester 2018 einige Änderungen im Kursaufbau umgesetzt. Die kleineren Tests während des Semesters wurden von 30 auf 60 Minuten verlängert und dafür von 3 auf 2 pro Semester verringert. Diese Änderung wurde durchgeführt, um den Studierenden eine bessere Vorbereitung für die Vorlesungsprüfung zu geben. In diesen nun einstündigen Tests wurden ebenfalls theoretische Zwischenfragen gestellt. Wie auch in der Vorlesungsprüfung wurden die Mitschriften dieser Tests abgesammelt um eine nachträgliche Bepunktung der Theoriefragen durchführen zu können. Im Wesentlichen stellt nun die Vereinigung der beiden Zwischentests eine Art Vorlesungsprüfung dar.

4 Evaluierung

In diesem Abschnitt wird der Einsatz der in Kapitel 1 beschriebenen E-Learning Tools in Hinblick auf Kosten und Nutzen diskutiert. Weitere Punkte sind Feedback der Studierenden und die Nutzung der verfügbaren Angebote.

4.1 Problematik der Vergleichsgruppe

Bei der Einführung neuer Medien ist ein großes Thema immer die Vergleichsgruppe. Grundsätzlich ist es schwer überhaupt einen Vergleich anzustellen. Alleine die Tatsache, dass beide Gruppen voneinander wissen, bzw. vielleicht sogar in Lerngruppen oder ähnlichem interagieren ist es schwierig verlässliche vergleichenden Aussagen treffen zu können.

In den hier vorgestellten Lehrveranstaltungskonzepten gibt es in dem Sinne keine Vergleichsgruppen. Da die

mathematischen Lehrveranstaltungen für Studierende der Elektrotechnik abgehalten werden, könnte man Studierende anderer Ingenieursrichtungen als Vergleich heranziehen. Da deren Mathematiklehrveranstaltungen gänzlich anders aufgebaut sind und die Daten dieser Gruppen für die Forschungsgruppe nicht zugänglich sind, macht ein derartiger Vergleich jedoch wenig Sinn.

Allerdings wurde speziell das Konzept der online Vorlesungsprüfung anfangs, 2012-2015, auf freiwilliger Basis angeboten. Studierende konnten sich in dieser Phase entscheiden, ob sie die Prüfung lieber im klassischen Stil mit Stift und Papier oder aber am Computer im online System ablegen wollen.

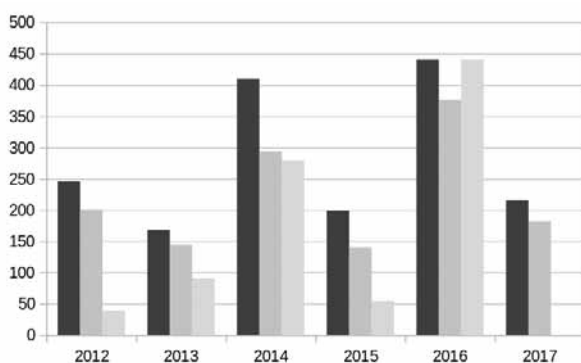


Abbildung 3: Diese Graphik stellt die Prüfungsantritte pro Jahr gegenüber - Absolutzahlen (dunkelgrün), ohne Studierende die die Prüfung zum wiederholten Male ablegen (hellgrün), online Prüfungen (gelb).

Abbildung 3 zeigt die Anzahl der Prüfungsantritte in den Jahren 2012 bis 2017. Hierbei sei erwähnt, dass an der Technischen Universität Wien pro Semester 3 Antritte pro Lehrveranstaltung angeboten werden müssen. Weiters ist man als Studierender nicht verpflichtet die Prüfung sofort abzulegen, es sei denn, diese ist Voraussetzung für ein anderes Fach. Dies ist aber nur selten der Fall. Daher haben Studierende völlig freie Wahl des Prüfungstermins und können in Summe fünfmal pro LVA antreten. An dem Histogramm kann man sehr gut ablesen, dass die Teilnahme an den online Prüfungen in den Jahren 2012 bis 2014 ansteigt. Dafür gibt es mehrere Gründe.

In den Anfängen wurden die Mitschriften in den online Prüfungen nicht abgesammelt, dafür wurde aber auch nur die Eingabe im online System bewertet. Dies hatte offensichtliche Nach- und Vorteile für Studierende. Multiple Choice Fragen konnten beantwortet werden ohne begründet zu werden. Andererseits wurden dafür

auch Ergebnisse aberkannt, die lediglich Syntaxfehler oder Vorzeichenfehler enthielten. Jeder Studierende konnte selbst abwägen, ob die online Prüfung oder die schriftliche Prüfung gewählt wurde. Die beiden Prüfungen waren von Anfang an ähnlich aufgebaut. Die abgefragten Themen, aber auch die Struktur der Beispiele waren die gleichen. Bei der online Prüfung hatte natürlich jeder Teilnehmer andere Zahlen bzw. Funktionen. In der schriftlichen Prüfung bearbeiteten alle dieselbe Aufgabenstellung.

Durch den Wechsel der Vortragenden jedes Jahr ist der Einfall der Teilnehmeranzahl in 2015 zu erklären. Der für die Vorlesungen in ungeraden Jahren verantwortliche Professor ist mit dem online System für die Vorlesungsprüfung nicht zufrieden. In den Übungen wird es zwar noch für die Tests eingesetzt, die Prüfungen laufen aber seit 2015 nur mehr schriftlich ab. Der Peak in 2016 kann damit erklärt werden, dass im Zyklus der geraden Jahre, seit 2016, online Prüfungen der einzige angebotene Modus ist. Um hier dem gewünschten Niveau nachzukommen, wurden die Beispiele umgestellt, Theoriefragen inkludiert und die Ausarbeitungen am Ende der Prüfung abgesammelt und für die Benotung herangezogen.

4.2 Zeitersparnis von online Prüfungen im Vergleich zu schriftlichen Tests

Ein oft diskutiertes Thema im Zusammenhang mit online Aufgaben und deren Einsatz in der Lehre ist die mögliche Zeitersparnis. In Bezug auf die Zusammenstellung der Prüfungen und Übungsaufgaben gibt es hier eine eindeutige Antwort. Das Zusammenstellen einer Prüfung mit neuen Beispielen dauert für online Prüfungen länger als am Papier. Hier muss man sicherstellen, dass die Randomisierung der Beispiele für alle Studierenden eine ähnliche Schwierigkeit darstellt und es auch zu keinen Fehlern kommt. Im Hinblick auf Übungsaufgaben ist es so, dass man in den ersten Jahren sehr viel Zeit in die Entwicklung von Beispielen stecken muss. Nach einer Einarbeitungsphase verkürzt sich aber die Entwicklungszeit für neue Beispiele drastisch. Ausschlaggebend ist in diesem Fall auch das Testen dieser Beispiele. Diese Beispiele werden von unterschiedlichen Leuten getestet, bevor sie in den Beispielpool aufgenommen werden können. All das benötigt eine gewisse Zeit.

Im Hinblick auf Zeitersparnis ist dann allerdings die Verbesserungszeit zu erwähnen. Falls man Mitschriften nicht einbeziehen will, reduziert sich diese Zeit fast zu

Null. In den oben beschriebenen Tests unter dem Semester hält sich daher die Verbesserungszeit in Grenzen. Selbst nach Einführung der Theoriefragen ist der Zeitfaktor überschaubar. Hierbei dauert es fast länger, zu definieren wofür es welchen Anteil an Punkte geben soll als dann das Verbessern an sich.

Im Falle der erwähnten online Prüfungen ist dies anders. Die Daten der letzten Jahre haben gezeigt, dass der gesamte Prozess, in diesem Fall umfasst dieser den Ausdruck der Prüfungsergebnisse, das Verbessern und Ausfüllen des zugehörigen Prüfungsbogens, im Durchschnitt 5-7 Minuten pro Prüfling in Anspruch nimmt. Das Zusammenstellen der Prüfung und Proberechnen benötigt, abhängig davon wie viele der Beispiele gänzlich oder aber teilweise neu programmiert werden müssen, 15-60 Stunden. Dies inkludiert die Zeit sich Prüfungsbeispiele zu überlegen, diese zu implementieren und Proberechnen zu lassen. Je mehr Prüfungen durchgeführt werden, steigt natürlich das Potenzial vorhandene Beispiele zu verwenden oder nur geringfügig abzuändern. Abbildung 4 veranschaulicht die Zeitersparnis bezogen auf die Zufriedenheit der Studierenden. Da Zufriedenheit eine nicht messbare Größe ist, kann man diese Graphik nur als eine schematische Darstellung der durch unzählige Feedbacks erhaltene Hypothese bezeichnen. Hierbei steht MM für maschinelle Bewertungen, BM für die Mischung aus automatischer und händischer Bewertung und HM für die reine händische Verbesserung. Die Zufriedenheit der Studierenden ist bei der gemischten Bewertung wesentlich größer obwohl sich der Zeitaufwand nicht maßgeblich erhöht.

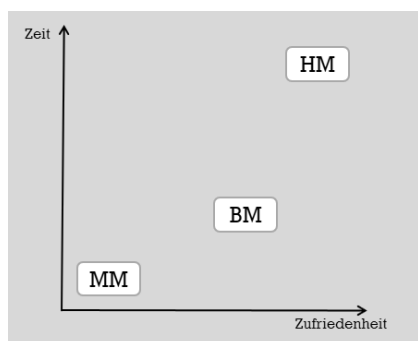


Abbildung 4: Diese Graphik zeigt den schematischen Zusammenhang der Zeitersparnis und der Zufriedenheit von Lehrenden und Studierenden. Hierbei steht MM für maschinelle Bewertungen, BM für die Mischung aus automatischer und händischer Bewertung und HM für die reine händische Verbesserung.

Alles in allem kann man sagen, dass sich die Verwendung so eines Systems dann auszahlt, wenn man für längere Zeit gleiche oder ähnliche Lehrveranstaltungen damit betreuen will.

4.3 Weiterentwicklung

Der Bedarf einer Weiterentwicklung beeinflusst ebenfalls die eben diskutierte Zeitersparnis. Bei der Verwendung eines eigens implementierten online Systems stellt sich diese Frage meist nicht. Mit Weiterentwicklung ist in diesem Fall nicht gemeint, mehrere Beispiele zu einem neuen Thema zu entwickeln, diese steht wohl außer Frage. In diesem Fall geht es darum, am System selbst, bzw. an der Funktionalität der Beispiele etwas zu verändern.

Im Falle des MMT Systems ist es so, dass das System vor Jahren auch auf eine neue Plattform versetzt wurde, um eine einfachere Handhabung der Beispiele für Lehrende zu ermöglichen. Im Sinne der Weiterentwicklung wurden weitere Softwareprodukte für die Verwendung in den Beispielen hinzugefügt.

Im Falle von Maple T.A. entwickelt sich das Produkt, da es sich um eine kommerzielle Software handelt, jedes Jahr. Dies ist natürlich vorteilhaft, da man so den wachsenden Ansprüchen der Studierenden nachkommt und auch die Handhabung für Lehrende verbessert wird. Die immer wachsenden Features einer kommerziellen Software stellen aber sowohl einen Vorteil als auch Nachteil dar. Bei einigen der Verbesserungen wäre es zielführend, Teile der eigenen Beispiele immer wieder zu überarbeiten, um die Neuigkeiten gleich in den alten Beispielen umsetzen zu können. Eine andere Möglichkeit ist natürlich, die bereits verfügbare Aufgabensammlung durch neue, das Feature verwendende, Beispiele zu ergänzen. Da der erste Einsatz dieser Software mehr als 10 Jahre zurückliegt und sich daher auch die Mitarbeiter, die für das System zuständig sind, geändert haben, ist es hier auch notwendig gewesen, die gesamte Aufbewahrungsstruktur wie auch die einzelne Beispielstruktur an sich nochmals zu überdenken. Dies führte zu einer neuen Art der Beispielablage und einem überarbeiteten Qualitätsmanagement.

Für eine eigens kreierte Plattform stellt sich dieses Problem unter Umständen erst gar nicht. Abhängig davon, wie viel Zeit in der Vorarbeit und Konzeption aufgewendet wurde, und natürlich aber auch der Stand der Technik beeinflussen diesen Faktor. Im besten Fall baut die Software auf dem letzten Stand der Technik auf und

beinhaltet somit alle zurzeit relevanten und gewünschten Features. Daher ist eine Weiterentwicklung nicht notwendig. Hier ist es dann, genau wie bei kommerziellen Lösungen, zeitintensiv die benötigten Inhalte zu erstellen. In Hinblick auf Weiterentwicklung ist es aber meist so, dass je mehr Zeit man mit einem neuen Medium bringt, desto mehr werden Lücken oder fehlende Features erkannt. Hier ist es ausschlaggebend, dass die Entwicklung der Plattform nicht in „privater“ Hand liegt, sondern institutionalisiert ist. Ansonsten könnte man an einen Punkt kommen, an dem die Plattform zwar noch verfügbar, aber nicht mehr erweiterbar ist, da die ursprünglichen Entwickler nicht mehr angestellt sind und es keinen gibt, der sich damit auskennt.

Zusammenfassend zeigt sich, dass im universitären Bereich die Verwendung von kommerzieller bzw. allgemein verfügbaren open-source Lösungen besser ist, als ein projektfinanziertes eigens entwickeltes System, da die Wahrscheinlichkeit hoch ist, dass die Entwicklung des Systems von einigen wenigen Personen abhängig ist.

5 Zusammenfassung

Man könnte nun die Frage stellen, ob online Tools die Zukunft der Lehre darstellen. Es steht wohl außer Frage, dass es in Zukunft eher mehr als weniger technische Hilfsmittel gibt, die uns in unserem Alltag unterstützen sollen. Niemand kann garantieren, dass die Technologie und ihre Produkte unsere Zukunft sind. Allerdings ist es, sehr wahrscheinlich, dass technische Hilfsmittel weiterhin eine wachsende Rolle in der Bildung einnehmen werden.

Die Auswertungen der oben beschriebenen Lehrveranstaltungen haben ergeben, dass in den Konzepten, bei

denen online Tools im Lehrkonzept integriert waren, die Studierenden den Systemen gegenüber sehr positiv gestimmt sind. Sie schätzen die Möglichkeit sehr viele Übungsbeispiele für die Prüfungsvorbereitung zur Verfügung zu haben. Auch die Visualisierung der Problemstellung bzw. der Modelle wird auf die Weise unterstützt. Den Studierenden ist aber auch klar, dass man ohne den nötigen Ehrgeiz und Zeitaufwand, ohne sich mit den verfügbaren Beispielen auf dem System, keine außergewöhnlichen Leistungen auf die Tests und Prüfungen erreichen lassen. Diese Fähigkeit zu Reflektieren lässt sich immer wieder in den Einsichtnahmen der Prüfungen oder aber in Abgabegesprächen von Projekten feststellen. Dieses direkte Feedback, aber auch die Evaluationen bestärken, die Konzepte diverser online Tools verstärkt in der Lehre einzusetzen, beizubehalten, weiter voranzutreiben und stetig zu verbessern.

References

- [1] U. Iberer. *Vom E-Learning zum Blended Learning: Aktuelle Entwicklung und didaktische Chancen virtueller Lehr- und Lernformen*. Theo-Web. Zeitschrift für Religionspädagogik 9, H.1, pp. 15-27, 2010.
- [2] O. Jandura, V. Karnowski. *Digital Natives vs. Digital Immigrants – fruchtbares empirisches Konzept für die Kommunikationswissenschaft oder populärwissenschaftliche Fiktion?* Publizistik, Vol. 60, Number 1, p 63, 2015.
- [3] <http://www.sussex.ac.uk/its/chrome/home>
- [4] T. Jungmann, *AssessING competences. Ansätze zur kompetenzorientierten Gestaltung von Prüfungen in den Ingenieurwissenschaften*. Reinhard Bork (Hrsg.), Prüfungsforschung, pp. 106 – 116, Schriften zur rechtswissenschaftlichen Didaktik, Bd. 6

Eignung kombinierter Simulation zur Darstellung energetischer Aspekte in der Produktionssimulation

Anna Carina Römer^{1*}, Martina Rückbrod¹, Steffen Straßburger¹

¹Fachgebiet Wirtschaftsinformatik für Industriebetriebe, Technische Universität Ilmenau, Postfach 100 565, 98684 Ilmenau, Germany; *anna-carina.roemer@tu-ilmenau.de

Abstract. In vielen produzierenden Unternehmen ist Energie ein wesentlicher Kostenfaktor. Energieaspekte werden deshalb in das Entscheidungssystem der Produktionsplanung und -steuerung einbezogen, um die Herstellungskosten zu senken. Die Simulation von Produktionsprozessen erfordert neben der Berücksichtigung technischer und logistischer Produktionsfaktoren auch die Integration von kontinuierlichen Energieverbräuchen. Da Fertigungssysteme im Allgemeinen in diskreten Simulationsmodellen beschrieben werden, könnte ein Ansatz, der die beiden Systemdynamiken kombiniert, vorteilhaft sein. Die kombinierte Simulation nutzt einen kontinuierlichen Simulationsansatz zur Abbildung des Energiebedarfs relevanter Produktionsprozesse und kombiniert diesen mit einem diskreten Simulationsansatz zur Abbildung von Material- und Logistikprozessen. Durch die Zusammenführung der Modelle können die Wechselwirkungen zwischen Materialfluss und Energieverbrauch in der Produktion realitätsnäher simuliert werden.

Einleitung

Ansätze zur kontinuierlichen Verbesserung der Produktionsprozesse mit dem Ziel einer hohen Produktivität bei höchster Produktqualität werden im Unternehmen bisher vor allem produktseitig umgesetzt. Konventionelle Planungsziele, wie etwa Zeit, Kosten, Verfügbarkeit und Qualität, haben einen hohen Stellenwert in der Produktionsplanung und -steuerung erlangt. Auch die Bedeutung der Energie- und Rohstoffmärkte hat in den letzten Jahrzehnten stetig zugenommen. Steigende Preise fossiler Energieträger sowie Debatten über den Klimawandel und endliche Ressourcen führen in der Industrie zu einem Umdenken. Die effiziente Nutzung des Faktors Energie und die konsequente Einbindung von Energieeffizienzzielen in betriebliche Abläufe sind für die Unternehmen zu einer Grundvoraussetzung geworden, um sich nachhaltig wettbewerbsfähig am Markt zu positionieren.

Da sich die Nutzung der Simulation als modellgestütztes Analysewerkzeug bei der Abbildung und Bewertung des dynamischen Verhaltens von Produktionssystemen in der Praxis etabliert hat, ist auch die Betrachtung von Fragestellungen der Energieeffizienz innerhalb der Simulation ein mittlerweile verbreiteter Untersuchungsansatz. Der Großteil existierender Arbeiten hierzu beschränkt sich jedoch zumeist auf die Berücksichtigung der Ressourcenverbräuche anhand wesentlicher messtechnisch erfasster Betriebszustände, die über einen definierten Zeitraum als konstant angesehen werden. Der Energieverbrauch der Ressourcen wird wie in [1] als statusbasiert angenommen und im Rahmen ereignisdiskreter Simulationsansätze (Discrete Event Simulation (DES)) dargestellt und analysiert. Unterschiede liegen in der Art der Implementierung der Energieflüsse, wobei nach [2] und wie in Abbildung 1 dargestellt drei Implementierungsparadigmen auf Basis rein ereignisdiskreter Simulation unterschieden werden können.

Hochdynamische Prozesse eines Produktionssystems, insbesondere im Bereich der energieintensiven Industrien, lassen sich mit solchen quasi-statischen Betriebszuständen nicht ausreichend realitätsnah modellieren. Die Einbeziehung kontinuierlicher Energie-Lastprofile kann mehr Transparenz schaffen. Eine geeignetere und präzisere Simulation energetischer Aspekte von Produktionsprozessen bietet nach [3] deshalb eine Kombination aus diskreter und kontinuierlicher Simulation.

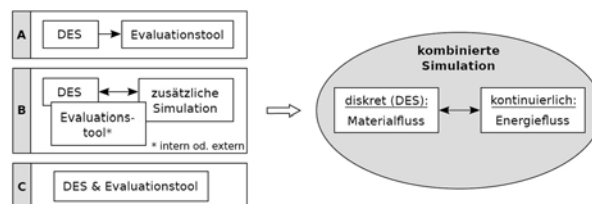


Abbildung 1: Übersicht über mögliche Implementierungsparadigmen zur Energieflusssimulation (in Anlehnung an [2]).

Die Untersuchung der Eignung der kombinierten Simulation zur Darstellung energetischer Aspekte in der Produktionssimulation ist Gegenstand der Masterarbeit [4], deren Ergebnisse in diesem Artikel zusammengefasst werden.

Die kombinierte Simulation ist keineswegs eine neue Methode; es sind Literaturstellen zu Untersuchungen und Praxisanwendungen von sogenannten hybriden oder diskret-kontinuierlichen Systemen bzw. kombinierter Modellierung über mehrere Jahrzehnte hinweg auffindbar, etwa in [5] und [6]. Dennoch hat sich diese Art von Simulation bisher nur geringfügig etabliert und auch der Begriff der hybriden oder kombinierten Simulation ist wie in [7] diskutiert nicht einheitlich definiert, sodass verschiedene Ansätze in der Literatur gefunden werden. Zum einen kann wie in [8] ein Multimethod-Modeling unterstützender Simulator gewählt werden, der die kombinierte Simulation per se durch adäquate Weltansichten und eine interne Verknüpfung ermöglicht, zum anderen besteht die Option, einen diskreten und einen kontinuierlichen Simulator wie in [3] über eine Schnittstelle zu koppeln oder wie in [9] diskrete und kontinuierliche Modelle in einer die hybride Simulation unterstützenden Softwarelösung zu implementieren.

In diesem Paper soll anhand eines Modells gezeigt werden, wie sich die kombinierte Simulation zur Darstellung der energetischen Aspekte in einer Produktionssimulation nutzen lässt und welche Vorteile sie in Abhängigkeit der Leistungscharakteristik der Maschinen im Vergleich zur rein ereignisdiskreten Simulation bringt.

1 Kombinierte Simulation zur Darstellung des Energieflusses

Nicht immer reicht eine rein diskrete oder eine rein kontinuierliche Simulation aus, um den Anforderungen zu simulierender Systeme gerecht zu werden. Nach [10] haben beide Systemdynamiken ihre Grenzen: die diskrete in der ganzheitlichen Betrachtungsweise sowie der dynamischen Abbildung von Systemen und die kontinuierliche Simulation auf dem Level der Details. Zudem weist die DES den Nachteil auf, dass die Komplexität der Modelle mit der Größe des zu simulierenden Systems exponentiell steigt. Die kontinuierliche Simulation positioniert sich auf makroskopischer Ebene und eignet sich durch ihren hohen Abstraktionsgrad nicht zur Abbildung einzelner Individuen oder Entitäten. Darüber hinaus ist

sie im Anwendungsbereich Produktion und Logistik wenig verbreitet. Sie ist deterministisch und enthält keine zufallsbedingten Einflussgrößen.

1.1 Ansätze für die kombinierte Simulation

Die kombinierte Simulation ist – wie der Terminus schon andeutet – eine Kombination aus den genannten zwei kontroversen Systemdynamiken; sie vereint DES mit der kontinuierlichen Simulation. In einigen literarischen Werken wird auch von Discrete Event System and Differential Equation System Specification (DEV&DESS) gesprochen. Dieser Ausdruck nimmt Bezug auf die theoretisch orientierten Formalismen nach [11] und stellt eine von zwei Varianten der kombinierten Simulation dar, wobei für die kombinierte Simulation eine sogenannte Condition Function verwendet wird, die für die Verbindung der diskreten und kontinuierlichen Variablen sorgt. Sobald eine Grenzwertüberschreitung stattfindet, wird diese Funktion aktiviert, wodurch ein bedingtes Ereignis erzeugt und der Systemzustand geändert wird.

Die konträre Variante wird Hybrid State Transition Machine (HSTM) genannt, und verfolgt einen praktischeren Ansatz. Die HSTM beruht auf Zustandsdiagrammen der Unified Modeling Language (UML). Diese bestehen aus Zuständen die durch Transitionen zusammenhängen. Nach [12] funktioniert die kombinierte Simulation so, dass Grenzwerte für die kontinuierlichen Variablen definiert werden und sobald ein solcher überschritten wird, ein diskretes, bedingtes Ereignis (State Event) erzeugt. Dadurch werden eine oder mehrere Transitionen getriggert, die eine Änderung von einem Zustand in den nächsten erlauben und auf diese Weise eine Zustandsänderung des Systems hervorrufen. In Ergänzung dazu kann es vorkommen, dass ein diskretes Ereignis während der Ausführung eine kontinuierliche Variable beeinflusst und dieser ungeachtet der zugrundeliegenden Funktionen neue Werte zuteilt.

1.2 Für die kombinierte Simulation geeignete Simulationssysteme

Simulatoren, die per se für die kombinierte Simulation geeignet sind, müssen in der Lage sein, sowohl eine diskrete als auch eine kontinuierliche Simulation durchzuführen und darüber hinaus beide Simulationsarten miteinander zu verknüpfen. Insbesondere die Verknüpfung trägt dazu bei, dass Wechselwirkungen zwischen den Modellsimulationen erkannt werden und sich die Simulationen gegenseitig beeinflussen können. Schließlich

können diskrete Variablen Veränderungen an kontinuierlichen Variablen hervorrufen und umgekehrt auch kontinuierliche Variablen diskrete Ereignisse erzeugen. Um diese Wechselwirkungen erkennen und ausführen zu können sowie den Zeitfortschritt der Simulation sicherzustellen, ist wie beispielsweise bei [10] ein sogenannter Solver (meist auf der kontinuierlichen Seite) erforderlich, der die Signale des DES-Systems empfängt, Differentialgleichungen löst und intervenieren kann. Ein häufig für die Simulation von Fertigungsbeispielen genutztes Simulationssystem ist Anylogic, das als Multimethod-Modeling-Software die kombinierte Simulation durch Zusammenführung mehrerer auf verschiedenen Weltansichten basierenden Modellen ermöglicht. Insbesondere im wissenschaftlichen Bereich als Multimethod-Modeling-Software eingesetzt und ursprünglich für die kontinuierliche Simulation vorgesehen ist die Software MATLAB/Simulink.

Als konträre Lösung für die kombinierte Simulation können auch mehrere verschiedenartige Simulatoren eingesetzt und beispielsweise über eine TCP/IP-Schnittstelle [3] oder über Standards wie HLA [13,14] miteinander gekoppelt werden. Voraussetzung dabei ist zum einen, dass mindestens ein Simulator DES und mindestens ein zweiter den kontinuierlichen Ansatz unterstützt, um durch Kopplung eine kombinierte Simulation zu erreichen. Zum anderen müssen beide Simulatoren für die Kopplung und die gegenseitige Beeinflussung ausgelegt sein. Es wird auch bei gekoppelten Simulatoren ein Solver benötigt, der den synchronen Zeitfortschritt zwischen den Simulatoren steuert und deren Wechselwirkungen realisiert.

2 Simulationsmodell

2.1 Modellbeschreibung

Um die Eignung der kombinierten Simulation anhand eines Simulationsvergleichs nachzuweisen, wird das nachfolgend beschriebene Modell parallel zur kombinierten Simulationsstudie auch einmal rein ereignisdiskret aufgebaut.

Das herangezogene Beispiel lehnt sich an die Ausarbeitungen von [15] an, der die Energie- und Ressourceneffizienz beim Aluminiumdruckguss untersucht hat. Es werden u. a. der Ablauf sowie die dafür relevanten Maschinen für eine industrielle Aluminiumdruckguss-Wertschöpfungskette beschrieben. Zusätzlich werden die

jeweiligen Energieverbräuche der Maschinen (gemessen mit einem „ChauvinArnaud 8335“) unter Berücksichtigung der Arbeitszyklen dargestellt.

Die Produktionslinie beginnt mit der Druckgusszelle und somit mit einem Warmhalteofen, der die bereits geschmolzenen und flüssigen Aluminium-Legierungen aus der Schmelze empfängt. Der Ofen erhitzt das Aluminium wieder auf 690 °C; währenddessen findet eine permanente Temperaturkontrolle statt. Dann geht der Ofen in den Warmhaltevorgang über. Bei Erreichen der Temperatur schaltet sich der Ofen aus, bei Unterschreitung hingegen beginnt der Heizvorgang erneut.

Zudem fungiert der Warmhalteofen als Puffer; nach dem Heizvorgang und während des Warmhaltens führt er das flüssige Aluminium in dosierten Portionen der Druckgussmaschine in die Druckgusskammer zu, die dann hieraus ein Gehäuse formt. Der entstandene Druckguss wird dann mithilfe einer Stanze aus der Form gelöst. An dieser Stelle wird gleichzeitig ein Abluftsystem betrieben, das die gelösten Materialreste absaugt. Anschließend erfolgt die Fertigstellung der Getriebegehäuse in einer Computerized Numerical Control (CNC)-Maschine, d. h. eine geometrische und funktionale Nachbearbeitung sowie Bohrungen für einsetzbare Schrauben. Eine Sandstrahlmaschine entgratet das Gehäuse und finalisiert es aus materieller Sicht, damit es zur Reinigung zur nachgelagerten Waschmaschine befördert werden kann. Diese besteht aus einem großen mit Waschflüssigkeit gefüllten Becken, durch das die Getriebe geschoben werden. Sie kann vier Gehäuse zeitgleich aufnehmen. Danach durchlaufen die Getriebe eine Dichtigkeitsprüfmaschine, die die defekten Gehäuse aussortiert.

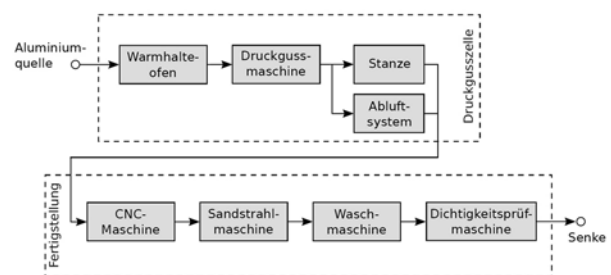


Abbildung 2: Flussdiagramm der Produktionslinie für Aluminium-Getriebegehäuse

Die diskrete Modellierung erfolgt in Anlehnung an das in **Abbildung 2** dargestellte Modell. Die Produktionslinie bildet den Prozess der einzelnen Entitäten ab, da-

bei handelt es sich zunächst um die Aluminium-Legierungen, später um die Getriebegehäuse. Betrachtet wird der Fluss der Entitäten durch das Modell, d. h. durch die einzelnen Maschinenstationen. Der dazugehörige Energieverbrauch wird im kombinierten Modell in System Dynamics (SD) dargestellt. Dabei fungiert ein Stock als Zähler für die Höhe des (Gesamt-)Energieverbrauchs (W_{ges}). Dieser Stock erfährt einen kontinuierlichen Zuwachs aus der Gesamtleistung (P_{ges}) über die Zeit (t) – gemäß der Gleichung (1).

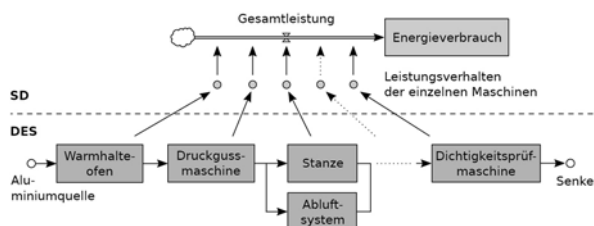


Abbildung 3: Einfluss von DES auf SD nach [15]

Das Ausmaß des Zuwachses/der Gesamtleistung hängt von den summierten Leistungen der einzelnen Maschinen (P_i) ab. Der Buchstabe n sei die Anzahl der Maschinen.

$$W_{ges} = P_{ges} * t; \quad P_{ges} = \sum_{i=1}^n P_i \quad (1)$$

Jede Maschine weist in Abhängigkeit von ihrem jeweiligen Bearbeitungszustand ein unterschiedliches Leistungsverhalten auf. Diese Leistungskurven der einzelnen Maschinen werden in SD modelliert (typischerweise in Form von Funktions- bzw. Differentialgleichungen) und beeinflussen den Fluss der Gesamtleistung; folglich haben die diskret modellierten Maschinen-Objekte jeweils Auswirkungen auf den kontinuierlich modellierten Energieverbrauch.

In der Praxis wird die Leistung oft mit entsprechenden Geräten ermittelt, insofern keine Funktionen für den Maschinentyp vorab vorliegen. Um einen möglichst realistischen Kurvenverlauf zu erhalten, sollte das Messintervall passend zur betrachteten Produktionseinheit und den ausgeführten Prozessen gewählt werden. Ist es zu groß, bleiben eventuell bestehende Leistungsspitzen oder -veränderungen unerkant. Dies würde die kontinuierliche Kurve verfälschen. Es ist jedoch nicht ratsam, immer so kleine Messintervalle wie möglich durchzuführen, da mit der Häufigkeit der Messung auch der Aufwand, ggf. Speicherplatzbedarf und möglicherweise damit verbundene Kosten steigen.

Da es in SD üblich ist, Funktionsgleichungen zu integrieren, wäre es nun angebracht, aus der Menge an Messwerten bzw. der daraus resultierenden Diagrammkurve eine Funktion aufzustellen, die diesen Verlauf mathematisch widerspiegelt und in die SD-Modellierung eingebettet werden kann. In diesem Beispiel soll jedoch auf eine explizite Funktion verzichtet werden und die Umsetzung im Simulator auf Basis der getakteten Messwerte erfolgen. Der Hauptgrund für den Verzicht liegt in der Komplexität des Musters der Leistungsmesswerte. Die Generierung einer akzeptablen Gleichung, die die Verbräuche realitätsnah abbildet, bringt einen erheblichen Aufwand mit sich, der in der industriellen Praxis ungerne betrieben wird. In **Abbildung 4** wird der Unterschied von diskreter und kontinuierlicher Abbildung der Leistungsverläufe einer beliebigen Maschine anhand eines Ausschnittes signifikanter Messwerte in einem kurzen Messintervall gegenübergestellt.

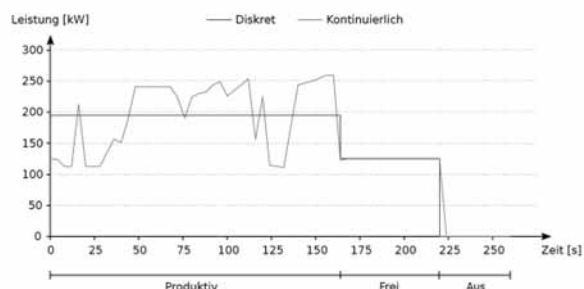


Abbildung 4: Kontinuierlicher Leistungsverlauf auf Basis von Messpunkten

Während in der diskreten Simulation genau ein Durchschnittswert über den gesamten Zeitraum eines Maschinenzustandes verwendet wird, nutzt die kombinierte Simulation eine Menge an Messwerten in festgelegter Auflösung über den Zeitraum eines Zustandes. Somit erfolgt die Abbildung energetischer Aspekte nicht in Abhängigkeit punktueller Zustandsänderungen, sondern zeitabhängig durch eine Tabellenfunktion. Darüber hinaus wird der kontinuierliche Charakter durch die simulative Umsetzung in SD bestätigt, da die Leistungswerte innerhalb SD mittels Parametern aufgezeichnet und so die Gesamtleistung abgebildet wird. Sie enthält somit weder Intervallzeiträume noch ist sie sprunghaft.

2.2 Modellimplementierung

Für die kombinierte Simulation wird ein Multimethod-

Modeling unterstützendes Simulationssystem herangezogen, um die Vorteile in nur einer alleinigen Applikation ohne Kopplung oder Verbindung mit anderen Simulatoren nutzen zu können. Zur Auswahl seien MATLAB/Simulink und AnyLogic gestellt; da MATLAB/Simulink zwar die kombinierte Simulation ermöglicht, jedoch für eine adäquate Nutzung von DES zusätzliche Module empfiehlt, erfolgt die Modellimplementierung in AnyLogic.

Die Modellierung der Leistungskurven einzelner Maschinen erfolgt typischerweise in Form von Funktions- bzw. Differentialgleichungen. Liegt keine Funktion für den Maschinentyp vor, lässt sich die Leistung oft mit entsprechenden Messgeräten erfassen und über eine Tabellenfunktion in einem realistischen Kurvenverlauf darstellen. Um die von den Maschinen benötigte elektrische Energie zu modellieren, greift AnyLogic auf die in Excel vorbereiteten Daten zu. Es werden zusätzlich Events benötigt, die von den diskreten Maschinenobjekten in Abhängigkeit von deren Bearbeitungszustand getriggert werden. Diese Ereignisse wiederholen sich im für die jeweilige Maschine erforderlichen Messintervall der gesammelten Daten, lesen den Leistungswert zum passenden Zeitpunkt aus und aktualisieren den Maschinenparameter. Gestoppt werden sie letztlich durch den Zustandswechsel der Maschinen bzw. durch das Triggern eines anderen Ereignisses zum Auslesen neuer Leistungswerte. Die Events sind die Verbindung zwischen der diskreten und der kontinuierlichen Modellierung: getriggert im DES-Modell beeinflussen sie die sich im SD-Modell befindlichen Maschinenparameter. In SD ergeben alle Maschinenparameter zusammen die Gesamtleistung zu jedem möglichen Zeitpunkt. Diese wird durch einen Fluss repräsentiert, der den Wert dieser Gesamtleistung kontinuierlich über die Laufzeit aufzeichnet und die Berechnung des Energieverbrauchs im Stock ermöglicht. Mit geeigneten Statistik-Objekten und Datasets werden diese Maschinenparameter analysiert und ergebnisorientiert ausgewertet.

3 Simulationsergebnisse

Die Output-Analyse der kombinierten Simulation erfolgt anhand ausgewählter Vergleichskriterien zur parallel durchgeführten rein ereignisdiskreten Simulationsstudie.

3.1 Datenbeschaffungsaufwand und Simulationslaufzeiten

Bei der Modellerstellung fällt auf, dass der Datenbeschaffungsaufwand für die kontinuierlichen Energie-Lastprofile durch die Bestimmung der idealen Messintervalle zur Rekonstruktion der realen Verbrauchsverläufe vergleichsweise hoch ist. Ebenso sind der Programmieraufwand sowie die Modellkomplexität bei der kombinierten Simulation durch eine zusätzliche Weltsicht deutlich höher. Hierdurch steigt – unabhängig von gekoppelten oder Multimethod-Modeling-Systemen – der Programmieraufwand.

Beim Blick auf die Laufzeiten der Simulation fällt auf, dass die kombinierte Simulation durchschnittlich dreimal länger benötigt als die diskrete Simulation des gleichen Modells. Die Ursache liegt darin, dass die kombinierte Simulation in ihrer Implementierung aufwendiger und umfangreicher ist sowie einen kontinuierlichen Teil in SD beinhaltet. Dies erhöht sowohl die Modellkomplexität als auch den Rechenaufwand bei der Ausführung.

3.2 Vergleich des Leistungsverlaufs

Aufgrund der über einen Maschinenzustand konstanten und gemittelten Leistungswerte weist der Leistungsverlauf bei einer rein diskreten Modellierung weniger stark ausgeprägte Extremwerte und auch eine weniger volatile Kurve auf (Abbildung 5). In kombinierter Simulation wird dieser Verlauf feiner modelliert. Hieraus lassen sich die Leistungsspitzen sowohl zeitlich als auch in ihrem Ausmaß genauer bestimmen. Diese Peaks sind evidenter und führen zu einer wahrheitsgetreueren Darstellung des Leistungsverlaufs.

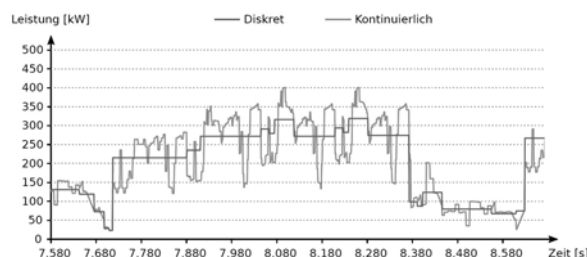


Abbildung 5: Vergleich des diskret und kontinuierlich dargestellten Leistungsverlaufs

Die Leistung ist, wenn sie beim Simulationszweck im Vordergrund steht, ein ausschlaggebendes Kriterium für den Einsatz der kombinierten Simulation. Ein passender

Anwendungsfall aus der Produktion ist der Fokus auf die Extremwerte und die Analyse der Maschineneinzelleistung/Gesamtleistung. Die kombinierte Simulation erlaubt an dieser Stelle die bessere Gestaltung der Produktionslinie mit dem Ziel, Peaks und besonders hohe Leistungsbeanspruchungen zu identifizieren.

Vergleicht man die durchschnittliche Leistung der Energieverläufe der diskreten und kombinierten Simulation, ähneln sich die Werte. Dies ist nicht weiter verwunderlich, da die diskreten, über einen Zustand konstanten Leistungswerte aus der Mittelwertbildung über die kontinuierlichen Leistungswerte resultieren.

Grundlegend ergibt sich aus den theoretischen Erkenntnissen, dass die kombinierte Simulation als wahrheitsgetreuer und vorteilhafter für die Abbildung von Energieaspekten gilt. Aus eigener Analyse ist jedoch zu konstatieren, dass diese Aussage nicht ohne Weiteres verallgemeinert werden kann; die bessere Eignung der kombinierten Simulation gegenüber einer rein diskreten Simulation fußt auf zwei entscheidenden Gesichtspunkten: dem Simulationszweck sowie der Ressourcenverfügbarkeit. Führt man eine Sensitivitätsanalyse durch und zerlegt das Simulationsmodell in zwei Varianten, einmal in die Maschinen mit relativ konstantem Leistungsverlauf und einmal in die Maschinen mit leistungsvariablem Verlauf, so wird deutlich, dass die kombinierte Simulation nicht in jedem Anwendungsfall die zu bevorzugenden Lösung darstellt.

3.3 Sensitivitätsanalyse

Der Leistungsverlauf des Abluftsystems, der Sandstrahlmaschine sowie der Waschmaschine weist jeweils nur geringe Abweichungen zwischen diskreter und kombinierter Simulation auf. Dies ist auf einen wenig volatilen Verlauf in der kombinierten Simulation zurückzuführen, der sich dem diskreten, über die Maschinenzustände konstanten Verlauf approximiert.

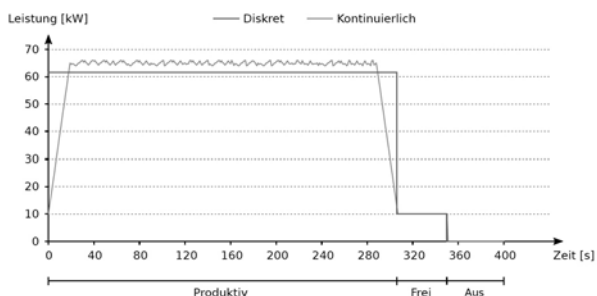


Abbildung 6: Energie-Lastprofil einer Waschmaschine

Diese Maschinen werden im Folgenden als leistungs-konstante Maschinen bezeichnet (Abbildung 6).

Die Druckgussmaschine, die Stanze, die CNC-Maschine sowie die Dichtigkeitsprüfmaschine werden als leistungsvariabel bezeichnet, da der Leistungsverlauf in kombinierter Simulation sehr volatil ist und somit erhebliche Abweichungen zum diskreten Leistungsverlauf bestehen (Abbildung 7).

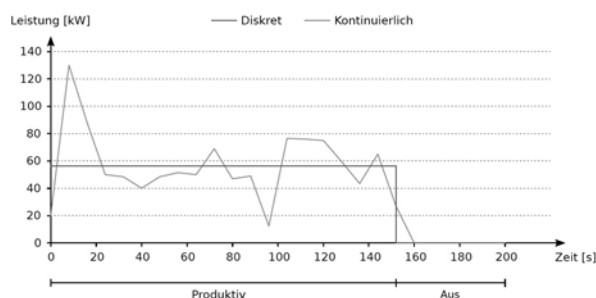


Abbildung 7: Energie-Lastprofil einer CNC-Maschine

In den nachfolgenden zwei Abbildungen ist der Kontrast der beiden Modellteile mit Blick auf die Gesamtleistung illustriert. Die Diagramme zeigen jeweils einen Ausschnitt über genau ein Muster, das sich im Simulationsdurchlauf periodisch wiederholt.

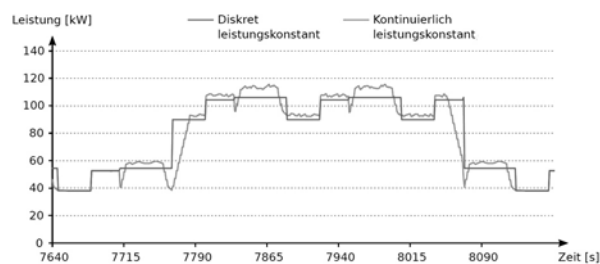


Abbildung 8: Vergleich des Leistungsverlaufs - Szenario leistungskonstante Maschinen

Die Abbildung 8 veranschaulicht – unter der Bedingung, dass sich die Produktionslinie aus nur leistungskonstanten Maschinen zusammensetzt – die Similarität zwischen der diskreten und der kombinierten Simulation. Trotz der offenkundig präziseren Leistungsdarstellung über die Zeit in der kombinierten Simulation, beschreiben beide Kurvenverläufe stets ein gemeinsames Leistungsniveau. Im Gegensatz dazu weichen die Kurvenverläufe in der Abbildung 9 bei einer Produktionslinie, die aus leistungsvariablen Maschinen besteht, signifikant voneinander ab und stellen ein kontrastreiches Muster dar.

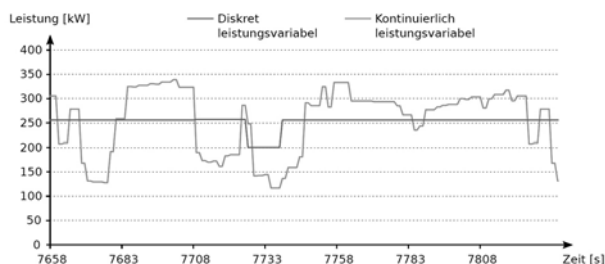


Abbildung 9: Vergleich des Leistungsverlaufs - Szenario leistungsvariable Maschinen

Die Entscheidung für oder gegen die kombinierte Simulation hängt zum großen Teil von der Maschinenzusammenstellung in der Produktionslinie ab. Besteht die Produktionslinie hauptsächlich aus Maschinen, die während des Betriebs eine verhältnismäßig konstante Leistung aufweisen, so ist eine kombinierte Simulation nicht zwingend erforderlich, da sich die Messwerte der energetischen Kennzahlen ähneln bzw. teilweise gleichen. Für eine äußerst detailgetreue Nachbildung der Produktionslinie und für eine gezielte Fokussierung des Leistungsverlaufs (bspw. bei der Analyse der Peaks) eignet sich die kombinierte Simulation dennoch besser, um Unterschätzungen zu vermeiden. Sind in der Produktionslinie Maschinen integriert, die eine variable Leistung aufweisen (auch neben leistungskonstanten Maschinen), ist dies ein Indiz für den Einsatz der kombinierten Simulation, um die reale Maschinencharakteristik adäquat nachstellen zu können.

3.4 Eignungsfeststellung der kombinierten Simulation

Der Verlauf der kumulierten Gesamtenergie ist in der kombinierten Simulation besser abbildbar, insofern sich leistungsvariable Maschinen in der Produktionslinie befinden. Jedoch wird auch hier im Ergebnis der gleiche Gesamtenergieverbrauch errechnet. Wesentlicher Mehrwert ist jedoch, dass sich Extremwerte, wie die minimale oder maximale Verbrauchsleistung, in der kombinierten Simulation exakter darstellen und bestimmen lassen. Handelt es sich lediglich um leistungskonstante Maschinen, kann auch hierfür ggf. schon die DES ein für den Zweck der Simulation hinreichend genaues Ergebnis liefern.

Die bessere Eignung der kombinierten Simulation hängt damit von drei Faktoren ab:

1. Ressourcenverfügbarkeit (zeitlich, personell sowie technisch im Hinblick auf Rechengeschwindigkeit,

Speicher, Energie- und Leistungsmessgeräte, Datenverfügbarkeit, u.Ä.)

2. Maschinenkonstellation in der Produktionslinie sowie der Leistungscharakteristik der Maschinen
3. Simulationszweck (Ziele und erstrebte Messwerte: Leistungs-/Peak-Analyse vs. Durchschnittswerte oder kumulierter Gesamtenergieverbrauch).

Unter der Annahme, dass in der heutigen Zeit sowohl Rechnerkapazitäten, als auch benötigte Messgeräte auf verschiedenste Art und Weise verfügbar sind und zudem oft eine Lösung für den personellen und zeitlichen Aufwand geschaffen werden kann, führt die Ressourcenverfügbarkeit alleine in der Regel nicht zu einer Entscheidung gegen den Einsatz der kombinierten Simulation. Des Weiteren besteht kaum eine Produktionslinie in der industriellen Praxis lediglich aus leistungskonstanten Maschinen. Aus dieser Perspektive kristallisiert sich im Endeffekt eine deutliche Tendenz zur besseren Eignung der kombinierten Simulation heraus, die lediglich vom Simulationszweck und den Zielsetzungen abhängt.

4 Zusammenfassung und Ausblick

Die Verknüpfung von Material- und Energieflusssimulation ist ein aktueller Forschungsgegenstand. In rein diskreter Simulation bedingt dies die Definition von zustandsbasierten und über einen Zeitraum konstanten Maschinen- sowie parallelen Energiezuständen. Diese Abbildung der Energieverbräuche ist jedoch häufig nicht angemessen exakt. Aus Gründen der präziseren Abbildung der Energie bei einer ihrer Charakteristik entsprechenden kontinuierlichen Modellierung ist die kombinierte Simulation für die Verknüpfung von Materialfluss in DES und Energiefluss in SD ein neuer Lösungsansatz. Die kombinierte Simulation umfasst sowohl einen diskret als auch einen kontinuierlich umgesetzten Modellteil; sie führt beide Systemdynamiken zusammen, sodass sich die Vorteile ergänzen. Es alternieren jeweils eine diskrete (mit Ereignisausführung und Zustandsänderung) und eine kontinuierliche Phase (Berechnungen und Zeitfortschritt). Auf diese Weise sind interdependente Wechselwirkungen untereinander möglich, die eine Kommunikation zwischen dem diskreten und dem kontinuierlichen Modell erlauben. Insbesondere in Einsatzfällen, wo in der Realität starke Wechselwirkungen zwischen dem diskret modellierten Teil des Produktionssystems und dem kontinuierlich abgebildeten Energieverbrauch existieren,

erscheint eine kombinierte Simulation daher zielführend zu sein. Weitere Forschungsaktivitäten sind jedoch notwendig, um aus den in der Praxis häufig erfassten Messwerten von Energieverbräuchen adäquate Funktionsgleichungen zur Modellierung in SD zu erstellen. Die Herausforderung besteht hier nicht nur im geeigneten Fitting der Gleichungen, sondern in der Modellierung der ggf. vorhandenen Wechselwirkungen zwischen Zustand des diskreten Modells und der gemessenen Energieverbräuche.

Zusätzliche Erweiterungen des konkreten Beispiels könnten in der Nutzung von zufallsbedingten Verteilungsfunktionen anstelle der festen Maschinenbearbeitungszeiten bestehen; so würde es möglich, Replikationen durchzuführen und auch stochastische Einflüsse in diesem Kontext zu untersuchen. Limitationen bestehen dann allerdings wieder in der Frage, wie hierbei mit den real vorliegenden Messwerten umzugehen ist, die sich auf eine deterministische Bearbeitungsdauer beziehen. Im vorliegenden Modell wurde lediglich die elektrische Energie betrachtet. Zur Erfüllung der Ganzheitlichkeit könnte die Betrachtung anderer Energieformen und peripherer energetischer Anforderungen (z. B. der Beleuchtung) hinzugezogen werden. Auch die maschinelle Erweiterung bzw. Modifizierung des hier verwendeten Modells oder die Wahl eines alternativen Simulators können ggf. weiterführende Erkenntnisse nach sich ziehen.

Die kombinierte Simulation zur Darstellung energetischer Aspekte stellt aktuell und voraussichtlich auch in den nächsten Jahren einen umfassenden Forschungsgegenstand dar, der zukünftig eine noch zentralere Rolle in der Simulation von Produktionssystemen einnehmen wird, da er für die industrielle Praxis höchst relevant ist.

References

- [1] Haag H. *Eine Methodik zur modellbasierten Planung und Bewertung der Energieeffizienz in der Produktion* [dissertation]. Universität Stuttgart; 2013.
- [2] Thiede S. *Energy Efficiency in Manufacturing Systems* [dissertation]. TU Braunschweig; 2012. doi: 10.1007/978-3-642-25914-2.
- [3] Peter T, Wenzel S. Simulationsgestützte Planung und Bewertung der Energieeffizienz für Produktionssysteme in der Automobilindustrie. In Rabe M, Clausen U, editors. *Simulation in Production and Logistics*; 2015 Sep. 16. *ASIM Fachtagung Simulation in Produktion und Logistik*; 2015; Dortmund. Stuttgart: Fraunhofer Verlag. pp. 535-544.
- [4] Rückbrod M. *Eignung kombinierter Simulation für die Ausbildung energetischer Aspekte in der Simulation von Produktionssystemen* [master thesis]. Technische Universität Ilmenau; 2018.
- [5] Pritsker A. A Panel On Combined Modeling. In Gantz D, Blais G, Solomon S, editors. *Proceedings of the 1985 Winter Simulation Conference; 17th Winter Simulation Conference*, 1985 Dec; San Francisco. pp. 111-112.
- [6] Schwarz P. Simulation of systems with dynamically varying model structure. *Mathematics and Computers in Simulation*. 2008; Vol. 79: pp. 850-863.
- [7] Mustafee N, Brailsford S, Djanatliev A, Eldabi T, Kunc M, Tolk A. Purpose and Benefits of Hybrid Simulation: Contributing to the Convergence of its Definition. In Chan W, D'Ambrogio A, Zacharewicz G, Mustafee N, Wainer G, Page E, editors. *Proceedings of the 2017 Winter Simulation Conference; 50th Winter Simulation Conference*, 2017 Dec; Las Vegas. Piscataway, NJ: IEEE. pp. 1631-1645.
- [8] Pawletta T, Schmidt A, Junglas P. A Multimodeling Approach for the Simulation of Energy Consumption in Manufacturing. *Simulation Notes Europe*. 2017; 2: pp. 115-124. doi: 10.11128/sne.27.tn.10377
- [9] Schlüter W, Henninger M, Buswell A, Schmidt J. Weak Point Analysis and Process Improvement of Non-ferrous Melting and Die-casting Plants Using Bidirectional Coupling of a Material Flow Model with an Energy Model. In: Wenzel S, Peter T editors. *Simulation in Produktion und Logistik 2017. 17. ASIM-Fachtagung*; 2017 Sep; Kassel. Kassel: kassel university press GmbH. pp. 19-28.
- [10] Helal M, Rabelo, L, Sepúlveda J, Jones A. A methodology for Integrating and Synchronizing the System Dynamics and Discrete Event Simulation Paradigms. In: Sterman J, editor. *Proceedings of the 25th International Conference of the System Dynamics Society. 25th International Conference of the Systems Dynamics Society*; 2007 July; Boston. pp. 1-24.
- [11] Zeigler BP, Kim TG, Prähofer H. *Theory of Modelling and Simulation*. San Diego: Acad. Press, 2nd ed., 2000, 510 p.
- [12] Maler O, Manna Z, Pnueli, A. From Timed To Hybrid Systems. In: de Bakker JW, Huizing C, de Roever WP, Rozenberg G., editors: *Proceedings of the Real-Time: Theory in Practice. REX Workshop*; 1991 June; Mook. London: Springer-Verlag. pp. 447-484.
- [13] Lantzsich G, Straßburger S, Urban C *HLA-basierte Kopplung der Simulationssysteme Simplex III und SLX*. In: *Proceedings Simulation und Visualisierung '99*, eds. O.Deussen, V. Hinz, P. Lorenz. Magdeburg, March 4.-5. 1999, SCS International, pp. 153-166.
- [14] Straßburger S, Schulze T, Lantzsich G. *Simplex 3 und SLX-gemeinsam unter HLA*. *Proceedings of the 13th Simulation Symposium ASIM 99*. September 21-24, 1999. Weimar, Germany.
- [15] Heinemann T. *Energy and Resource Efficiency in Aluminium Die Casting*. Heidelberg: Springer Verlag; 2016. p. XXVI, 242.

Hybrid modeling approach for prediction of energy demand and power peaks in intralogistic systems

Armin Siegel^{1*}, Karsten Turek¹, Elisa Michelini², Thorsten Schmidt¹

¹Chair of Material Handling, TU Dresden, 01062 Dresden, Germany; *armin.siegel@tu-dresden.de

²Department of Industrial Engineering, Bologna University, Viale Risorgimento 2, 40136 Bologna, Italy

Abstract. For economic and ecological reasons, the demand for energy efficiency in material flow systems continues to increase. In addition to minimizing energy consumption, reducing power peaks is a second objective. This article presents an approach which integrates an energy and power demand model into a discrete-event material flow simulation. A hybrid modeling approach is demonstrated which is built with the material flow simulator *AutoMod*[®]. The mathematical environment *MATLAB*[®] is used for continuous modeling. As proof of concept for predicting electrical power requirements an automated storage and retrieval system is employed. The results of the prediction are analyzed and an outlook for a model based predictive control system is given.

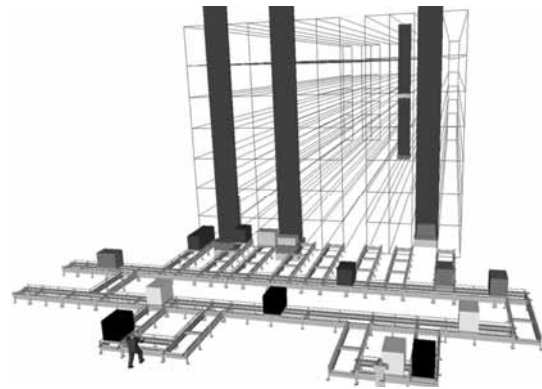


Figure 1: Simulation model of an automated storage and retrieval system (AS/RS)

Introduction

Material flow systems enable the flow of goods within companies. They include subsystems with transport, picking, and storage processes. Discrete-event simulation plays an important role in the prediction of system performance [1, 2, 3]. As an example, Figure 1 shows a simulation model of an automated storage and retrieval system (AS/RS) with two order picking stations.

The increasing focus on energy aspects in material flow processes requires more accurate energy prediction models. Energy efficiency aims to minimize transport energy consumption while maintaining the main throughput figures. In addition, minimizing short-term power peaks is important as well. Particularly in the context of the increasing use of renewable energy, the aim should be to keep grid load as constant as possible. New methods of Peak Shaving and Load Shifting are gaining in importance (see [4, 5]). The electrical installation (the connection point, power cable cross-sections

and transformers) is designed on the basis of the expected maximum power peak. In Industry, long term energy demand and short term maximum power peaks influence energy billing [6, 7, 5]. Therefore, the need to evaluate both arises already in the planning phase. The appropriate energy demand modeling and prediction are crucial parts in this context [8, 9, 10, 11, 12, 13].

The article presents an approach to link a continuous energy demand modeled calculation to a discrete-event simulation of material flow systems. It starts with an introduction to the evaluation of material flow systems followed by a discussion of typical energy demand and power peak calculation models. Thereafter, concepts for combining material flow and power demand calculation are reviewed. Afterward, hybrid modeling approach is presented. An example of an AS/RS hybrid model as a proof of concept is applied. The functionality of the approach and results are presented. The article concludes with a summary and an outlook.

1 Simulation of material flow systems

Material flow systems are often evaluated with discrete event simulation [1, 3]. Until now, the primary focus was on performance figures such as goods throughput, delivery times, resource utilization, buffer size, or required number of storage bins. Additional evaluation about energy consumption is increasing but limited by linking constant demand to the basic system states such as "waiting", "switched on", or "paused". A lack of incorporated precise models with detailed power curve projections into material flow simulation can be stated. In addition, current works concentrate primarily on production machines [14, 15, 16, 17].

There are two basic approaches to supplement simulation with energy figures. On the one hand, an offline calculation is carried out after the simulation run that computes the used energy on the basis of the recorded processes and states. As a second option, the energy figures are online determined in the simulation run by an immediate calculation of the power demand for every transport move [11].

2 Energy demand and power peak modeling

To evaluate energy efficiency for material flow systems, various energy demand models have been developed. Of particular note here are models for storage and retrieval machines (SRMs) used in automated warehouses with different levels of detail [9, 10, 18, 19, 20]. A common approach is based on the abstraction of the movement process of the SRM (referred as motion trajectories, see Figure 2 below) in a continuous oriented modeling environment. The motion kinematics and the physical device parameters are applied in a physical parameter model determining the curve of active power consumption. The total energy consumption can be calculated by integrating these power consumption curves and adding portions of mechanical and electrical conversion losses. The model used in the article follows this approach. Figure 2 above shows the power consumption curve as a comparison between modeled and measured values.

The power demand model for a motion trajectory of an SRM was developed by the authors in earlier projects and validated by several real system measurements [20]. A seven phase motion model is used to

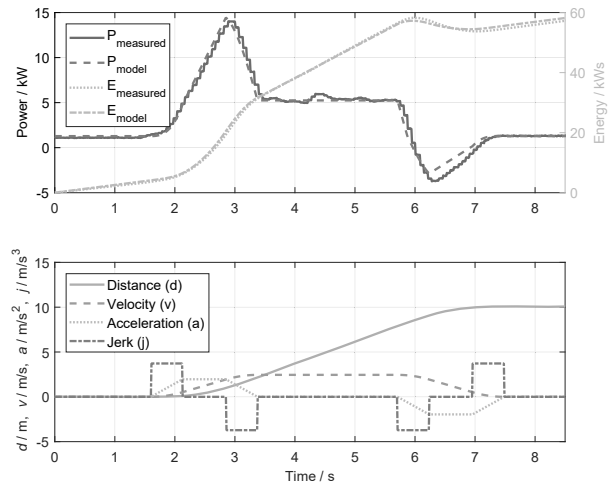


Figure 2: Energy and power demand of an SRM chassis system (top) and motion trajectory (below)

model the trajectory as shown in Figure 2 below section. A movement begins with the acceleration process (Figure 2 from second 1.6 to 3.3). There the jerk is limited to a maximum value. For this reason, the acceleration is subdivided into three phases. This part is followed by the movement with constant velocity. The motion is completed with the braking process (see Figure 2 from second 5.8 to 7.5). Like the acceleration, the braking part consists of three phases due to the jerk limitation. The traveled distance $s(t)$, the velocity $v(t)$, the acceleration $a(t)$ and the jerk $j(t)$ of the machine are linked to each other by their time derivatives.

$$j(t) = \frac{\partial a(t)}{\partial t} = \frac{\partial^2 v(t)}{\partial t^2} = \frac{\partial^3 s(t)}{\partial t^3} \quad (1)$$

Together with the start and boundary conditions at the phase edges, the complete motion trajectory can be determined. The required mechanical power results with the following formula.

$$P(t) = F(t) \cdot v(t) \quad (2)$$

$v(t)$ is the velocity determined by the movement trajectory. $F(t)$ is the force required for the movement consisting of two components $F_{\text{acc}}(t) + F_{\text{dr}}(t)$. The first part results from acceleration $a(t)$ of mass m (vehicle and load).

$$F_{\text{acc}}(t) = m \cdot a(t) \quad (3)$$

The second part is the wheel driving resistance, which

is calculated with Scheffler's formula [21].

$$F_{dr}(t) = m \cdot g \left[\frac{2}{D} \cdot \left(\mu_B \cdot \frac{d}{2} + f \right) \right] \cdot \text{sign}(v(t)) \quad (4)$$

The following values are considered in the formula:

- g ... acceleration due to gravity
- D ... wheel diameter
- d ... hub diameter
- f ... lever arm of rolling friction
- μ_B ... bearing friction value

The sign of the velocity ($\text{sign}(v(t))$) is used to distinguish the operating direction for forward and reverse movement. In the deceleration phase, power $P(t)$ can assume negative values depending on the parameters in the Formulas 3 and 4 due to the negative acceleration $a(t)$. In addition to the motion power $P(t)$, the power dissipation of the drive train is calculated by its efficiency values.

$$P_{diss}(t) = |P(t)| \cdot (1 - \mu_{mot} \cdot \mu_{gb} \cdot \mu_{pt}) \quad (5)$$

The loss is taken into account by the efficiency of the drive μ_{mot} , gearbox μ_{gb} and power transmission μ_{pt} .

Power loss at the frequency inverter is calculated by a proportional relationship between the motor input power $P(t)$ and the associated power loss. The maximum power dissipation P_{maxL} and maximum output power P_{maxO} can be taken from the technical specification of the inverter. In addition, the basic power loss of the inverter P_{bli} is taken into account. The following equation is used to calculate the current power dissipation $P_{dissI}(t)$:

$$P_{dissI}(t) = \frac{|P(t)| \cdot P_{maxL}}{P_{maxO}} + P_{bli} \quad (6)$$

The power requirement for horizontal movement $P_{hor}(t)$ is determined by the sum of Equations 2, 5 and 6.

$$P_{hor}(t) = (F_{acc}(t) + F_{dr}(t)) \cdot v(t) + P_{diss}(t) + P_{dissI}(t) \quad (7)$$

When lifting and lowering the load with the hoist unit (index v for vertical movement), the earth's gravitational force must also be taken into account in addition to the acceleration resistance. The mechanical power for upwards defined movement is:

$$P_v(t) = m_v \cdot (a(t) + g) \cdot v(t) \quad (8)$$

In the formula, m_v is the proportion of the hoist unit that is lifted and the load mass. The power loss of the drive train for lifting is calculated similarly using assumed efficiencies (see Equation 5). These values must be adjusted depending on the actual SRM. The power loss at the frequency inverter P_{dissIv} is determined in the same way as the horizontal movement. The timeline of the total machine power results from the superposition of horizontal and vertical power.

An additional constant basic loss in the system results from the energy requirement for the ready-to-operate state of the system and the power supply. The value is determined at the idle state of the SRM and added to the model as a constant value.

3 Energy demand model integration

Three concepts are possible for modeling an online energy demand prediction in a material flow simulation. The first concept is the calculation of the energy demand within the framework of the discrete-event material flow simulator [13]. Secondly, a continuous oriented software environment can be expanded to a material flow simulation [8, 9]. The third possibility is the combination of both modeling types to a new, hybrid system [14]. For this purpose, it is necessary to check the interfaces which are provided by the software systems and consider the boundary conditions which have to be fulfilled.

The combining approach provides the advantage that both parts make use of the strengths in their modeling domain. The enhancement of an existing material flow simulation in this way needs less effort in comparison with the other concepts. As conclusion, the hybrid modeling approach was used in a proof of concept. It is the starting point for a model-predictive control concept which basic structure is presented in the following section.

4 Basic concept for predictive energy control

In a material flow system the operative control directs the physical transport operations based on a given batch of transport orders. Figure 3 upper section shows an example of an AS/RS control scheme with shelf storage orders for two parallel machines. Each of these SRM

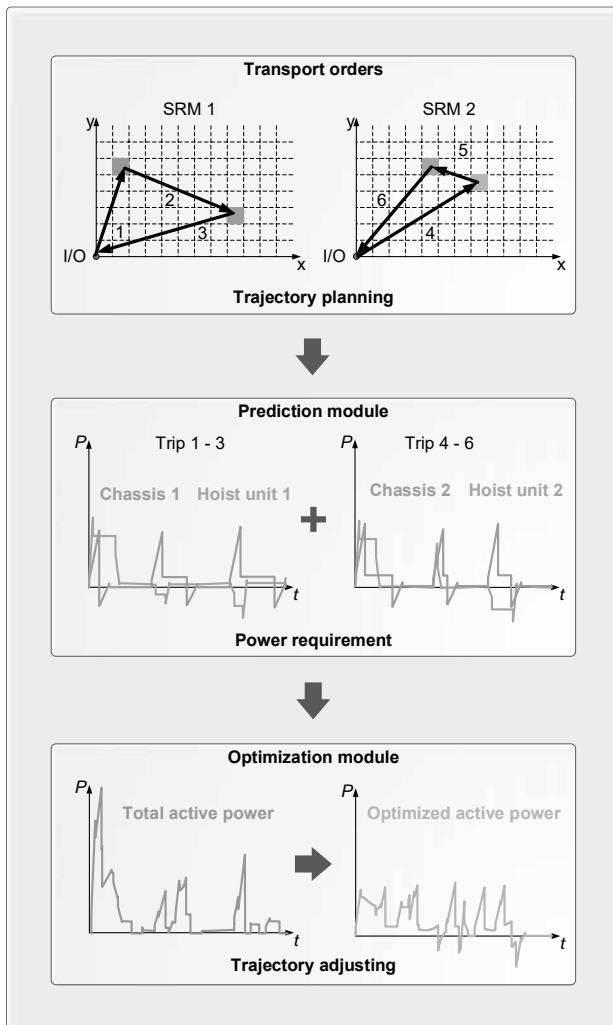


Figure 3: Concept for predictive energy control

received a double-cycle transport order. One pallet is put in and one pallet is removed from storage within one movement control cycle. Usually, the SRMs work independently and carry out immediately the movement with maximum kinematic parameters (velocity, acceleration) simultaneous for the chassis and hoist unit. This operation mode results in high energy consumption and power peaks.

The concept of predictive energy control requires additional steps for prediction and optimization of power and energy demand before transport execution. Figure 3 illustrates the basic concept which enables the integration of energy storage systems and the consideration of volatile offers of renewable energy.

The order data are transferred to the prediction mod-

ule and the expected active power requirement for chassis and hoist unit is calculated (Figure 3 middle). The resulting total active power consumption is shown in the graph at the lower left.

In a second step, the expected energy consumption and power peaks can be reduced by modifying the transport trips through optimization. For example, shifting the starting times of the devices adjusts the trajectories of the chassis and hoist drive (see Figure 3 at lower right) for lower power peaks. The modified trip parameters are returned to the operational control system. Finally the transport is carried out with lower power peaks and less energy consumption.

In this article we introduce the approach for the necessary online power prediction. In the following, the utilized software components and interfaces are presented before the first results of the interactive modules are compared. The optimization module is presented in a first approach to demonstrate its integration. Module details are subject to further work.

5 Hybrid modeling

5.1 Software systems introduction

The simulator *AutoMod*[®] is used for the discrete-event material flow simulation. *AutoMod*[®] is a 3D industrial simulation software. The user can describe accurately a variety of material flow systems in production and logistics with realistic imaging. The tool is used for modeling large systems like airports, semiconductor factories, postal, warehousing and others. For this article a small model of an AS/RS is used as seen in Figure 1.

The *MATLAB*[®] Software is a multi-paradigm numerical computing environment developed by MathWorks. It has been used to model the mathematical equations that calculate the motion trajectories and the power consumption timelines of the AS/RS. The software choice is mainly based on the fact that the environment has a large database of built-in algorithms and functions for fast data processing.

These two software tools are used in our work for a hybrid modeling approach (see [22, 23]). The implementation details are described in the next section.

5.2 Connecting of AutoMod and MATLAB

There are different ways to link *AutoMod*[®] with other programs. The variants presented in literature name the use of DDE (Dynamic Data Exchange), OPC

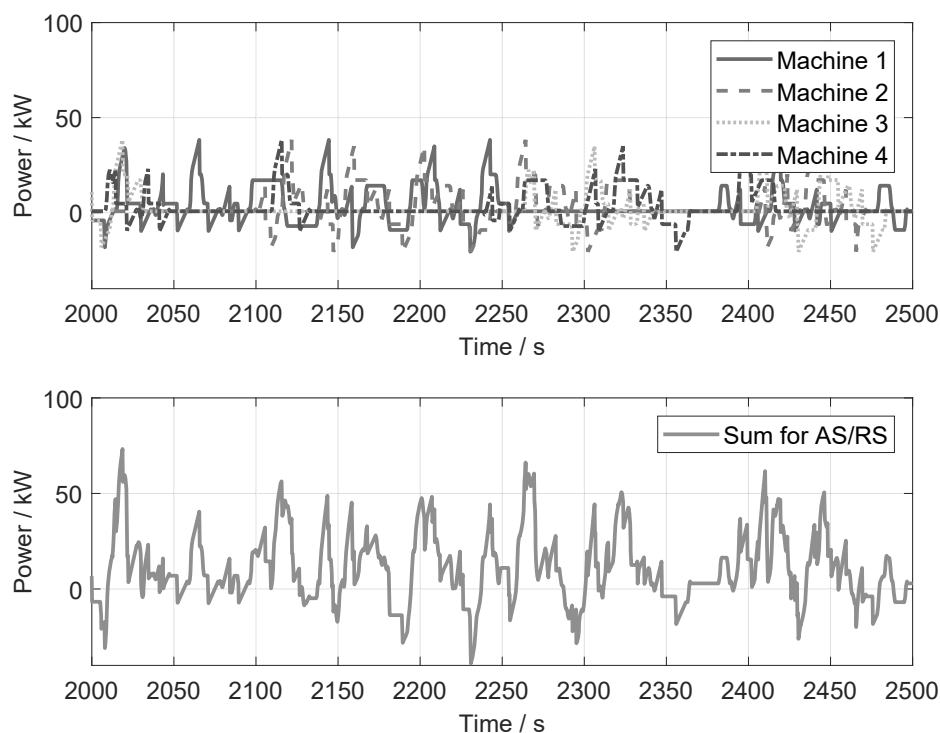


Figure 4: Power consumption of SRM's (top) and AS/RS (below)

(Open Platform Communications), TCP/IP (Transmission Control Protocol/Internet Protocol) socket, and a DLL (Dynamic Link Library) (see [24, 25]). The DLL integrates an encapsulated software code into another program. For reasons of data transfer rate we selected the DLL approach as linking option. The concept of model modularization and integration of complex algorithms via DLL in AutoMod was presented by the authors already in [3].

The *MATLAB*[®] environment offers the possibility to compile developed functions to a DLL using the Matlab Coder. The restriction of this method, that not all build-in functions of *MATLAB*[®] are supported, is not limiting in the selected model context. A list of all supported commands can be found in [26].

The simulation model in *AutoMod*[®] requires additional header files and the declaration of the utilized DLL functions for integrating. At runtime the simulation model calls the functions implemented in the DLL and gathers the calculation results for further control decisions. In contrast to a co-simulation (see [27, 28]) this approach is an integration of *MATLAB*[®] in *AutoMod*[®].

5.3 Power demand calculation example

By these means, the power demand calculation is realized online. At model initialization the technical parameters of the SRM devices are supplied to the DLL. During simulation the start and finish coordinates of the transport within the rack, the device number, and the current time are transmitted. The motion trajectory and the active power demand timeline is calculated by the DLL.

Figure 4 shows a sample section of the calculated timeline. Above is the active power calculation for the SRMs (refer to Figure 1) shown. Below is the overall total active power consumption of all four machines presented. The maximum power consumption of one machine is approximately 37 kW. The accumulated power of all machines reaches about 70 kW in this example.

Demonstrating the proof of concept for hybrid modeling simulation, experiments are carried out by considering a maximum total peak power of further SRM movement. If the given sum limit of 50 kW is exceeded, the AS/RS transport control can shift the start time of the SRM motion by a few seconds in order to

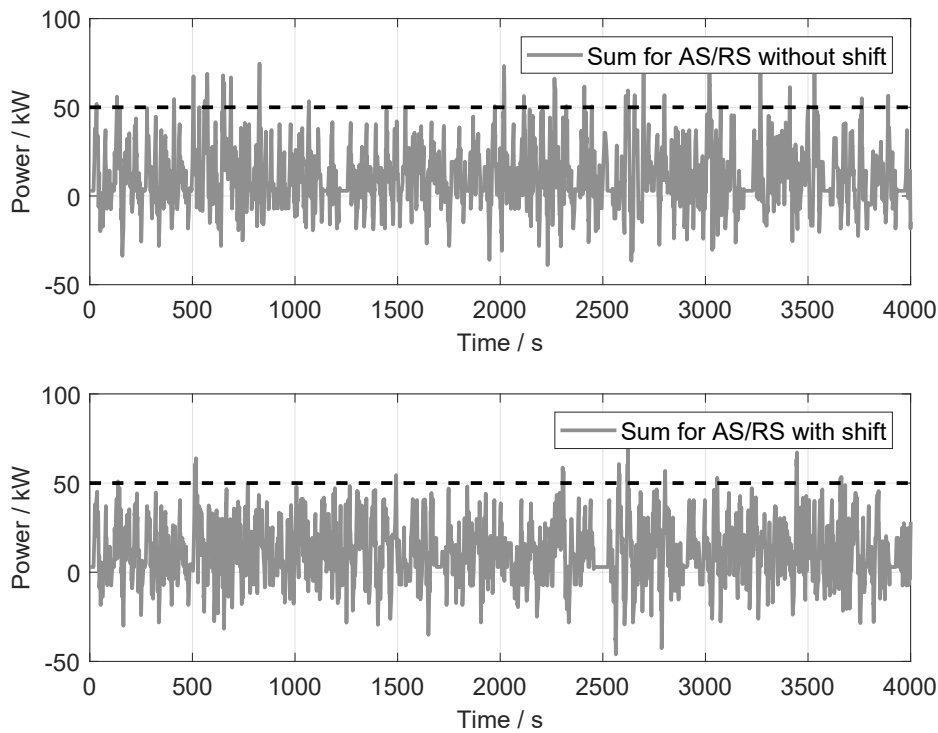


Figure 5: Comparison of conventional operation (top) and the motion shift (below).

avoid power peaks overlapping. The DLL calculates the power timeline for motion in advance and sums it with the previously calculated total power of all earlier movements. If a new peak above the power limit arises, the start time of the new motion is shifted second-by-second within the shifting range (5 s). The shift time value with the lowest power peak is returned to the material flow simulation and applied. The result is that the SRM transport start is delayed by the transport control (see [8]). The Figure 5 compares experiments with shifting control and without. The results show the basic suitability of peak power limiting within a material flow control. However, in this basic experiment a complete guarantee of maximum power limitation to a certain value can't be secured and needs further control measures and adaptation.

6 Conclusion and outlook

The article describes the enhancement of discrete-event material flow simulation by a continuous calculation model of electrical power timelines for transport moves in order to evaluate power peaks and energy demand.

The material flow model of an AS/RS was implemented in the standard discrete-event simulator *AutoMod*[®]. The power calculation model was separately developed in the *MATLAB*[®] environment and afterward compiled into a DLL. The calculation model evaluates the technical parameters of the AS/RS devices with the transport order data and predicts the detailed power demand timeline for each transport. The integration of the DLL into the material flow simulation creates a hybrid model for a combined evaluation of the material flow with the corresponding power peaks and energy demands in intralogistic systems.

In the next steps, we will focus on the energy efficient control of material flow systems. The derived knowledge and developed tools for the prediction of power and energy demand builds the basis for developing a predictive energy control. The coordination of the individual movements of the devices is a promising step for increasing energy efficiency and reducing power peaks. The modification of the start times and the kinematic parameters of the device trajectories provides an appropriate way of influencing these key figures [29]. As a result predictive energy control can con-

tribute to power peak shaving and load shifting techniques for material flow systems. Subsequently in further research, we will investigate the impact on smoothing power consumption from the grid by integrating an energy storage system in the model.

Acknowledgement

The IGF project 19324 BR/1 of the Research Foundation Intralogistics / Material Handling and Logistics (FG IFL) Lyoner Strasse 18, 60528 Frankfurt am Main is funded through the AiF under the program of Industrial Collective Research for SMEs (IGF) by the Federal Ministry for Economics and Technology, following a decision of the German Bundestag.



References

- [1] Manivannan MS. Simulation of Logistics and Transportation Systems. In: *Handbook of Simulation*, edited by Banks J, pp. 571–604. Hoboken, NJ, USA: John Wiley & Sons, Inc. 1998;.
- [2] Carson J. Introduction to modeling and simulation. *IEEE Engineering Management Review*. 2009; 37(2):13–13.
- [3] Siegel A, Turek K, Schmidt T. Modular, complexity reduced Material Handling Simulation Model for Air Cargo Terminals. In: *The 2016 European Simulation and Modelling Conference*, vol. 30. Las Palmas, Spanien: EUROSIS-ETI Publication. 2016; pp. 467–472.
- [4] Uddin M, Romlie MF, Abdullah MF, Abd Halim S, Abu Bakar AH, Chia Kwang T. A review on peak load shaving strategies. *Renewable and Sustainable Energy Reviews*. 2018;82:3323–3332.
- [5] Chua KH, Lim YS, Morris S. Energy storage system for peak shaving. *International Journal of Energy Sector Management*. 2016;10(1):3–18.
- [6] Federal Republic of Germany. Regulation on charges for access to electricity supply networks. 2017.
- [7] Autorità di Regolazione per Energia Reti e Ambiente (ARERA). Modifiche alla Bolletta 2.0 e al Codice di condotta commerciale. 2017.
- [8] Hahn-Woernle P, Günthner WA. Power-load management reduces energy-dependent costs of multi-aisle mini-load automated storage and retrieval systems. *International Journal of Production Research*. 2017;1(1):1–17.
- [9] Ertl RF. Energiebedarfsermittlung und Energieeffizienzbewertung von Regalbediengeräten in automatischen Kleinteilelagern. phdthesis, TU München, München. 2016.
- [10] Braun MSA. Entwicklung, Analyse und Evaluation von Modellen zur Ermittlung des Energiebedarfs von Regalbediengeräten. phdthesis, KIT, Karlsruhe. 2016.
- [11] Peter T, Wenzel S. Simulationsgestützte Planung und Bewertung der Energieeffizienz für Produktionssysteme in der Automobilindustrie. In: *Simulation in Production and Logistics 2015*, edited by Gesellschaft für Informatik, vol. 16 of *ASIM-Mitteilung*. Stuttgart: Fraunhofer Verl. 2015; pp. 535–544.
- [12] Sommer T. Entwicklung und Bewertung von Lagerstrategien zur Steigerung der Energieeffizienz in automatischen Hochregallagern unter Beachtung des Umschlags. phdthesis, Universität Stuttgart, Stuttgart. 2015.
- [13] Turek K. Simulationsbasierte Entwicklung energieeffizienter Steuerungsstrategien für fördertechnische Systeme. *final report 17014 BR*, TU Dresden, Dresden. 2013.
- [14] Peter T, Wenzel S, Reiche L, Fehlbier M. Coupled simulation of energy and material flow — A use case in an aluminum foundry. In: *Winter Simulation Conference (WSC)*. IEEE. 2017; pp. 3792–3803.
- [15] Wenzel S, Junge M, Pöge C, Spieckermann S. Energieeffizienz in der Automobilindustrie - Ganzheitliche simulationsgestützte Planung und Bewertung. *Productivity Management*. 2013;18:21–24.
- [16] Law AM, Kelton WD. *Simulation modeling and analysis*. McGraw-Hill series in industrial engineering and management science. Boston: McGraw-Hill, 3rd ed. 2000.
- [17] Banks J, Carson JS, Nelson BL. *Discrete-event system simulation*. Prentice-Hall international series in industrial and systems engineering. Upper Saddle River, NJ: Prentice Hall, 2nd ed. 1999.
- [18] Meneghetti A, Dal Borgo E, Monti L. Rack shape and energy efficient operations in automated storage and retrieval systems. *International Journal of Production Research*. 2015;53(23):7090–7103.
- [19] Meneghetti A, Monti L. Multiple-weight unit load storage assignment strategies for energy-efficient

- automated warehouses. *International Journal of Logistics Research and Applications*. 2014; 17(4):304–322.
- [20] Siegel A, Turek K, Schmidt T, Schulz R, Zadek H. Modeling the energy need of storage and retrieval machines and different storage operating strategies for the reduction of the energy need. *Logistics Journal* 2013. 2013;10(1):1–18.
- [21] Scheffler M. *Grundlagen der Fördertechnik - Elemente, Triebwerke*. Braunschweig, Wiesbaden: Vieweg. 1994.
- [22] Lunze J, Nixdorf B, Richter H. Hybrid modelling of continuous-variable systems with application to supervisory control. In: *1997 ECC*. Brussels: IEEE. 1997; pp. 1376–1381.
- [23] Fritzson P, Bunuş P. Modelica - a general object-oriented language for continuous and discrete-event system modeling and simulation. In: *35th Annual Simulation Symposium*. San Deigo: IEEE Comput. Soc. 2002; pp. 365–380.
- [24] Rohrer M, McGregor I. Simulating reality using AutoMod. In: *Winter Simulation Conference*, vol. 2. San Diego: IEEE. 2002; pp. 173–181.
- [25] Fritz R, Horvath S, Orellana C, Wohlers J, White KP, Fairbrother R, Terry W. Real-Time Control of a Discrete-Event Simulation Using an External Controller. *2002 IEEE Systems and Information Design Symposium, University of Virginia*. 2002;1(1):69–74.
- [26] The MathWorks, Inc. Functions and Objects Supported for C/C++ Code Generation — Alphabetical List. <https://de.mathworks.com/help/coder/ug/functions-supported-for-code-generation-categorical-list.html>. 2018.
- [27] Neema H, Gohl J, Lattmann Z, Sztipanovits J, Karsai G, Neema S, Bapty T, Batteh J, Tummescheit H, Sureshkumar C. Model-Based Integration Platform for FMI Co-Simulation and Heterogeneous Simulations of Cyber-Physical Systems. In: *Proceedings of the 10th International Modelica Conference*. 2014; pp. 235–245.
- [28] Vogt M, Marten F, Braun M. A survey and statistical analysis of smart grid co-simulations. *Applied Energy*. 2018;222:67–78.
- [29] Windmann S, Niggemann O, Stichweh H. Computation of energy efficient driving speeds in conveying systems. *at - Automatisierungstechnik*. 2018;66(4):308–319.

A flexible material flow and energy simulation in the context of Industry 4.0

Andreas Buswell^{1*}, Wolfgang Schlüter¹

¹Competence Center for Industrial Energy Efficiency, Ansbach University of Applied Sciences, Residenzstraße 8, 91452 Ansbach, Germany; **a.buswell@hs-ansbach.de*

Abstract. Industry 4.0 provides numerous measures to increase productivity within manufacturing plants. However, these measures have not yet been established as extensively as current technology allows. To enable the examination of changes (e.g. Industry 4.0 measures) in non-ferrous melting and die casting plants, a flexible model has been developed. In order to ensure compatibility between different plants, the model is adaptable to variable company structures and process operations. The development of a variable energy model has proven to be a key challenge due to the high complexity of modelling the thermodynamics within melting furnaces. To evaluate the accuracy of the simulation results, measurement data gathered within two plants is used. Consequently, the validated simulation is applied to examine measures of differing magnitude (short-term/long-term planning horizons). The results show the potential increases in efficiency and process reliability which are obtainable by introducing an improved data acquisition and automation technology.

Introduction

Since its first appearance in 2011, the term “Industry 4.0” has generated wide-ranging discussions concerning its effects on modern production processes. On broad agreement, Industry 4.0 provides great potential to increase both plant productivity as well as energy efficiency [1][2].

However, its integration into production plants has not reached its full potential yet. A study conducted among German companies revealed that about one third of the participants have not implemented any Industry 4.0 measures into their processes [2] due to their perceived additional effort and low expectations of resulting benefits.

Simulation-driven examinations of operational processes can serve as an important element in the introduction of Industry 4.0 into manufacturing companies. Using accurate and validated models of the production process, it is possible to circumvent threats to regulated operation. Additionally, the effects of various measures on the operation can be investigated without making the

necessary investments for new instrumentation or machinery. Therefore, the importance of simulations in the area of production and logistics is steadily increasing.

E.g. [3] describes the use of simulations to analyze and compare flexible transport systems in order to increase intralogistical efficiency. As another example, [4] examines simulations within the automotive industry based on the coupling of event-discrete (material flow) and continuous processes (energy model). A coupling of these two types of simulations is necessary if the effects of the in-plant material flow on the energy consumption of a continuous process need to be considered.

Such a constellation arises in companies of the non-ferrous melting and die casting industry. In this industry branch, the melting furnaces are responsible for the major share of energy consumption with up to 60% of the total energy consumption of a plant [5]. A research on recent work in the field of coupled (hybrid) material flow and energy simulations (based on [6]) showed that currently no simulation model is available which can precisely simulate different companies of this sector and determine the occurring energy consumption.

1 Flexible Plant Simulation

This study is based on a flexible plant simulation for the non-ferrous melting and die-casting industry, consisting of a material flow and energy model. Its basic structure results from the general plant outline detailed in [7]. While no two plants are completely alike, they share basic processes:

- melting of solid aluminum in melting furnaces
- impeller processes (removal of impurities)
- production within die-casting machines (DCMs)
- transports (commonly accomplished by forklifts)

Therefore, similarly to the aforementioned models, the plant simulation needs to combine an event-discrete material flow model with an energy model to calculate the melting rate and energy consumption of the em-

ployed furnaces. Additionally, the simulation includes a central control module that generates orders for the individual components (in particular forklifts) based on predefined rules. Within each simulation step, data is exchanged between the material flow and energy model. The control module evaluates the resulting process parameters to generate new commands (Figure 1).

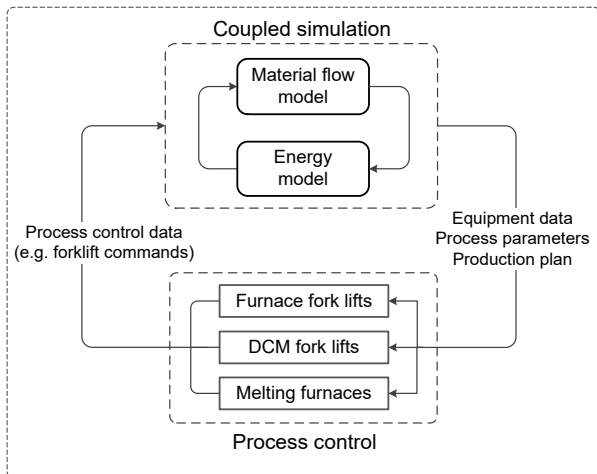


Figure 1: Simulation overview

The material flow model is built upon configuration files containing information about a specific plant (e.g. equipment number, control strategies). Based on the configuration files, the individual components of the plants are assigned to a higher-level material flow model (Simulink) and the assigned objects are parameterized in Matlab. This allows the creation of flexible models determined by a defined set of information, and the addition of further components and features.

Two types of energy models have been developed as a result of the complexity of modelling the thermodynamics inside a melting furnace. The complex model (CEM) requires extensive data and effort but offers a precise calculation of the melting rate and energy consumption of the furnace. To achieve this goal, the model calculates a set of state variables to determine the heat transfer and the resulting temperatures within the furnace. The necessary data to complete the modelling process is required to be available at high measurement frequencies (max. 1 min) and includes:

- Flue gas temperatures
- Filling levels (melting shaft/holding area)
- Aluminum temperatures (solid/liquid)
- Momentary gas consumption

Furthermore, this modelling approach requires basic knowledge about the control mechanisms of the employed melting furnace burners. This information is necessary to determine furnace behaviour, e.g. after solid aluminum is charged into the melting shaft.

Since the required measurement data is oftentimes not available and the necessary process knowledge is subject to the furnace manufacturers' confidentiality, a simplified modelling approach was developed. The simplified energy model (SEM) was acquired through the following steps:

- 1) Examination of influences on furnace melting rate
- 2) Derivation of empirical factors
- 3) Determination of empirical factors using a reduced data base

By abandoning the exact calculation of the thermodynamic processes within the furnace, the SEM gains a major advantage: the determination of the empirical factors only requires measurement data at a greatly reduced level of detail and temporal resolution. Observations of furnace behaviour have shown that the melting rate is mainly influenced by its current filling level as well as past charging processes. To address these influences with empirical factors, knowledge of the furnace filling levels (melting shaft/holding area) and charging events (time + mass) suffice. Using measurement data, the empirical factors can be fitted to depict furnace behaviour without modelling the actual thermodynamic processes. As a result, a higher number of melting furnaces becomes available for modelling (due to less data requirements) in combination with a greatly reduced effort. The comparison of the SEM with the (previously validated, [8]) complex model shows that the simplified approach allows an equally accurate depiction of furnace behaviour (Figure 2).

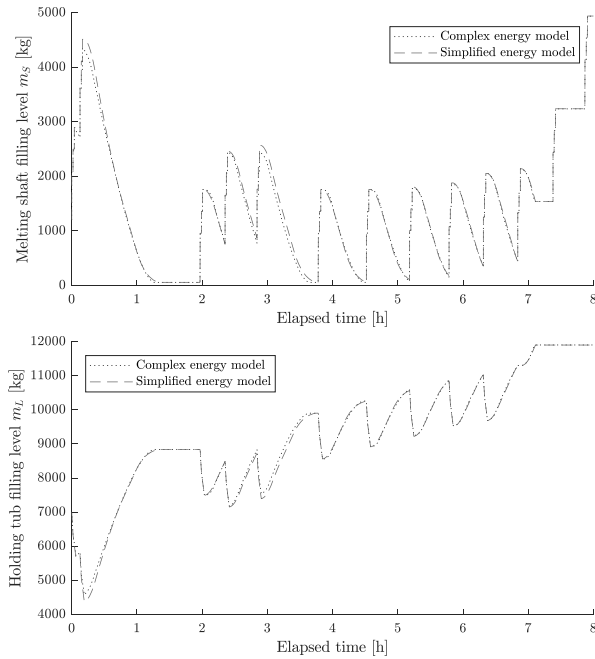


Figure 2: Comparison of simulated furnace filling levels gained from CEM and SEM

Using the calculated melting rate, the SEM is also able to provide a rudimentary modelling of energy consumption which is based on the observation that high melting rates (= high burner loads) result in an increased gas consumption.

The combination of a flexible material flow and energy model enables the examination of a multitude of different die-casting plants. For either modelling approach, a baseline of available process information and data acquisition is required. Therefore, the process of creating a feasible simulation model profits highly from the introduction of Industry 4.0 into the examined plants. The same is true for the validation of the resulting models.

2 Validation

The validation process is based on measurement data gathered from two distinct companies bearing their own equipment configurations and control strategies.

Prior to validating the simulation, an overview of the required data is necessary in order to define the limits of accuracy the simulation offers. Hypothetically, if extensive data was available at high temporal resolutions (e.g. machine vibrations) a prediction of future equipment downtimes would become available. This method (Predictive Maintenance) has been implemented within different industries, e.g. a research project within an

automotive supply manufacturer is described in [9]. Like most Industry 4.0 concepts, Predictive Maintenance requires a high digitalization standard and wide-ranging sensor data availability.

Within both plants, a basic level of centralized data acquisition was available, which allowed a great reduction of required manual measurements. However, this did not provide sufficiently detailed information to realize the aforementioned applications. A full-scale introduction of Industry 4.0 not only allows a much quicker gathering of plant-specific model parameters but also enables more advanced examinations for future simulations.

Based on the available data, the described simulation allows the analysis of the overall production process and energy efficiency as a result of various interferences within the in-plant material supply. Therefore, the validation process focuses on key parameters which allow the evaluation of the material flow model as well as the applied energy model. Table 1 summarizes the validation results for plant A based on measurement and simulation data of one calendar week (consisting of 17 shifts):

	Measurement	Simulation
Number of aluminum products	149943	148629 (- 0.9 %)
Melted aluminum	901 t	908 t (+ 0.8 %)
Gas consumption	79843 m ³	78650 m ³ (-1.5 %)

Table 1: Validation results for Plant A

Using the number of manufactured aluminum products as an indicator for the quality of the material flow simulation, the applied model and plant configuration adequately represent the actual die-casting plant. The amount of melted aluminum is strictly tied to the number of products and therefore shows an equally accurate result. The slight underestimation of produced parts results from the fact that the simulation assumes a constant percentage of reject products for each die-casting form. In the actual plant however, the number of rejects can vary, based on a multitude of factors.

The simulated gas consumption demonstrates the quality of the applied energy model which in case of plant A was created using the complex modelling approach. The resulting values are slightly less accurate than the ones

gained from the material flow model. However, with a deviation below two percent, the energy model is nonetheless capable of providing a precise prognosis of the in-plant gas consumption.

After adapting the simulation to represent plant B and using measurement data gathered during one calendar week, a model validation of the second company was conducted. Table 2 displays the achieved results:

	Measurement	Simulation
Number of aluminum products	100445	100214 (- 0,2 %)
Melted aluminum	213.0 t	214.8 t (+ 0.8 %)
Gas consumption	19770 m ³	19359 m ³ (- 2.1 %)

Table 2: Validation results for Plant B

The simulated material flow offers equal accuracy for plant A and B with slight increases for the second plant. In both cases, a correct representation of the actual material flow is possible. Concerning energy consumption, the simplified energy model provides less accurate results as compared to the complex modelling approach. However, the results indicate that the created simulation environment is nonetheless able to represent the basic workings of actual die-casting plants.

Additionally, the examinations have proven the importance of data acquisition and transmission, which forms the basis for both Industry 4.0 measures, as well as model creation. Especially the aspect of data integrity is crucial to success. The analysis of recorded downtimes for plant B DCMs showed significant gaps in the relevant databases. These gaps needed to be filled in order to allow a complete validation of the in-plant material flow. This was achieved through the combination of said databases with additional data sources (production records). A thorough analysis of measurement data to ensure its integrity is therefore just as important as the raw amount of available data. If further information were to become available, a more in-depth modelling and validation could result. This could include advanced applications like Predictive Maintenance or simpler approaches, like integrating more plant-specific control strategies (e.g. melting furnace priorities). However, the examinations conducted in this paper require a correct representation of the overall material flow which the simulation in its current state provides.

3 Use Cases

The validated simulation environment allows analysing the intralogistical processes within a plant. In the work of März *et al.* [10] different planning horizons are described that become available through means of simulation. Short planning horizons include measures that can be promptly executed within a plant without changes to present machinery. Within melting and die-casting plants, these measures could include alterations to the supply strategies for DCMs and melting furnaces or an optimization of maintenance schedules.

Mid-term planning affects production within a timespan of several days or weeks and therefore considers changes made to machine utilization (DCM $x \rightarrow$ product y). A possible application includes the optimization of production plans based on simulation results to ensure the timely completion of prioritized orders while guaranteeing high production stability (no downtimes due to aluminum shortage). The reduction of melting furnace load provides another example. Using the simulation, the demand for liquid aluminum can be determined. As a result, specific furnaces (inefficient/non-needed alloy) can be put into a stand-by operation, reducing both energy consumption and costs.

If fundamental changes to the configuration of a plant are to be considered, they affect the long-term planning process. Common examples include the extension of existing plants with new machinery (e.g. DCMs, fork lifts) or the withdrawal of old (and possibly inefficient) equipment. The simulation allows an analysis of the resulting effects on the in-plant aluminum supply by determining whether the DCMs are still sufficiently supplied with liquid metal.

As a result, the simulation provides higher planning reliability for the different planning horizons, and thereby ensures production safety. The variable plant model was applied to examine use cases of short- and long-term planning horizons to display potential applications of the created model.

3.1 Short-term measure: Intralogistical control strategies

The selection of control strategies influences the tasks fork lifts complete within the plant. Therefore, these strategies have extensive consequences on the overall production process. To determine their significance, three selection mechanisms for the supply of die-casting machines (DCMs) with melted metal were analysed.

Three examples of possible selection strategies are compared:

- Color panel: a central display provides information about DCM priorities based on discrete filling levels (e.g. red: filling below 60 % → high priority)
- Delivery order: DCMs are supplied in a fixed order
- Relative filling level: DCM selection is based on their precise percentage filling levels

While the first two methods are common examples of currently applied strategies, the relative filling level approach is a more advanced and data-dependent method that requires interconnection between the melting and die-casting operations of a plant, as well as high data availability. Therefore, this strategy benefits greatly from a high automation standard within companies. To analyse these strategies, the validation data gathered during one calendar week was used to run simulations of both plants under consideration of each selection method.

In order to increase the in-plant aluminum demand, which subsequently increases the effects of inefficient DCM supply, the machine downtimes within both plants were significantly reduced (50 %). Table 3 displays the results on plant productivity represented by the Overall Equipment Effectiveness (OEE) and the downtimes due to lack of liquid aluminum.

Plant/ Supply strategy	OEE [%]	Downtimes due to Al-shortage [%]
Plant A (color panel)	76.6	3.5
Plant A (relative filling)	76.4	3.6
Plant B (delivery order)	64.9	6.4
Plant B (color panel)	67.5	3.7
Plant B (relative filling)	70.7	1.1

Table 3: Plant productivity as a result of varying DCM supply strategies

The results illustrate that in case of an increased aluminum demand, neither plant can sufficiently supply all DCMs. However, the reasons within the two examined plants vary. For plant A, the selected supply strategy does not reap any benefits on plant efficiency. Further

analysis revealed that the melting furnaces of plant A cannot provide enough aluminum to satisfy the increased demand. As a result, the DCM supply fork lifts do not operate at full utilization, wherefore the selected strategy does not have any positive effects. In this case, measures to increase furnace effectiveness need to be considered (s. segment “Outlook”).

Within plant B however, the selected DCM supply strategy has obvious effects on equipment efficiency. This results from the fact that the melting furnaces can provide enough melted aluminum and the DCM supply fork lifts act as a bottleneck. The currently applied method within the actual plant (delivery order) is not feasible to ensure a high level of production safety in case of an increased aluminum demand. The two alternative strategies however, allow a more efficient DCM supply which results in less downtimes due to material shortages, with the most precise (and data-dependent) strategy (relative filling) offering the highest potential.

The analysis demonstrates how the simulation tool can be used to determine the effectiveness of short-term changes within a plant. These changes however, as mentioned before, require increased data availability and therefore the implementation of a more advanced data structure into the plant. The examination therefore demonstrates the interdependency between simulations and the introduction of Industry 4.0 into a company: gathering sufficient data is crucial to create and validate a model. This model can thereafter be used to examine the benefits of a further improved data availability, which could possibly result in increased Industry 4.0 efforts. As observed for the relative filling strategy within plant B, these efforts can provide visible increases in equipment efficiency.

3.2 Long-term measure: Plant expansion

In addition to the previous short-term observations, a digital representation of the plant allows a more precise planning process for possible expansions.

For example, adding another DCM into an existing plant can have adverse effects on the in-plant metal supply, resulting in increased machine downtimes. Using the simulation, the effects of additional equipment on the supply chain can be examined for two cases:

- Addition of a low load DCM (parameterized like the smallest DCM inside the plant)
- Addition of a high load DCM (parameterized like the biggest DCM inside the plant)

Due to long downtimes within both plants, the addition of a single machine into the validation configuration does not have any effects. Therefore, equipment availability was increased in accordance to the last segment (50 %) to increase the in-plant aluminum demand. Since in this configuration both plants are not fully able to supply all DCMs, the effects of the added machinery can be perceived more clearly. Again, the results on the overall plant OEE as well as potential downtimes due to aluminum shortages are compared:

Plant/ Supply strategy	OEE [%]	Downtimes due to Al-shortage [%]
Plant A (starting configuration)	76.6	3.4
Plant A (low load DCM)	75.4	5.1
Plant A (high load DCM)	74	6.5
Plant B (starting configuration)	64.9	6.4
Plant B (low load DCM)	66.2	5.7
Plant B (high load DCM)	64.2	7.8

Table 4: Plant productivity as a result of additional machinery

As expected, the additional equipment increases the aluminum shortages in plant A, which result from low melting capacities of the employed furnaces. Furthermore, the difference between low and high load DCMs becomes obvious through an increase of aluminum shortages. In further examinations this information could be used to determine the necessary increase of melting capacities to sufficiently provide all machinery as well as additional DCMs.

For plant B, the addition of a small load machine has different effects on the overall plant. Since the furnaces can provide enough aluminum, an additional machine leads to a slight increase in equipment efficiency, even though the fork lifts are operated with an inefficient strategy (s. last segment). This effect results from the fact that fork lifts can supply more than one DCM with

a filled aluminum transport container. Consequently, after supplying a DCM with a high load (and high storage capacity), the remaining metal can be used to provide equipment with lower aluminum demand. This way, supply orders for single DCMs (with a partially filled container) can be avoided. The addition of a high load DCM into the plant does not enable this effect because filling high capacity machines with remaining metal results in inefficiently used driving durations. As a result, the overall efficiency decreases. This effect could be lowered however, through the measures described in the previous segment.

The results demonstrate how the simulation allows a detailed examination of in-plant material flows. Since the simulation generates highly detailed process data, it also grants access to the analysis of correlations that would remain inaccessible with the current state of plant digitalization. Therefore, the simulation not only enables extensive examinations, but also showcases the potentials of process analysis through data availability.

4 Outlook

The described use cases display the benefits of Industry 4.0 in combination with simulations for the melting and die-casting sector. Through improved data acquisition, increasingly precise plant models become available. These models allow the examination of changes to the in-plant material flow within different planning horizons. Both economic and ecological factors are considered within the simulation, which allows the balancing of these two aspects within melting and die-casting plants. This leads to increased competitiveness, especially considering the growing share of energy costs within this branch.

However, the examinations also describe how data availability and integrity can prove problematic for model creation and application. To enable the creation of reliable models, sufficient measurements and interconnection of different processes are fundamental. Therefore it is necessary to raise the awareness for the benefits that simulations and their applications can reap within the industry.

A suitable example might be the implementation of a basic smart service controlling the furnace charging intervals in a real plant. This smart service would be based on a short-term simulation similar to segment 3.1. Using the simulation results, optimized charging orders

for melting furnaces would be transferred to fork lift drivers. As previous examinations have shown, this measure can lead to a reduction of specific energy costs by up to 20 % [7]. In order to implement these measures into plants however, an accurate measurement of the solid aluminum mass within the furnace is necessary. Additionally, feasible data visualization for the affected workers is required.

If the necessary changes to introduce smart services can be executed, the interconnection between simulations and Industry 4.0 could be used to its full extent, reaping the most benefits out of both concepts.

The simulation in its current form is able to depict variable material flow structures and calculate the energy consumption of melting furnaces based on thermodynamics and measurement data. However, the plant representation still offers potential for expansions. For example, an overall energy model would allow analysing the in-plant energy consumption in detail. The extension of the energy model would include the heat losses during:

- transportation
- impeller processes
- transfer (furnace → forklift; forklift → DCM)

Again, two variants of energy models could potentially be implemented: a complex model, determining heat losses based on thermodynamics, and a simplified model based on the assumption of a fixed heat loss per time unit. Modelling these processes would allow the calculation of aluminum temperatures within the aluminum storages of the employed DCMs. To complete the overall energy model, a simulation of the liquid aluminum within the DCMs is necessary, consisting of additional heat loss models as well as a depiction of the DCM heating units. If all of these steps are conducted, the simulation gains further options for in-plant optimization, e.g. determining whether liquid aluminum should be stored within melting furnaces or DCMs. Thus, the number of applications for different planning horizons increases.

References

- [1] Pereira A, Romero F. A review of the meanings and the implications of the Industry 4.0 concept. *Procedia Manufacturing*. 2017; vol. 13: 1206-1214. doi: 10.1016/j.promfg.2017.09.032
- [2] Arntz M, Gregory T, Lehmer F, Matthes B, Zierahn U. Arbeitswelt 4.0 - Stand der Digitalisierung in Deutschland. *IAB-Kurzbericht*. 2016; vol. 22: 1-8
- [3] Scholz M, Kreitlein S, Franke J. E|Flow - Decentralized Computer Architecture and Simulation Models for Sustainable and Resource Efficient Intralogistics. *Applied Mechanics and Materials*. 2017; vol. 856: 117-122. doi: 10.4028/www.scientific.net/AMM.856.117
- [4] Wenzel S, Peter T. Simulationsgestützte Planung und Bewertung der Energieeffizienz für Produktionssysteme in der Automobilindustrie. In Rabe M, Clausen U. Simulation in Production and Logistics 2015. *16. ASIM-Fachtagung Simulation in Produktion und Logistik*. 2015 Sep; Dortmund. Stuttgart: Fraunhofer Verlag. p 535-544.
- [5] Belt C. Current State of Aluminum Melting and Holding Furnaces in Industry. *JOM*. 2015; vol. 67(11): 2690-2695.
- [6] Wenzel S, Peter T, Stoldt J. Betrachtungen energetischer Einflussfaktoren in der Simulation in Produktion und Logistik: Eine Literaturanalyse. In Wenzel S, Peter T. Simulation in Produktion und Logistik 2017. *17. ASIM-Fachtagung Simulation in Produktion und Logistik*. 2017 Sep; Kassel. Kassel: kassel university press. p 9-18.
- [7] Buswell A, Schlüter W. E|Melt: Simulation-Driven Analysis of Energy Efficiency Measures inside Non-Ferrous Melting and Die-Casting Plants. *Applied Mechanics and Materials*. 2018; vol. 882: 182-189. doi: 10.4028/www.scientific.net/AMM.882.182
- [8] Schlüter W, Schmidt J, Henninger M. Key figures for production control in non-ferrous melting and die-casting plants based on the assessment of the operating state. *Applied Mechanics and Materials*. 2017; vol. 871: 176-185. doi: 10.4028/www.scientific.net/AMM.871.176
- [9] Lindström J, Larsson H, Jonsson M, Lejon E. Towards intelligent and sustainable production: combining and integrating online predictive maintenance and continuous quality control. *Procedia CIRP*. 2017; vol. 63: 443-448. doi: 10.1016/j.procir.2017.03.099
- [10] März L, Weigert G. Simulationsgestützte Optimierung. In: März L, Krug W, Rose O, Weigert G. *Simulation und Optimierung in Produktion und Logistik*. Heidelberg: Springer-Verlag; 2011. p 3-12.

Approach on Evaluating Material Handling Simulation Runs under Consideration of different Target Groups

Martin Däumler, Karl-Benedikt Reith*, Simon Hochholzer, Thorsten Schmidt

Chair of Material Handling, TU Dresden, Dresden, Saxony, Germany; *karl_benedikt.reith@tu-dresden.de

Abstract. Usually two user groups with different knowledge background are involved in visualizing and exploring discrete event simulation results: the simulation expert who crafted the simulation and the external user, whose system was simulated. In order to provide these results in a way that enables visual exploration of the simulation results for both types of users a visualization tool for discrete event simulation data was developed. The tool is responsive, interactive and based on the charting library *plotly*. It can be used to filter and compare multiple simulation runs with various input and output data or to analyze a single simulation run in depth. Furthermore, it enables visual mapping without the need of previous knowledge about simulation tools or any restrictions to license issues.

1 Introduction

Motivation

Due to increasingly powerful simulators and data storage capacities, the amount of simulation data steadily grows. In consequence, evaluating the results becomes more and more challenging. This paper focuses on a visualization tool which helps to explore and analyze data resulting from discrete event simulation (DES) runs of systems in logistics and material handling.

Since simulation includes different target users, a shared basis for communication is needed. A suitable visualization or visual interface helps to improve the representation and interpretation of input and output data resulting from simulation runs. This process is relevant for all target groups regarding to their requirements [1]. Furthermore, there is no generally applicable methodology for the interpretation of simulation results. The methodology depends on the nature of

the simulated system as well as on the questions examined [2].

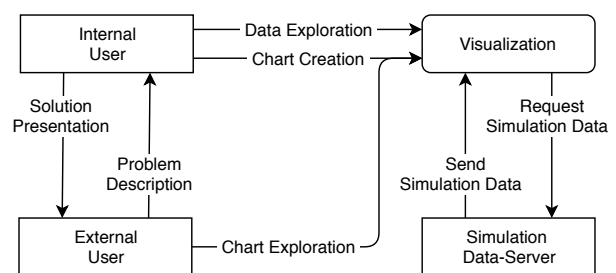


Figure 1: Internal and external user interaction in the context of information visualization with a mediating visualization tool separated from simulation.

More generally, the target groups addressed by [1, 3] can be divided into internal and external users. Usually simulation studies are performed by a simulation expert (internal user) who models a given system or problem. He conducts multiple simulation runs with various input parameters in order to answer questions given by his client (external user) who is familiar with the real system but usually not with the simulation itself. On the one hand a post-simulation visualization tool helps the simulation expert to sort and explore the data (see fig. 1). On the other hand a visualization tool especially enables an external user to explore and understand the simulation results on his own, without needing a deep understanding of simulation or visualization. So an external user – like a client – is able to use this framework or to collaborate with the internal user. The external users knowledge about the real world system behavior and therefore his ability to merge results can be included in the complex evaluation process. Furthermore, some additional questions that might have not existed during the simulation phase, can be answered

quickly, without extensively getting back into the simulation phase.

Generally speaking, there are two occasions, when a simulation is performed. Either to answer a specific question – like how many vehicles are necessary in a material handling system to perform all transport jobs – or to gain a general deep understanding of the system behavior and the effect of input parameter constellations [4]. There are a couple of methods to search efficiently in a (input) parameter space and to still avoid extensive computing times [5, 6]. These methods allow to reduce the number of simulation runs dramatically and to still uncover the influence of significantly positive or negative input parameter constellations. Nevertheless, with practically complex simulation studies these methods still lead to a high number of simulation runs. Consequently, a reasonable visualization tool supporting the analysis of the generated amount of data is still necessary.

Outline

In the following section, related work concerning discrete event simulation in material handling, the resulting type of data and its visualization is described (see sec. 2). In section 3 a responsive, interactive visualization tool is introduced, to deal with the problems outlined previously. To get a better understanding for the application a sample implementation of data resulting from the simulation of a material handling system with automated guided vehicles (AGVs) is described in section 4.

2 Related Work

DES is one of the most widely used tools to analyze and support the planning process of systems in production, manufacturing, logistics and material handling [7]. On the one hand, simulation experts are focusing on modelling concepts or other questions, like the reduction of complexity [8, 9], because the simulation itself is challenging and time consuming. On the other hand, specific questions are the focus of investigation. Some examples are described in the following.

[10] evaluated the performance of a flexible manufacturing system, by analyzing different job dispatching and vehicle assignment rules for different layout configurations and different job sets. [11] simulated the supply of auxiliary resources in a semiconductor

fab with AGV. They examined different layout configurations, vehicle dispatching rules, various numbers of parking locations and empty vehicle balancing etc. by comparing different key performance indicators (KPIs). [12] investigated the impact of decentral vehicle control on the overall performance. He simulated multiple layout configurations, various vehicle dispatching strategies for different real-world transport scenarios, etc. However, the system knowledge lies with the simulation expert, the transfer of knowledge to a third party is usually not envisaged.

Common DES tools offer integrated methods for visualization, or even live-visualization. *AutoMod*[®] and especially its statistical analysis tool *AutoStat*[®] help the user investigating the impact of parameter variations by offering a two-dimensional visualization (lineplots, bar-charts etc.) that can be applied on multiple factors [13]. Due to the limitation to single static charts, an extended visualization with multiple input and output parameters can get confusing quickly. The simulation tool *AnyLogic*[®] provides diagrams like bar or pie charts to illustrate distributions and charts for time dependent data. Furthermore live visualization during simulation runs is possible [14]. *Plant Simulation*[®] enables the live-displaying of important values during the simulation and multiple charts (histogram, Sankey-diagram, etc.) for data visualizing and exploration [15].

Two perspectives have been established for the analysis of data from simulations of logistic systems: (1) order-oriented viewpoint and (2) element-oriented viewpoint. The second view implies data of mobile model elements, data of stationary model elements, data of statistical model elements and data of queues [2]. Depending on the application area, there is a different relevance of visualization in relation to the target group [1].

The tool presented in this publication is based on the principles of visual analytics. The basic concept of visual analytics is to support decision making by enabling the possibilities of interaction and visual representation of data [16]. In the field of visual analytics this work focuses on information visualization.

[17] give a survey on methods for visualization and visual analysis of multifaceted scientific data. They categorize techniques depending on the type of data starting at visual mapping over interactive visual analysis and ending at computational analysis. They also map the categorized approaches to the type of data with distinction to multimodel, multirun, multimodal, multi-

variate and multidimensional data. [18] describe a visualization pipeline in their taxonomy from data transformation, over visual mapping to the transformation of views. They build subcategories, like axis based method approaches in visual mapping. Furthermore, three main types of user interaction were identified – computation centric, interactive exploration and model manipulation. Regarding the categorizations of [17] and [18], the resulting data from simulation can be classified to multirun and multivariate data. Moreover, the focus of this work is to support the process of interactive visual analysis, mapping and exploration for an external user and the communication between internal and external user. Also, the selection of plots and their complexity is an essential part for building a base of communication between different target groups.

Radar plot, scatter plot or matrices of scatter plots and parallel coordinates plot are popular for visualizing multivariate data [19]. Furthermore, they are implemented in various static and dynamic visualization libraries¹. There are also existing combinations of parallel coordinates, scatter plots or scatter matrices [20, 21] and other plots, like bar plots to support the user in the exploration of datasets from many different perspectives [22]. These complex plots must be seen in the context of our idea proposed in section 1, whereas especially an external user should be able to analyze the data. This seems to be very challenging when using uncommon plots that are difficult to understand.

Nevertheless, an interactive visualization tool to enable and improve post simulation visual data exploration is not included within common DES software. In case the included charts are sufficient for data exploration, there is still the main disadvantage of using visualization methods within simulation tools – it is usually not possible to provide the data to an external user (see fig. 1) for interactive and user friendly exploration due to license problems. Often DES-tools provide interfaces (to *Excel*, databases or text files) for exporting simulation data, where visualization tools are based on.

3 Visualization Tool

The presented visualization tool is reusable for various simulation studies and simulators, especially flexibly definable regarding to the structure of input and output data. The focus was to build a tool which helps to

¹See e.g. *plotly.js*: <https://plot.ly>, *matplotlib*: <https://matplotlib.org>, *D3*: <https://d3js.org>

improve the communication between internal and external users in different phases of simulation studies and to answer unique or even recurring questions. Furthermore, filtering and comparing simulation runs as well as analyzing single runs in detail is eased, even without extensive prior knowledge to the technical system.

3.1 Structure

The tool consists mainly of the following three components which are shown in fig. 2: 1. Simulation Component 2. Data Component 3. Visualization Component and the linkage between them.

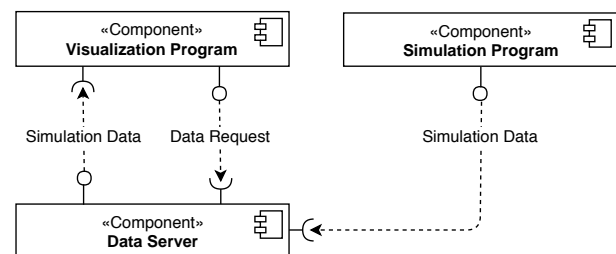


Figure 2: Component diagram showing the three main components of the tool.

3.1.1 Simulation Component

The simulation component can be regarded as a black box, that transfers a certain input parameter constellation in a specific simulation output [7], as shown in the practical use case in section 4. The described framework is especially applicable for visualizing data sets consisting of various simulation runs with a large number of input and corresponding output data, as described later in section 3.1.3. However, the specific use case and even the specific simulator are not crucial for a connection with the visualization tool. As long as the shape of the data is similar or can easily be transferred, discrete event simulators, like *Plant Simulation*[®] or *AnyLogic*[®] can be used in combination with the tool. Originally, the source of inspiration for the visualization tool came from different simulation projects with the discrete event simulator *AutoMod*[®] where the material flow and layout characteristics for automated material handling systems were modeled and analyzed.

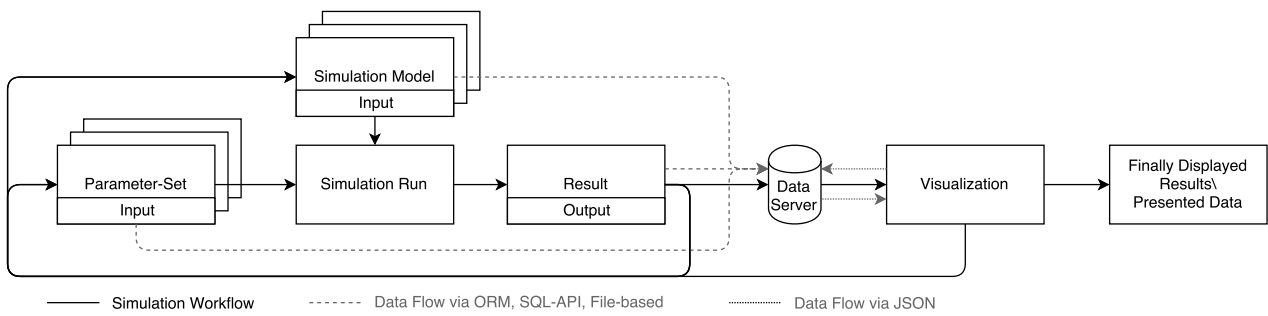


Figure 3: Workflow and data flow diagram showing data resulting from simulation runs. Data storage and the visualization enabling information visualization or presentation of results.

3.1.2 Data Component

The data component is used to connect the visualization component and the simulation component. The input and output data from the simulation are stored and are applicable for further visualization. The data can be stored in two main ways.

On the one hand, data can be transferred in the form of files – ideally in JSON format. Data in CSV format may also be used, with the limit that parameters and output values are not captured here, thus prohibiting these data sets from being filtered. This option does not create the overhead of a database, which is especially advantageous during the development phase, when the data model may be under constant changes.

On the other hand, SQL-based databases or any kind of database – depending on the purpose – can be used for storing data. In addition, a representational state transfer (REST) API is required to answer the data request of the visualization component. This request must be mapped to a query for the database and the data need to be converted into JSON format. We used the Python-based web framework *Flask*² at this point because it is compatible with various databases, allows the integration of other packages and supports object-relational mapping (ORM).

3.1.3 Visualization Component

The last few years have shown increasingly powerful JavaScript and HTML5 applications for visualization of various types of data in an interactive and user friendly way. *Chart.js*, *D3.js*, *Google Charts* and *plotly.js* are just some examples.

²<http://flask.pocoo.org/>

We decided to choose the charting library *plotly.js*³ to create charts from simulation data, because compared to other libraries such as a standalone *D3.js*, *plotly.js* offers a very high-level access to charts and many common chart types are already implemented. It uses the powerful *D3.js* under the hood and due to the popularity of both libraries, we expect that a vibrant community will continue with the development in the future. The *Ionic Framework*⁴ bundles useful technologies for building cross-platform applications based on web technologies, such as *Angular*⁵ and *Cordova*⁶. It also offers commonly used interface components, which increases development speed and creates a consistent appearance. We designed the system to be runnable on as many systems and browsers as possible to enable external users to access the visualization component independently of their hardware and operating system. Therefore we used *Cordova*, which allows to create mobile applications, as well as *Electron*⁷, which helps to create desktop applications for any major platform.

[23] reviewed taxonomies for information visualization with relevance to interaction and proposed their own taxonomy based on the user’s intent. In their taxonomy they describe seven types of interaction: select, explore, reconfigure, encode, abstract/elaborate, filter and connect. We tried to follow their taxonomy as design guideline, to allow the user to interact with the charts in multiple ways in order to find meaning from the data.

The visualization component allows the user to select data sets from a list to visualize only the selection. It is possible to explore visualizations through panning,

³<http://www.plot.ly>

⁴<https://ionicframework.com>

⁵<https://angular.io>

⁶<https://cordova.apache.org/>

⁷<https://electronjs.org/>

which is useful after having zoomed in on some detail of a chart. Different representations of the same data can be created, compared and moved around spatially by drag and drop interaction in a dashboard-like style, which correlates with the encode intent. *Plotly* provides two common interaction techniques to show more details: 1) a plot may be zoomed in and out, thus providing a more detailed view or going back to the overview and 2) hovering over a data item shows a tooltip with the concrete values associated to the visual representation. These allow to switch between an abstract and an elaborate representation of the data. A parallel coordinates visualization allows the user to visually filter data sets and proceed to the further analysis only with the filtered data sets. The corresponding selection of data sets is displayed in a table below the parallel coordinates visualization.

Generally, the internal user can implement all charts provided by *plotly* to make them available for the external user. First, we implemented the diagram types that are most commonly used, such as line plot, bar plot, box-whisker plot and scatter plot. The user may edit all of them, e. g. to adjust axis markings, a legend or the title. As the tool is not restricted to one-time visualizations, the usage for answering recurring tasks is also considered. Therefore, the configured diagrams can be saved in *views*, selected and modified again later in order to deal with recurring questions. This can mean an adjustment for different data.

3.1.4 Linkage

The relation between the separate components of the tool is shown in fig. 3, including the typical workflow and data flow. The input parameters as well as the results are stored on the data server, which provides the data for the visualization component via JSON. If a generic class with a corresponding data model is used, the provision of the input parameters via an ORM will be advantageous, see sec. 3.1.2. Neither the data server nor its interface or the visualization component needs to be adapted. In addition, run-control can be carried out easily. We ourselves categorize the data input for the simulation in two sets – 1. *structural specifications* in the following called *major parameters* and 2. *minor parameters*.

Major parameters are defined as changeable values but in the meaning of not directly comparable like two different layout options in material handling systems or transport patterns used for simulation. Usually they are

qualitative in nature. So, they are changing the system structurally. Similar to major parameters, minor parameters are input to simulation runs but they vary naturally. They are usually quantitative in nature, like handling times for carriers or machining times. Minor and major parameters can be of both categories (order-oriented, element-oriented) as discussed in sec. 2.

3.2 Design Characteristics

The selected structure assumes independent components. This allows a coupling to various simulators. All packages used are open source software. By use of a lightweight browser-based rich internet application (RIA) there is no dependency on operating systems. The external user does not need to install any software-package.

Regarding data supply we wanted users to feel free how to store and interact with their data. Therefore, we integrated database support and also file-based working options. Even if the data structure is flexible, a certain form must be adhered to, as otherwise different data records cannot be compared with each other. However, the internal user can already take this structure into account in the modeling phase. Therefore, different data structures require preprocessing.

The external user is dependent on the internal user in the usability of the visualization component. This applies not only to the data provided, but also to the repertoire of charts.

In the end, users should be able to make better decisions because they can explore their data with thousands of simulation runs as well as deeply analyze a few runs. This allows different decision levels within the same simulation study and furthermore various perspectives on the same problem. Quite often detailed questions come up after a simulation projects was finished – in consequence post-simulation analysis is very useful.

4 Sample Implementation for an Automated Guided Vehicle System Simulation

4.1 Simulation

The described visualization tool was applied in a use case similar to example 12.6 in [7]. A fleet of AGVs was intended to automate the transports within a semi-conductor fab.

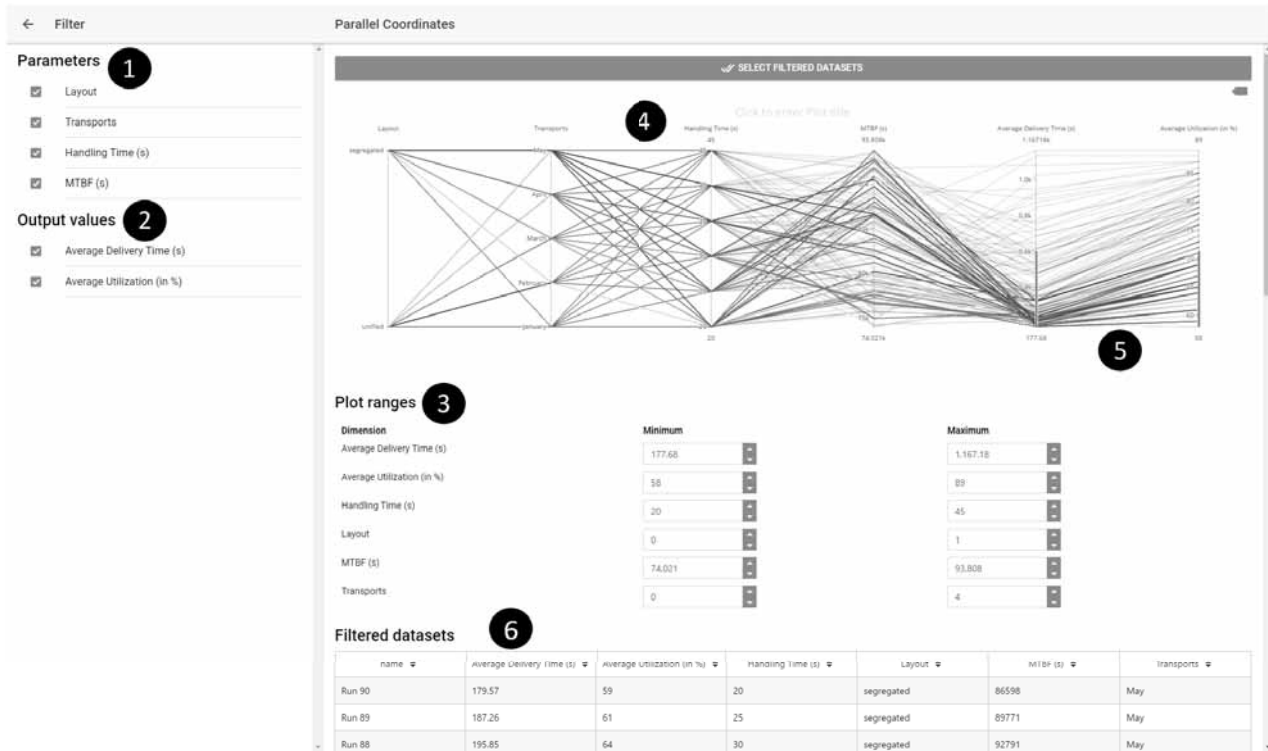


Figure 4: Screenshot of a parallel coordinates plot for run selection in the visualization tool: (1) Selection of input parameters of interest (2) Selection of output values of interest (3) range selection for each axis (4) Parallel coordinates plot of all runs; color adjusted according to the order of the axes and the values on the axes (5) Manual selection of runs according to values of the last two axes (6) List of selected runs (only the first three are visible here).

The goal of a supporting study is to find solutions to interdependent questions like the total number of vehicles necessary to reach a specific system performance, the best aisle layout for the vehicles and what system or vehicle parameters (e.g. speed, handling time, etc.) are most influential to the overall system performance. In addition, not yet specified questions that may come up during the AGV implementation phase should be answered quickly, without having to restart extensive analyses again. Therefore, the results of the study should be easily accessible and understandable. Due to the dynamic nature of the problem and the diversity of the tasks, we decided to perform an extensive simulation study in combination with the presented tool for visualization of various results.

The simulation system was modeled using the discrete event simulator *AutoMod*[®]. As already pointed out in section 3.1.1 the used simulation software itself is not relevant for the the regarded further analysis and visualization of the results with the visualization tool.

The simulation model was designed as generic as possible. Various minor simulation input parameters, like the number of AGVs, vehicle acceleration, vehicle velocity, battery consumption, charging speed, material handling times for loading machines and storage, mean time between failure (MTBF) and mean time to repair (MTTR) are implemented in the simulation and can be varied. Additionally, some major input parameters, like the overall area layout or the transportation pattern can be varied as well. According to [2] both perspectives for analyzing data are taken into account. To control the different simulation runs in *AutoMod*[®] we use a *Python* script, i. e. creating and handing over the input, collecting the output and finally storing the data in a suitable format.

To provide a solid basis for answering the mentioned specific questions and those that might come up in the future, it is necessary to build a wide data basis. Therefore, various scenarios were simulated which differ in their specific input, i.e. they have different major parameters and minor parameters. In total, more than two

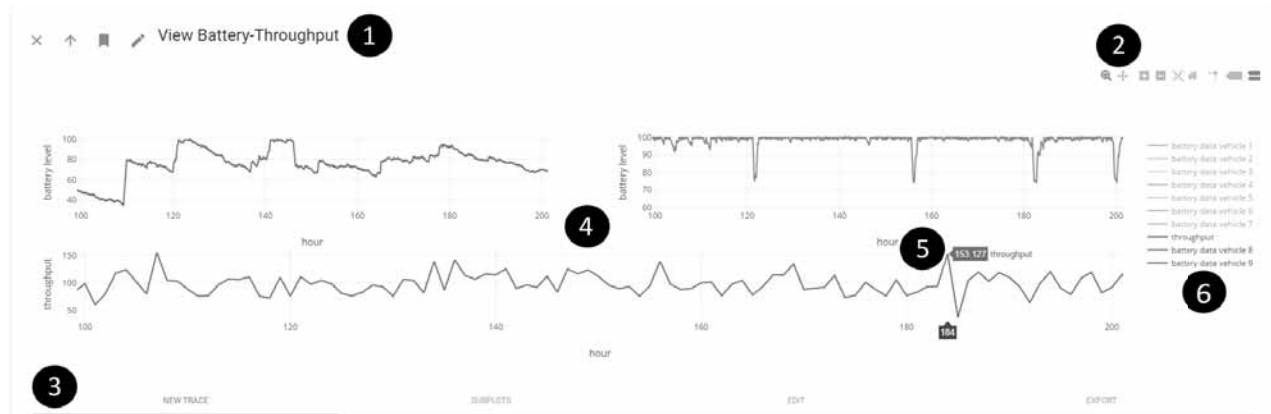


Figure 5: Full size view of an in-depth analysis of a single run: (1) Menu bar with skipped navigation area (2) *plotly* mode bar (3) Task bar (4) Manually selected and adjusted diagrams: vehicle battery of vehicle 1 (top left) vehicle battery of vehicle 2 (top right) and overall throughput (bottom) in an adjusted time range from 100 to 200 (see axes of abscissae) (5) Cursor with highlighting of extremal value (6) Legend with selection for interactive layering.

thousand simulation runs were performed to cover as many input parameter constellations as possible.

4.2 Visualization

In the presented use case, the internal user (simulation expert) crafted the simulation model. Whereas the producer of the AGVs and the customer, who wants to implement AGVs for automating his production can be regarded as external users. An external user has a much deeper understanding of the underlying system restrictions and the real world transport system requirements, but not in detail about general simulation paradigms and the simulation model. During the whole time of the study and vehicle implementation, the visualization tool showed multiple advantages.

By clearly showing specific relationships between input and output data of multiple simulation runs, or visualizing time dependent data, like the vehicle battery level over time, the visualization tool supported the simulation expert in generally validating the simulation model at an early stage. Unexpected system behavior, or even deadlocks were recognizable at once. In addition to the early stage validation through the simulation expert, the visualization tool also enabled the external user to compare first simulation results with the actual real world system behavior and hence validate the simulation on another much deeper level. Inaccuracies of the model that are only recognizable with a deep real world system understanding could be detected and corrected early in the simulation phase. This has also lead to a

first involvement of the external user in the simulation study at an early stage and he was encouraged to deal with the simulation. Hence, open questions could be addressed, misunderstandings eliminated and disagreements were cleared which increased the acceptance of the results later on. After the early simulation validation, the visualization tool was used for selecting runs, comparing multiple runs and a detailed analysis of single runs. Fig. 4 shows an example visualization of a parallel coordinates plot in order to select runs of a set. Various input parameters like the used layout, the transport pattern, the AGV handling time, the MTBF and important KPIs like the average delivery time or the average vehicle utilization are displayed on the axes. Each line represents a single simulation run with specific input parameter constellations and corresponding simulation results (KPI). In fig. 4 the runs are filtered according to the KPIs. All simulation runs with a vehicle utilization below 70 % and an average delivery time below 600 s are selected. In a next step based on this subset, the selected runs can be compared using other (sub) plot types. This can be seen in fig. 5, where the battery level of two vehicles over a time period of a specific simulation run is compared to the total transportation system throughput. As this is a significant relationship, this view was saved and applied on multiple simulation runs and was provided for the external user, to explore the data and possible relations on his own. Having had the possibility to use many views and generate new views, the external user obtained a holistic picture of the (simulation) system behavior.

Furthermore, the simulation expert was able to provide data for re-visualization all input parameter ranges after the simulation phase and is not forced to create a large number of static graphics hoping to cover all relevant relationships that might come up in the future. As simulation runs can be filtered quickly (see fig. 4) and important views can be saved and applied on different runs (see fig. 5), results are visualized on a very short notice and can be discussed right away like during a conference meeting.

5 Conclusions

5.1 Summary

In this paper a tool was presented which helps to evaluate DES runs. Three main components – simulation, visualization and data – were described. We pointed out that there are no dependencies on a specific simulation software and that an external post visualization is very useful for many purposes. Furthermore the used software tools empower responsive and interactive charting. This makes it possible to use the tool for one-time and recurring simulation studies and questions.

The tool supports an internal user during the modeling phase and helps to evaluate first simulation runs with quick system analysis. Therefore, unusual system behavior can be found and bugs are identified much easier. It also supports the post analysis process by transferring the system knowledge to an external user. They can easily explore data by themselves or more likely, access data in views predefined by the internal user. So, the tool works like a moderator between the simulation expert (internal user) and the external user. It is not a substitute for an efficient design of experiment but it supports the user in visual analytics.

Finally, we showed a practical use case of an AGV system where we used the tool for an exemplary interactive visualization from simulation run data. Moreover, we described the way for selecting simulation runs by using an interactive colored version of parallel coordinates plots.

5.2 Further Work

There are still some open tasks. We plan to implement other plot types especially the ones, helping to differ between simulation runs and supporting the selection of runs of interest.

Concerning recurring questions in simulation, the implementation of special predefined analyses seems to be rewarding. The AGV system showed in section 4 serves as an example.

MongoDB has been very popular in recent years and can be a useful addition as well, especially regarding our current project where we created a text wrapper for standard simulation trace files, like from an *Automod*[®] run. In this case a *NoSQL* document-orientated database, like *MongoDB* is an alternative for storing the trace files.

Currently, a highlighting of outliers in box-and-whisker plots is implemented. In the future, we plan to implement a highlighting of outliers and interesting data point in all types of plots. The highlighting of interesting points, like local and global maximum or minimum values could be user interacted as well as automatic. We did not include reconfigure and connect interactions according to the taxonomy of [23], as they were not of highest importance for our project, but they certainly offer possibilities for future enhancement. In order to elaborate the idea of a tool which helps to moderate, we want to implement text areas for describing the data visualization and the intention. Furthermore, a feedback channel in the meaning of a comment section to receive information from users could be useful.

References

- [1] VDI-Richtlinie 3633: Simulation of systems in logistics, materials handling and production – Part 11 Simulation and visualization. 2009.
- [2] VDI-Richtlinie 3633: Simulation of systems in logistics, materials handling and production – Part 3: Planning and evaluation of experiments. 1997.
- [3] Wenzel S, Bernhard J, Jessen U. Visualization for Modeling and Simulation: A Taxonomy of Visualization Techniques for Simulation in Production and Logistics. In: *Proceedings of the 35th Conference on Winter Simulation: Driving Innovation, WSC '03*. Winter Simulation Conference. 2003; pp. 729–736.
- [4] Feldkamp N, Bergmann S, Strassburger S. Knowledge discovery in manufacturing simulations. In: *Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. ACM. 2015; pp. 3–12.
- [5] Sanchez SM, Wan H. Work smarter, not harder: a tutorial on designing and conducting simulation experiments. In: *Proceedings of the 2015 Winter*

- Simulation Conference*. IEEE Press. 2015; pp. 1795–1809.
- [6] Horne GE, Meyer TE. Data farming: Discovering surprise. In: *Simulation Conference, 2004. Proceedings of the 2004 Winter*, vol. 1. IEEE. 2004; .
- [7] Banks J. *Discrete-event system simulation*. Upper Saddle River, NJ ; , Munich [u.a.] :: Pearson Prentice Hall,, 5th ed. 2010.
- [8] Siegel A, Turek K, Schmidt T. Modular, complexity reduced Material Handling Simulation Model for Air Cargo Terminals. In: *The 2016 European Simulation and Modelling Conference*, vol. 30. EUROSIS-ETI Publication; pp. 467 – 472.
- [9] Rank S, Hammel C, Schmidt T, Schneider G. Reducing Simulation Model Complexity by Using an Adjustable Base Model for Path-based Automated Material Handling Systems: A Case Study in the Semiconductor Industry. vol. WSC '15. IEEE. 2015; pp. 2896–2907.
- [10] Reddy B, Rao C. Flexible manufacturing systems modelling and performance evaluation using AutoMod. *International Journal of Simulation Modelling*. 2011; 10(2):78–90.
- [11] Ndiaye MA, Dauzère-Pérès S, Yugma C, Rullière L, Lamiable G. Automated transportation of auxiliary resources in a semiconductor manufacturing facility. In: *Winter Simulation Conference (WSC), 2016*. IEEE. 2016; pp. 2587–2597.
- [12] Klein N. The impact of decentral dispatching strategies on the performance of intralogistics transport systems. 2012;.
- [13] Banks J. Getting started with AutoMod. *Bountiful, UT: Brooks-PRI Automation*. 2000;.
- [14] Borshchev A. *The big book of simulation modeling: multimethod modeling with AnyLogic 6*. AnyLogic North America. 2013.
- [15] Bangsow S. *Manufacturing simulation with plant simulation and simtalk: usage and programming with examples and solutions*. Springer Science & Business Media. 2010.
- [16] Keim DA, Mansmann F, Schneidewind J, Thomas J, Ziegler H. Visual analytics: Scope and challenges. In: *Visual data mining*, pp. 76–90. Springer. 2008;.
- [17] Kehrer J, Hauser H. Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. *IEEE Transactions on Visualization and Computer Graphics*. 2013;19(3):495–513.
- [18] Liu S, Maljovec D, Wang B, Bremer PT, Pascucci V. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Transactions on Visualization and Computer Graphics*. 2017;23(3):1249–1268.
- [19] Claessen JHT, van Wijk JJ. Flexible Linked Axes for Multivariate Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*. 2011; 17(12):2310–2316.
- [20] Yuan X, Guo P, Xiao H, Zhou H, Qu H. Scattering Points in Parallel Coordinates. *IEEE Transactions on Visualization and Computer Graphics*. 2009; 15(6):1001–1008.
- [21] Viau C, McGuffin MJ, Chiricota Y, Jurisica I. The FlowVizMenu and Parallel Scatterplot Matrix: Hybrid Multidimensional Visualizations for Network Exploration. *IEEE Transactions on Visualization and Computer Graphics*. 2010;16(6):1100–1108.
- [22] Viau C, McGuffin MJ. ConnectedCharts: Explicit Visualization of Relationships between Data Graphics. *Computer Graphics Forum*. 2012;31(3pt4):1285–1294. URL <http://doi.org/10.1111/j.1467-8659.2012.03103.x>
- [23] Yi JS, Youn aK, Stasko J. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*. 2007;13(6):1224–1231.

Acknowledgements

This work was carried out as part of the research project "Investigation of basics and concepts for the design of an automatically reacting semiconductor factory regarding to changing requirements in production volume and product mix" (Responsive Fab), co-funded by grants of the European Union and the Development Bank of Saxony.



Simulation-based Evaluation of Dynamic Vehicle Routing Problem Features for Algorithm Selection

Thomas Mayer*, Tobias Uhlig, Oliver Rose

Institute of Computer Engineering, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany; *thomas.mayer@unibw.de

Abstract. Algorithm selection based on problem instance features is a common method to choose approaches for NP-hard combinatorial optimization problems. Features concerning the location of nodes for the Vehicle Routing Problem (VRP) are intensively studied. Although the dynamic version of the problem is gaining steadily in importance, there is a lack in research targeting problem features describing the dynamism of an instance. In our paper we translate known VRP features to features for the Dynamic Vehicle Routing Problem (DVRP). We investigate the suitability of them for algorithm selection. To this end, we explore the performance space of a Greedy and a Re-planning algorithm for various dynamic problem instances using simulation and an evolutionary algorithm. For the algorithm selection we investigate our features in combination with a supervised machine learning technique. The applicability of our features is demonstrated in a use case for autonomous algorithm selection of DVRP instances.

Introduction

In the last years, the research interest in vehicle routing has significantly increased due to our modern way of life. We order different goods of daily use over the Internet, we share cars and bicycles and use on-demand transportation services. To provide all these services, complex Vehicle Routing Problems (VRPs) with several constraints have to be managed and solved. The fundamental VRP was introduced by Dantzig and Ramser in [1] as generalization of the Traveling Salesman Problem (TSP) introduced from Flood in [2]. Essentially, it encompasses the planning of routes for vehicles to satisfy customer demands. The routes usually start and end at a central depot. A current survey and taxonomy on the VRP is, for example, [3]. Es-

pecially, the advances of information and communication technologies and the changing nature of the real-world problem instances put the dynamic version of the VRP into the research focus. Pillac et al. in [4] and Psaraftis et al. in [5] describe an explosion in the number of related papers after the year 2000. The Dynamic Vehicle Routing Problem (DVRP) was first introduced from Wilson and Colvin in [6], with a description of a computer controlled Dial-A-Ride system in Rochester, NY (USA). The DVRP is an extension of the VRP and is characterised by changing problem parameters over the time, [4], for example, additional customer demands occur during route execution. A current survey on the DVRP is provided in [7]. As a result of the increased research interest, an increasing number of different solution approaches to tackle the DVRP are available. Psaraftis et al. identifies the five main solution methods: *Tabu Search*, *Various Neighborhood Search*, *Insertion Methods*, *Dynamic Programming*, and *Markov Decision Process* in [5]. For each method several different heuristics and implementations exist. Simulation is also commonly used to handle the dynamic and stochastic aspects of vehicle routing. For example Juan et al. improves the Clarke and Wright heuristic, [8], by Monte Carlo Simulation, [9]. Juan et al. provide a review of methods using simulation to solve combinatorial optimization problems in [10]. The concept of Simheuristic is introduced, a methodology that integrates simulation into the solution finding process for combinatorial optimisation problems. [11] use this methodology to solve the Two-Echelon Location Routing Problem, a combination of the Capacitated Location Routing Problem (CLRP) and a VRP. Due to the intense research in this area and the increased number of algorithms and solution approaches for different variations of the DVRP, the following research questions arise: Which is the best solution approach or algorithm for a specific problem instance? And, is it possible to distin-

guish between problem instances to select the algorithm that has a better expected performance? With regard to the No Free Lunch Theorem introduced in [12], we know that there is no strictly superior heuristic that outperforms all other heuristics for all problem instances. That means some problem instances are more difficult or harder to solve for certain algorithms compare to others. Our research focuses on learning which problem instances are easier or harder for a certain algorithm based on problem features. The acquired knowledge can be used to select the likely better performing algorithm, means the algorithm that likely delivers the best solution. The question for the best algorithm given a problem instance is not only arising in the context of vehicle routing. It is a general problem, formalised and introduced by Rice in [13] as Algorithm Selection Problem (ASP). The bases for algorithm selection are features characterising the problem instances. For the static version of the VRP several problem features, considering the location of the customer demands, exist, [14]. But features for the dynamic version of the problem are rarely available, [15]. Our work closes this gap by transferring existing location-based TSP and VRP features, which we discuss in Section 2, into dynamic problem features outlined in Section 3. To validate the suitability of our features for algorithm selection we use them for algorithm performance prediction and autonomous algorithm selection supported by simulation. To this end, we first evolve dynamic problem instances from existing static instances with the help of the general purpose metaheuristic optimisation framework SEREIN introduced in [16]. The evolved problem instances are solved with a Greedy and a Re-planning algorithm, which we implemented in the open-source Rich Vehicle Routing Problem Simulator (RVRP Simulator) introduced in [17]. With the generated dynamic instances and the simulation results we are able to investigate our introduced features and whether they are suitable to meaningful distinguish between problem instances. The results of this investigation and the dynamic instance generation are also outlined in Section 3. To validate our problem features in the context of algorithm selection, we implemented a machine learning technique to model the relationship between the problem features and the algorithm performance. The basis for the implemented supervised learning approach is data which were collected with the help of simulation. This model is validated with the grouped cross validation method and additionally evaluated by algorithm

performance prediction for unseen problem instances. The model and its evaluation are outlined and discussed in Section 4. Our introduced DVRP features show their validity as base for autonomous algorithms selection. Finally, conclusions are drawn in Section 5.

1 Related Work

This Section introduces the DVRP and the ASP and outlines current research about algorithm selection in the domain of vehicle routing. Additionally common location-based features, used in algorithm selection and performance evaluation studies for the TSP and the VRP, are discussed.

1.1 The Dynamic Vehicle Routing Problem

The DVRP was first introduced in [6] as an extension of the VRP, with a description of a computer-controlled Dial-A-Ride system in Rochester, NY (USA). The VRP, a generalization of the TSP, is defined on a graph $G = (V, E, C)$, where $V = \{v_0, \dots, v_n\}$ is a set of nodes. The set $E = \{(v_i, v_j) | (v_i, v_j) \in V^2; i \neq j\}$ defines the neighborhood of the nodes V as edge set. $C = (c_{ij})_{(v_i, v_j) \in E}$ defines a cost matrix over E . Nodes are usually customer demands, characterized by a location, which needs to be serviced. The DVRP extends the VRP notion by adding a point in time $(t_i)_{v_i}$ for all v_i , that defines when the customer request appears. In a VRP instance all $(t_i)_{v_i}$ for all v_i would be 0. In a dynamic setting (DVRP) at least one $(t_i)_{v_i}$ would be greater than 0. The TSP, VRP, and DVRP have in common, that the main characteristics of a node $v_i \in V$ is the location.

1.2 The Algorithm Selection Problem

The ASP is formalised and introduced by Rice in [13]. Figure 1 visualises the framework for the ASP proposed in [13]. It aims to predict the best algorithm performance for a problem instance, based on measurable features, [18]. The following four main components are part of the framework.

- the problem space P contains a set of problem instances;
- the feature space F represents a set of measurable features, which can be extracted from all problem instance in P ; the feature extraction process has to be less time-consuming compared to solving the

problem instances with algorithms from the algorithm space A ;

- the algorithm space A contains a portfolio of algorithms which are able to solve the problem instances from P ;
- the performance space Y describes a mapping from algorithm result to a performance metric.

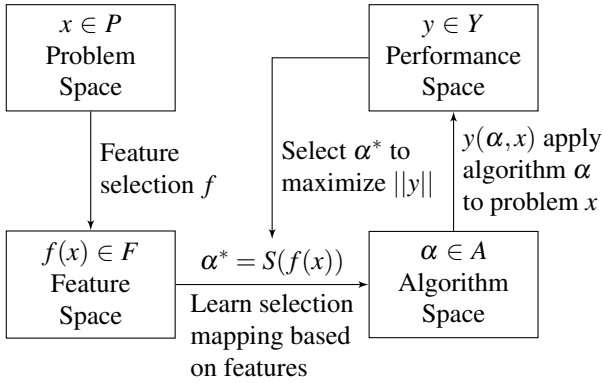


Figure 1: A framework for the Algorithm Selection Problem (ASP) introduced from Rice in [13].

The main challenge of the ASP is to find a selection mapping $S(f(x))$ into the algorithm space A for a problem instance $x \in P$ with feature vector $f(x)$. The mapping has to maximise the performance metric $\|y\|$ for $y(\alpha, x)$, where α is an algorithm in A . A broad discussion on algorithm selection across a variety of disciplines is given in the survey paper [19].

In the domain of static vehicle routing, [20] investigates the TSP difficulty by learning from evolved instances. The work uses an evolutionary algorithm to evolve TSP instances which are intentionally easy or hard for algorithms based on the Lin-Kernighan heuristic method [21]. Features are derived from the problem instances and the impact of these features for the difficulty of the algorithms is investigated. [22] investigates the success of 2-opt based local search algorithms for solving the TSP and shows important features that make problem instances hard or easy for 2-opt approaches. [23] studies algorithm selection for the Traveling Thief Problem (TTP) based on TSP features. The TTP is a combination of the TSP and the Knapsack Problem (KP). Earlier work focuses on the impact of problem features for the problem difficulty in general. For example, [24] demonstrates that the variance of the distance matrix correlates with the TSP difficulty for exact

solving approaches. [25] shows that this is also true for heuristic solving approaches.

1.3 Problem Features

The algorithm selection and performance evaluation studies in the domain of static vehicle routing are usually based on features involving the location of the customer demands. Beside the number of customer demands, [14] classifies the 47 available VRP features, used in their study, into the following eight groups.

- *Distance Features* summarise features which involve the costs of the customer demands. The cost of a demand is the distance from the location of the customer to the depot. Considered are the lowest, highest, mean and median costs of all customer demands. Additionally statistics about the distance matrix, like distinct distances, or the standard deviation of the matrix are categorised in this group.
- *Mode Features* group all features concerning the mode of the customer demand cost distribution, [26]. For all customer the costs to all other customers are calculated. The mode is the cost value which appears most often. Additionally the frequency, quantity and mean of the mode values are considered. [14] additionally groups the number of modes of the customer demand cost distribution introduced in [27] in this category.
- *Cluster Features* bundle features based on statistically information extracted from the results of a clustering algorithm, like the number of clusters or the mean distance of the customer demands to the cluster centroid.
- *Nearest Neighbour Distance Features* group features derived from the calculated nearest-neighbour distances for each customer demand, like the minimum, maximum, mean, or median.
- *Centroid Features* summarise all features calculated dependent on the centroid of all customer demands, including the coordinates of the centroid, minimum, maximum, mean and median of the distances between customer demands and centroid.
- *MST Features* bundle features based on characteristics of the calculated Minimum Spanning Tree (MST). Considered are the minimum, mean, median, maximum and standard deviation of the depth and distance of the MST.

- *Angle Features* bundle features based on the calculated angle between the two nearest neighbour customer demands, like the minimum, mean, median, maximum and standard deviation of all calculated angles.
- *Convex Hull Features* group the features concerning the convex hull from the problem instance, like the area of the hull and the fraction of customer demands which are part of the hull.

Problem features for the DVRP are rarely described in the literature and are usually discussed independently from algorithm selection. [28] introduced the Degree of Dynamism (DOD), a measure of the dynamism of a dynamic routing problem. The DOD describes the ratio between dynamic customer demands and the sum of dynamic and static customer demands. The DOD is similar to the feature *Number of customer demands* in the context of static vehicle routing. [29] developed a framework based on the DOD. The framework classifies weak, moderate and strong dynamic systems and recommends solution approaches for these classes of dynamic routing problems. [29] also introduced the Effective Degree of Dynamism (EDOD) as extension of the DOD. The EDOD considers the planning horizon T for the calculation of the measure of the dynamism of a routing problem. [29] also studies the relation between DOD, EDOD and routing costs for the Partially Dynamic Traveling Repairman Problem (PDTRP). In [15] we introduced the Location-based Degree of Dynamism (LDOD) capturing the location of the dynamic customer request. We show that there is a positive correlation between our proposed LDOD and the resulting DVRP solution quality (performance) for a Greedy and a Re-planning algorithm. The LDOD is the only feature that focuses on the location of the dynamic customer requests. We categorise the LDOD in the group of *Distance Features*. The LDOD is the only DVRP instance feature that is based on the fundamental characteristic location of a node (customer request). A fundamental problem characteristic is an inherent property, i.e., it is applicable regardless of the problem variation, compare [15].

2 New Features for the Dynamic Vehicle Routing Problem

In this Section DVRP features derived from static VRP location-based features, which are building the Feature

Space F , are presented and evaluated. The generation of the Problem Space P , Performance Space Y and Algorithm Space A are also discussed within this Section.

2.1 Feature Space

Since problem features for the DVRP are rarely described in the literature, compare Section 2.3, we had to introduce new problem features. The TSP, the VRP and the DVRP are all defined on a graph G containing a set of nodes $V = \{v_0, \dots, v_n\}$, compare Subsection 2.1. The main characteristic of a node v_i is its location. Based on this consideration, we are able to base our location-based features for the DVRP on location-based features introduced for the TSP. In general there are two main groups of features we consider to characterise a DVRP problem instance. In the first group we classify existing TSP features which we applied to the dynamic customer demands only. Here we consider *Angle Features*, *MST Features*, *Nearest Neighbour Distance Features*, *Distance Features*, and *Cluster Features*. Features from the second group try to capture the dynamism of the problem instance, by considering both, dynamic and static customer demands. Known examples for the second group are the DOD ([28]) and the LDOD ([15]). For our research we transferred *Angle Features*, and *Nearest Neighbour Distance Features* discussed in Section 3 into features considering dynamic and static demands. Therefore we calculated the distances and angles between dynamic and nearest neighbour static customer demands, compare Figure 2. Since a route to a dynamic customer demand will likely not start at the nearest neighbour static customer, we considered neighbourhoods with size of 2, 3, and 5. For all neighbourhoods we determined features considering the sum, mean, median and maximum value. For example for the neighbourhood of 2 we calculated the distances a and b , compare Figure 2. For the distances we calculated the values $sum(a,b)$, $mean(a,b)$, $median(a,b)$, and $max(a,b)$ for all dynamic customer demands. The resulting features are the sum, maximum, mean, median, standard deviation, and variance coefficient of these calculated values.

We also considered *Centroid Features* and transferred them into features considering dynamic and static customer demands. To this end, we calculated the centroid of the instance including both dynamic and static customer demands. We determined the average and the sum distance from dynamic and the average distance from all static customer demands to the cen-

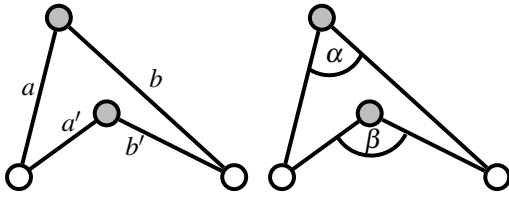


Figure 2: Distances and angles between dynamic customer demands (grey coloured circles) and nearest neighbour static customer demands (white coloured circles).

centroid. The resulting feature is the ratio between these two distances. The last feature we introduced is considering the ratio between the surface areas spanned by all dynamic and all static customer demands. Overall we determined 184 different DVRP features.

2.2 Problem Space

To investigate the suitability of our introduced features for algorithm selection we had to create a problem space P and a corresponding performance space Y with algorithms $\alpha \in A$, compare Figure 1. We investigated the suitability of our introduced features with the most basic version a DVRP. We are not considering any capacity constraint or any restrictions regarding the customer requests like time windows or service times. We are focusing on problem instances with one depot and one vehicle only. To expand the problem space as wide as possible, we applied the general purpose metaheuristic optimisation framework SEREIN introduced in [16] to derive dynamic from static VRP instances. Figure 3 visualises the four static VRP instances from [30], which we considered for our research. They have either uniformly distributed (CMT01, CMT04) or clustered (CMT11, CMT12) customer requests.

By deriving dynamic from static problem instances three decisions have to be made. First, how many customer requests shall be dynamic? Second, which requests exactly shall be dynamic? Third, when the customer demand appears during the time horizon? The number of dynamic customer demands is determined by the DOD. In our research the following $DOD \in \{0.1, 0.2, \dots, 0.9\}$ are considered. The second decision is made by SEREIN. It selects the determined number of dynamic requests out of all available request. The genetic algorithm evaluates the performance (solution costs) of the created instance with the help of the

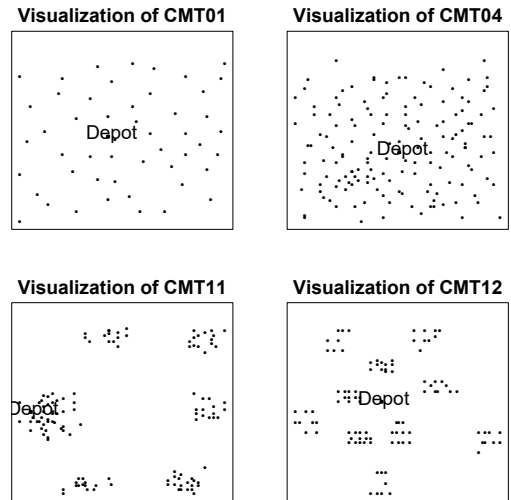


Figure 3: Subset of static VRP instances introduced in [30], considering equal distributed and clustered customer demands (black dots).

RVRP Simulator introduced in [17]. The RVRP Simulator provides a set of standard algorithms to handle the dynamic requests. For the problem instance generation we choose a simple Greedy algorithm provided in the simulator. For a more detailed description of the implemented algorithm see the following Section. The solution costs are used to navigate through the search space. The main task of SEREIN is to create dynamic problem instances where the Greedy algorithm performs very good (low solution costs, easy problem instances for the Greedy algorithm) and also instances where the algorithm performs very bad (high solution costs, hard problem instances for the Greedy algorithm). The third decision, when the dynamic requests appear, is not made by the genetic algorithm. Currently, we intend to minimise the influence of the time t_i when request i appears. It is obvious that requests appearing very late are probably producing higher routing costs. For example, if all dynamic requests appear when the vehicle finished its initial route planned for the static customer requests, the employed algorithm cannot integrate the dynamic requests into the existing route. Every time a dynamic request appears, the vehicle has to start from the depot to service the customer, which results in high routing costs. To prevent this interdependency, we partly follow the approach introduced in [15] and determined the time t_i for a dynamic request i as evenly distributed between 0 and the time horizon T . So, for all created points in time t_i for all dynamic requests n the following

rule is valid: $t_{i+1} - t_i = d_i; d_i = d_{i+1} \forall i$, where $t_{n+1} = T$. T is determined by solving the static instance with the Jsprit framework [31]. To prevent the interdependency between the location of request i and the time t_i , we executed the simulation to evaluate the dynamic problem instance several times with different assignments between t_i and i . Additionally, we simulated the instances in a reversed routing order. For example, if a routing starts with servicing the static customer A, B , and ends at static customer C . Additional simulations in reversed order where the routing starts at customer C and ends with servicing customer A are performed. To generate one dynamic problem instance using SEREIN, we had to determine 6 sample routing costs with the help of simulation. The average of these routing costs is used by SEREIN to unfold the search space, identifying instances which are easy as well as instances which are hard to solve for the Greedy algorithm. SEREIN created 2,000 dynamic problem instances for each combination of $DOD \in \{0.1, 0.2, \dots, 0.9\}$ and problem instance shown in Figure 3. This results in 72,000 dynamic problem instances which got evaluated with 432,000 simulations, only to create the problem space Y .

2.3 Algorithm and the Performance Space

For the Greedy algorithm, we already evaluated the Performance Space Y while creating the Problem Space P . The Greedy algorithm inserts a dynamic request, appearing during problem simulation, into the current planned route for the vehicle. We determined the initial route with the help of the Jsprit framework [31]. The Greedy algorithm evaluates all possible insertion points and chooses the one that produces the best solution costs. A detailed description of the algorithm can be found in [15]. Since we need at least two algorithms to evaluate our introduced DVRP features. We implemented a replanning approach, where every new dynamic request triggers a complete replanning of the tour. So both considered solution approaches are following very contrary ideas. The Re-planning algorithm, that we implemented uses the R package *tspmeta* from [14]. The implementation is based on a 2-opt optimisation algorithm, which was one of the first successful algorithms to solve larger TSP instances and which is still widely used in practice [14]. Due to the stochastic characteristics of the 2-opt approach, we had to simulate each dynamic problem instance (72,000) several times. Overall, we used additional 432,000 simulations

(also considering simulations in reverse customer request order) to determine the Performance Space Y for the Re-planning algorithm.

2.4 Evaluation of the new DVRP Features

We separate the generated problem instances into a group of instance which are easy (low solution costs) and into a group of instances which are hard (high solution costs) to solve for each algorithm. We calculate the feature vectors for all instances and evaluate all possible feature combinations and their predictive power to differentiate between hard and easy instances. In this Section we present an exemplary excerpt from our evaluations.

Figure 4 shows the relation between feature *angle_mean* and *mst_dists_mean* from the first group of features (features considering dynamic customer demands only). The graphs in the first row show results derived from the problem instance CMT04. The suitability of the features to distinguish between easy and hard instances is visible by optically separable point clouds for all DOD's. With an increasing DOD, the point clouds are harder to separate from each other. The graphs in the second row consider all instances (compare Figure 3). Here, we observe more point clouds which are harder to separate visually. The borders between the clouds get more indistinct with a decreasing DOD. Similar patterns occurred for other feature combinations from the first group. The evaluation of the features from the second group (features considering static and dynamic customer demands) show similar characteristics, compare Figure 5. But in general there are better and less visually separable point clouds even for higher DOD's. The combination of features of both groups show its suitability to distinguished between hard and easy problem instance too, compare Figure 6.

In general the feature evaluation shows, that the introduced features are suitable to distinguish between easy and hard problem instances for both applied algorithms. Note, that an important feature to distinguish between instances is the DOD, only implicitly shown in the graphs.

3 Algorithm Selection

To test our problem features in the context of Algorithm Selection, we implemented a regression machine

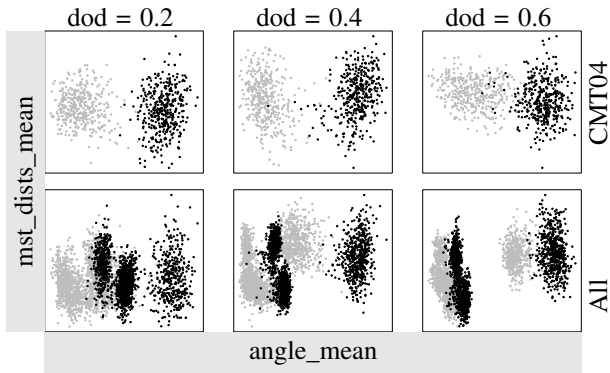


Figure 4: Evaluation of feature combination *angle_mean* and *mst_dists_mean* for Re-planning algorithm.

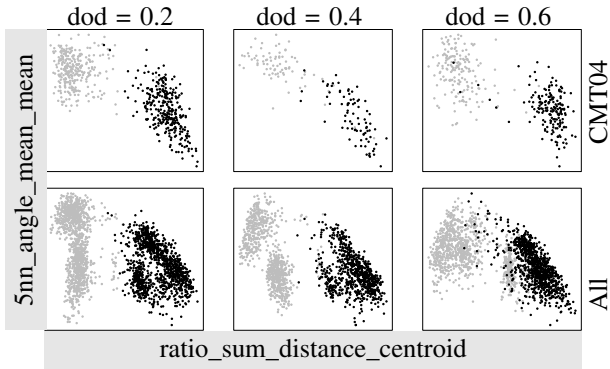


Figure 5: Evaluation of feature combination *ratio_sum_distance_centroid* and *5nn_angle_mean_mean* for Re-planning algorithm.

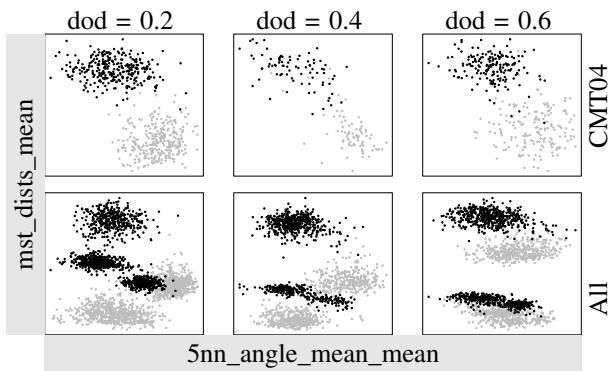


Figure 6: Evaluation of feature combination *mst_dists_mean* and *5nn_angle_mean_mean* for Re-planning algorithm.

learning technique to model the relationship between the problem features and the algorithm performance. We used Artificial Neural Network (ANN) which map unlabeled input to a label (output) using internal data structures. We standardise all values of the feature vectors using the standard scalar method implemented in [32]. These standardised feature vectors are the input for the model (balanced regrading instances and DOD). The output for the regression model is the standardised routing costs for the instances. For each algorithm (Greedy, Re-planning) we constructed one regression model. Due to its popularity we used Tensorflow from Google Brain introduced in [33] for the implementation. Both ANNs have 4 hidden layers with 60, 18, 20, and 22 neurons. The input layer has 184 neurons and the output layer has 1. All neurons are using the activation function RELU, [34]. The structure of the model was setup using systematic trial and error, a common approach to approximate the optimal number of hidden layers and nodes, [35]. All neurons of each layer are fully connected to all neurons of the following layer. We applied the optimisation function *Adam* introduced in [36] to determine the weights of the edges between the neurons of the ANNs. Beside testing our approach with unseen problem instance, we applied the grouped cross validation method to validate our trained ANNs. The method divides the available data into k groups and constructs k different ANNs using $k - 1$ data groups. The k^{th} group is used for testing the constructed ANN, [37]. The error is determined using the mean of testing set errors of the groups. The grouped cross validation method is currently considered as the gold standard for testing ANNs. Note, that we had to ensure that the combination of instance and DOD are evenly distributed within all created groups. For our validation we defined 5 data groups. The results of the validation of our constructed ANNs are shown in Table 1. The applied metric for the regression model is the mean squared error (MSE), see for example, [38]. The results in Table 1 show that our features are very suitable for performance prediction. The average error for both ANNs is less than half a percent for all cross validation groups. The models are not over-fitted and precisely predict the algorithm performance, even for unseen dynamic instances generated from the static instances shown in Figure 3.

To test our regression models with completely unseen problem instances, we created dynamic instances from a subset of VRP instances introduced in [39] and [40], see Figure 7. We created 500 random instances

Table 1: Grouped cross validation results (5 groups) of our constructed ANNs.

Algorithm	Mean square error (+/- Standard deviation)
Greedy	0.006775 (+/- 0.000126)
Re-planning	0.003549 (+/- 0.000105)

for each instance for each $DOD \in \{0.3, 0.4, \dots, 0.8\}$ and determined the routing costs with the Greedy and the Re-planning algorithm. The sum of the routing costs for each algorithm, $R = \sum_{i=0}^n r_{algorithm,i}$, is shown in Table 2 (column Greedy and Re-planning). Based on the known routing costs for the Greedy and for the Re-planning algorithm, we have been able to determine the correct algorithm selection for each problem instance. The correct selected algorithm is the one which produces the lower costs. The sum of the routing costs, based on a correct algorithm selection, $Correct = \sum_{i=0}^n \min(r_{greedy,i}, r_{replan,i})$, is also shown in Table 2 (column Correct). Additionally Table 2 shows the sum of the routing costs based on our algorithm selection with our regression (column Regression) models. In general the results, shown in Table 2, illustrate that we are able to successfully select algorithms for unseen problem instances with our approach. Our algorithm selection approach performs better, compared to exclusive use of either the Greedy or the Re-planning algorithm in most cases. However, for the dynamic problem instances derived from problem instance A-n60-k9, our algorithm selection leads to worse results in comparison to the pure Re-Planning algorithm.

4 Conclusion and Outlook

Our results show that our approach is able to reliably autonomously select the better performing algorithm for various known and partly also for unseen DVRP instances. Based on simulation results we have been able to derive characteristic DVRP from static problem instances. We developed and calculated features for these instances. Our introduced features in combination with the simulation results are the bases for our developed regression models which we trained to predict algorithm performance. The prediction of algorithm performance is the base for our successful autonomous algorithm selection.

For our performance prediction models we are currently considering only 4 different structured problem

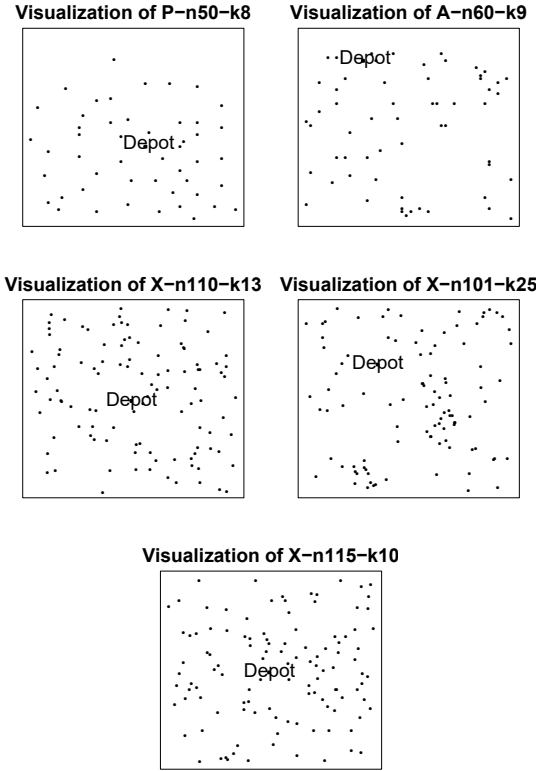


Figure 7: Subset of VRP instances introduced in [39] (P-n50-k8, and A-n60-k9) and [40] (X-n110-k13, X-n101-k25, and X-n115-k10). The dots represent the customer requests.

instances, compare Figure 3. The trained regression models are quite fitted to these instances which seems to be the reason for the difficulties with performance prediction for the unseen problem instances A-n60-k9, compare Table 2. The idea is to build a stronger basis for our performance prediction by considering more different kinds of problem instances. We plan to include the very geometrical structured VRP instances introduced in [41] in our research. We also have to evaluate the possibility to create own artificial instances.

Currently we only consider two different algorithms which perform similar for different problem instances, compare Section 4. Future work will be the training and testing of regression models based on simulation results of different DVRP algorithms. We will implement an DVRP algorithm portfolio with existing algorithms. We are currently developing a new DVRP solving approach based on the results from [15]. Beside considering more instances and more algorithms, the question occurs which evolved instances are good for training

Table 2: Validation results of our constructed ANNs for unseen instances.

3000 Instances based on	Greedy	Re-planning	Correct	Regression
X-n110-k13	31,664,462	31,836,229	31,405,740	31,413,135
X-n101-k25	34,794,388	34,725,471	34,389,425	34,455,935
X-n115-k10	39,754,906	39,870,583	39,402,544	39,429,201
P-n50-k8	534,005	534,690	528,157	533,961
A-n60-k9	2,787,325	2,781,479	2,752,555	2,783,146

and testing purposes. Currently we are not verifying if an evolved instances is a good representation for the performance class. In future work, we have to determine a metric which provides information about how good a certain instance represents a certain algorithm performance.

References

- [1] Dantzig GB, Ramser JH. The Truck Dispatching Problem. *Management science*. 1959;6(1):80–91.
- [2] Flood MM. The traveling-salesman problem. *Operations Research*. 1956;4(1):61–75.
- [3] Lahyani R, Khemakhem M, Semet F. Rich Vehicle Routing Problems: From a Taxonomy to a Definition. *European Journal of Operational Research*. 2015; 241(1):1–14.
- [4] Pillac V, Gendreau M, Gu ret C, Medaglia AL. A Review of Dynamic Vehicle Routing Problems. *European Journal of Operational Research*. 2013; 225(1):1–11.
- [5] Psaraftis HN, Wen M, Kontovas CA. Dynamic vehicle routing problems: Three decades and counting. *Networks*. 2016;67(1):3–31.
- [6] Wilson NH, Colvin NJ. *Computer Control of the Rochester Dial-a-ride System*. 77-22. Cambridge, Massachusetts. 1977.
- [7] Ritzinger U, Puchinger J, Hartl RF. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*. 2016; 54(1):215–231.
- [8] Clarke G, Wright JW. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations research*. 1964;12(4):568–581.
- [9] Juan AA, Faulin J, Jorba J, Riera D, Masip D, Barrios B. On the use of Monte Carlo Simulation, Cache and Splitting Techniques to Improve the Clarke and Wright Savings Heuristics. *Journal of the Operational Research Society*. 2011;62(6):1085–1097.
- [10] Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G. A Review of Simheuristics: Extending Metaheuristics to Deal with Stochastic Combinatorial Optimization Problems. *Operations Research Perspectives*. 2015; 2:62–72.
- [11] Gruler A, Kl uter A, Rabe M, Juan AA. A Simulation-Optimization Approach for the Two-Echelon Location Routing Problem Arising in the Creation of Urban Consolidation Centres. In: *Simulation in Produktion und Logistik 2017*, edited by Wenzel S, Peter T. Kassel: kassel university press. 2017; pp. 129–138.
- [12] Wolpert DH, Macready WG. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*. 1997;1(1):67–82.
- [13] Rice JR. The Algorithm Selection Problem. *Advances in Computers*. 1976;15:65–118.
- [14] Mersmann O, Bischl B, Trautmann H, Wagner M, Bossek J, Neumann F. A Novel Feature-based Approach to Characterize Algorithm Performance for the Traveling Salesperson Problem. *Annals of Mathematics and Artificial Intelligence*. 2013;69(2):151–182.
- [15] Mayer T, Uhlig T, Rose O. A Location Model for Dynamic Vehicle Routing Problems. In: *Simulation in Produktion und Logistik 2017*, edited by Wenzel S, Peter T. Kassel: kassel university press. 2017; pp. 149–158.
- [16] Uhlig T. *Self-Replicating Individuals*. M nchen: Verlag Dr. Hut. 2015.
- [17] Mayer T, Uhlig T, Rose O. An open-source discrete event simulator for rich vehicle routing problems. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, edited by Rossetti R, Wolf D. New York, USA: IEEE. 2016; pp. 1305–1310.
- [18] Smith-Miles K, Lopes L. Measuring Instance Difficulty for Combinatorial Optimization Problems. *Computers & Operations Research*. 2012;39(5):875–889.
- [19] Smith-Miles KA. Cross-disciplinary Perspectives on Meta-learning for Algorithm Selection. *ACM Computing Surveys (CSUR)*. 2009;41(1):6.

- [20] Smith-Miles K, van Hemert J, Lim XY. Understanding TSP Difficulty by Learning from Evolved Instances. In: *Proceedings of the 4th International Conference on Learning and Intelligent Optimization*, edited by Blum C, Battiti R. Berlin, Heidelberg: Springer-Verlag. 2010; pp. 266–280.
- [21] Lin S, Kernighan BW. An Effective Heuristic Algorithm for the Traveling-salesman Problem. *Operations Research*. 1973;21(2):498–516.
- [22] Mersmann O, Bischl B, Bossek J, Trautmann H, Wagner M, Neumann F. Local Search and the Traveling Salesman Problem: A Feature-Based Characterization of Problem Hardness. In: *Learning and Intelligent Optimization*, edited by Hamadi Y, Schoenauer M, pp. 115–129. Berlin, Heidelberg: Springer Berlin Heidelberg. 2012;.
- [23] Wagner M, Lindauer M, Misir M, Nallaperuma S, Hutter F. A Case Study of Algorithm Selection for the Traveling Thief Problem. *Journal of Heuristics*. 2018; 24(3):295–320.
- [24] Cheeseman P, Kanefsky B, Taylor WM. Where the Really Hard Problems Are. In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1*, edited by Mylopoulos J, Reiter R. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 1991; pp. 331–337.
- [25] Ridge E, Kudenko D. An Analysis of Problem Difficulty for a Class of Optimisation Heuristics. In: *Proceedings of the 7th European Conference on Evolutionary Computation in Combinatorial Optimization*, edited by Cotta C, Van Hemert J. Berlin, Heidelberg: Springer-Verlag. 2007; pp. 198–209.
- [26] Kanda J, Carvalho A, Hruschka E, Soares C. Selection of algorithms to solve traveling salesman problems using meta-learning 1. *International Journal of Hybrid Intelligent Systems*. 2011;8(3):117–128.
- [27] Mersmann O, Bischl B, Trautmann H, Preuss M, Weihs C, Rudolph G. Exploratory landscape analysis. In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM. 2011; pp. 829–836.
- [28] Lund K, Madsen OB, Rygaard JM. Vehicle Routing with Varying Degree of Dynamism. 1996; (IMM-REP-1996-1).
- [29] Larsen A. The Dynamic Vehicle Routing Problem. Ph.D. thesis, Institute of Mathematical Modelling, Technical University of Denmark. 2000.
- [30] Christofides N. The Vehicle Routing Problem. *Revue française d'automatique, informatique, recherche opérationnelle Recherche opérationnelle*. 1976; 10(V1):55–70.
- [31] Schröder. Jsprit. <https://github.com/graphhopper/jsprit>. 2018. Accessed July 2nd, 2018.
- [32] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: Machine learning in Python. *Journal of machine learning research*. 2011; 12(11):2825–2830.
- [33] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, Zheng X. TensorFlow: A System for Large-scale Machine Learning. In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16*. Berkeley, CA, USA: USENIX Association. 2016; pp. 265–283.
- [34] Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, edited by Fürnkranz J, Joachims T. USA: Omnipress. 2010; pp. 807–814.
- [35] Basheer IA, Hajmeer M. Artificial Neural Networks: Fundamentals, Computing, Design, and Application. *Journal of microbiological methods*. 2000;43(1):3–31.
- [36] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 2015;.
- [37] Twomey JM, Smith AE. *Validation and Verification*. New York, NY, USA: American Society of Civil Engineers. 1997.
- [38] Lehmann EL, Casella G. *Theory of Point Estimation*. 175 Fifth Avenue, New York: Springer Science & Business Media, 2nd ed. 2006.
- [39] Christiansen CH, Lysgaard J. A Branch-and-price Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands. *Operations Research Letters*. 2007;35(6):773–781.
- [40] Uchoa E, Pecin D, Pessoa A, Poggi M, Vidal T, Subramanian A. New Benchmark Instances for the Capacitated Vehicle Routing Problem. *European Journal of Operational Research*. 2017; 257(3):845–858.
- [41] Golden BL, Wasil EA, Kelly JP, Chao IM. The Impact of Metaheuristics on Solving the Vehicle Routing Problem: Algorithms, Problem Sets, and Computational Results. In: *Fleet Management and Logistics*, edited by Crainic TG, Laporte G, pp. 33–56. Boston, MA: Springer US. 1998;.

Hybride Modellbildung in Verbindung mit neuronalen Netzen

Stefanie Winkler^{1*}, Andreas Körner¹, Felix Breiteneker¹

¹Institut für Analysis und Scientific Computing, Technische Universität Wien, Wiedner Hauptstraße 8-10, 1040 Wien, Österreich; *stefanie.winkler@tuwien.ac.at

Abstrakt. Dieser Betrag behandelt die möglichen Anwendungen von neuronalen Netzen in der hybriden Modellbildung. Nach einer grundlegenden Einführung in beide Modellierungsansätze werden mögliche Applikationen diskutiert. Einerseits steht die Ersetzbarkeit von hybriden Modellen zur Debatte. Diese Frage wird im Folgenden Paper ansatzweise behandelt. Andererseits geht es auch um die mögliche Kombination von neuronalen Netzen mit physikalischen Modellen, z.B. zur Bestimmung einzelner Parameter. Anhand eines akademischen Beispiels werden die unterschiedlichen Zugänge angewandt und verglichen.

Einführung

Die Attraktivität von neuronalen Netzen hat in den letzten Jahrzehnten in vielen Forschungsgebieten stark zugenommen. Vor allem in den Bereichen Machine Learning und Deep Learning zählen sie zu den wichtigsten Werkzeugen. Aus der Forschungsarbeit im Gebiet der künstlichen Intelligenz sind diese nicht mehr wegzudenken. Die Anzahl der zu verarbeitenden Datensätze und deren Umfang wächst stetig. Dank der flexiblen Struktur und der breiten Anwendungsmöglichkeiten von neuronalen Netzen eignen sie sich optimal, um effizient und ohne großen Modellbildungsaufwand Datenmengen zu analysieren und Vorhersagen über das Modellverhalten zu treffen. Neuronale Netze werden demnach hauptsächlich verwendet, um große Datensätze zu verarbeiten.

Jedoch gibt es natürlich auch Modelle, die sich durch mathematische oder physikalische Gesetze beschreiben lassen, wie zum Beispiel im Bereich der hybriden Modellierung. Bei diesen Modellen geht es darum komplexe Strukturen und Abläufe unter Zuhilfenahme mehrerer Modellierungsmethoden zu realisieren. Das zugrundeliegende Problem erfordert eine Betrachtung unterschiedlicher mathematischer Rahmenbedingungen, welche in getrennten Modellen abgebildet werden müssen. Die Unterschiede in den Modellen reichen von einfachen Parameterwechseln bis hin zu komplexen Strukturänderungen. Ein einfaches Beispiel hierfür ist das mathematische Pen-

del. Das Fadenpendel kann in zwei unterschiedliche Bewegungen unterteilt werden, der Pendelbewegung und dem freien Fall. In diesem Modell handelt es sich nicht um einen Parameterwechsel, sondern um eine Änderung der mathematischen Beschreibung.

Im Folgenden werden die unterschiedlichen Modellierungsmethoden im Überblick erklärt und anhand eines einfachen Anwendungsbeispiels angewandt. Dabei werden die Verschiedenen Implementierung beschrieben. Im Folgenden wird versucht mögliche Rückschlüsse auf allgemeine Hybride Modelle zu ziehen.

1 Modellstrukturen

Die Grundlagen der angesprochenen Methoden werden hier überblicksmäßig thematisiert. Die Struktur und der Aufbau beider Modellierungsansätze und die verwendeten Parameter werden vorgestellt.

1.1 Hybride Modellierung

Hybride Modelle werden sowohl für die Beschreibung diskreter, als auch kontinuierlicher, vor allem aber für die Kombination beider verwendet. Eine mögliche Darstellung solcher Modelle ist in Abbildung 1 gegeben.

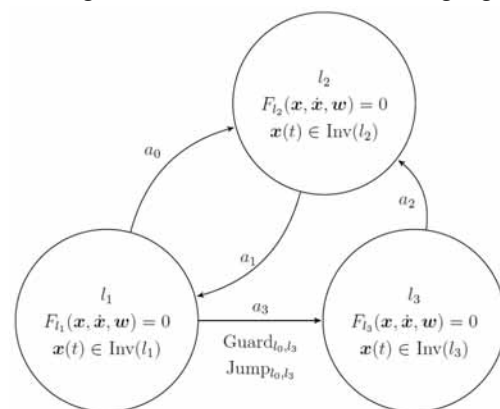


Abbildung 1: Jeder dieser Knoten symbolisiert eines der Submodelle. Die Gesamtheit aller Knoten und Verbindungen stellt das hybride System dar. [1]

Diese Grundstruktur aus Abbildung 1 beinhaltet unterschiedlichen Teilmodelle, durch Kreise visualisiert, inklusive ihrer mathematischen Beschreibung. In dieser Abbildung entsprechen die Beschreibung Differentialalgebraischer Systeme. Die Verbindungslinien zwischen den unterschiedlichen Teilmodellen beschreiben die Bedingungen des jeweiligen Systemwechsels. Zurückkehrend zu dem oben erwähnten Beispiel des Fadenpendels, kann man den Übergang zwischen den beiden Zuständen, dem freien Fall und der Pendelbewegung, durch eine physikalische Bedingung charakterisieren. Ist diese Bedingung nicht mehr erfüllt, verlässt das Pendel beispielsweise die Kreisbahn und der freie Fall setzt ein. Die Phase des freien Falls kann beim Erfüllen der gegenteiligen Bedingung an die Kraft wieder in die typische Pendelbewegung übergehen, das Koordinatensystem wird abermals gewechselt.

Man könnte diese Bedingung auch als eine Weiche interpretieren, die den aktuellen Zustand des Systems kennt und die notwendigen, meist diskreten Bedingungen, überprüft und gegebenenfalls die Umschaltung zum passenden Modell in die Wege leitet. Der Modellwechsel kann einerseits eine Änderung einzelner Parameterwerte andererseits eine Änderung des Zustandsvektors inklusive Modellbeschreibung nach sich ziehen. Das Beispiel des Fadenpendels gehört zu jenen hybriden Modellen, in denen sich die physikalische Beschreibung des Verhaltens ändert. In Abschnitt 2 wird ein Beispiel vorgestellt, dessen Modell durch eine Parameteränderung charakterisiert wird.

Im Grunde kann die Struktur hybrider Modelle noch oberflächlicher definiert werden. Jeder einzelne Knoten aus Abbildung 1, stellt ein System da, dass mit Hilfe beliebiger Methoden, wie z.B. ODEs, DAEs, DEVS, DESS, etc., ein Verhalten beschreibt. Der Wechsel zwischen den Systemen wird durch unterschiedlichste Bedingung definiert. Zumeist sind diese Bedingungen eindeutig und werden über bestimmte Zeitpunkte, mit mathematischen Gleichungen oder als Mengen beschrieben.

1.2 Neuronale Netze

Im Gegensatz zu hybriden Systemen handelt es sich bei neuronalen Netzen (NN) nicht um eine physikalische oder mathematische Beschreibung eines Systems, sondern vielmehr um eine Nachahmung der Informationsverarbeitung im menschlichen Gehirn. Hier arbeiten unzählige Neuronen zusammen, um einströmende Informa-

tionen effizient zu verarbeiten. Auf diesem Konzept aufbauend gestalten sich die neuronalen Netze ebenfalls als Netzwerk von Knotenpunkten – den Neuronen – und ihren Verbindungen. Jedes neuronale Netz besteht im Wesentlichen aus drei Komponenten, sogenannten Schichten: einer Eingangsschicht, einer oder mehrerer Neuronenschichten und der Ausgangsschicht. Die Struktur eines neuronalen Netzes wird maßgeblich durch die Anzahl der Neuronenschichten sowie der Anordnung ihrer Verbindungslinien bestimmt. Die einfachste Darstellung so eines Modells ist in Abbildung 2 zu finden.

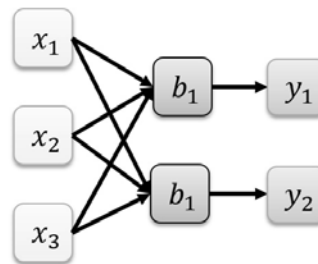


Abbildung 2: Schematische Darstellung eines einfachen neuronalen Netzes.

Hier bilden die grünen Boxen mit x_1, x_2, x_3 die Eingangsschicht mit den 3 Inputwerten. Die mittlere graue Schicht ist die Neuronenschicht mit zwei Neuronen b_1, b_2 . Schlussendlich bilden die orangenen Boxen y_1 und y_2 die Ausgangsschicht. In dieser Betrachtung werden die drei Eingangssignale auf zwei Outputwerte reduziert. Gleichung (1) beschreibt wie der Ausgang y_1 zu Stande kommt, wobei $w_{i1}, i \in \{1,2,3\}$ das Gewicht der Verbindung von x_i zu b_1 definiert.

$$y_1 = \sum_{i=1}^3 x_i \cdot w_{i1} + b_1 \quad (1)$$

Im Gegensatz zur mathematischen Beschreibung hybrider Modelle kann man die Parameter für ein neuronales Netz nur mit Hilfe von hinreichend vielen gegebenen Daten bestimmen. Die Verbindungslinien definieren hierbei welche Eingangsdaten in Zusammenhang gebracht werden. Jede der Verbindungslinien bekommt ein bestimmtes Gewicht zugeordnet. Die jeweiligen Eingangsdaten werden beim Durchlauf mit diesen Gewichten multipliziert und können somit, abhängig vom Wert, eine abschwächende bzw. verstärkende Wirkung haben. Die Struktur des neuronalen Netzes beeinflusst dabei, wie viele zu bestimmende Gewichte benötigt werden.

Die Werte der Gewichte werden durch die gegebenen Input- und Outputdaten bestimmt. Die Anzahl an verfügbaren Datensätzen beeinflusst die Genauigkeit der Vorhersagen, die ein neuronales Netz über das Verhalten des betrachtenden Systems machen kann. Zur Bestimmung der Gewichte werden sogenannte Trainingsalgorithmen ausgeführt. Dabei werden die Gewichte anfangs auf zufällige Werte festgesetzt. Anschließend läuft das neuronale Netz mit diesen und den gegebenen Eingangsdaten durch. Mit Hilfe des sogenannten Backpropagation Algorithmus wird der Fehler schrittweise vom Ausgangssignal zurückgerechnet und die Werte der Gewichte angepasst. Dies wiederholt sich solange, bis der Fehler klein genug ist bzw. sich nicht mehr verbessert.

Neuronale Netze werden daher meist verwendet, um aus gegebenen Datensätzen verschiedene Eigenschaften herauszufiltern. Eine, in diesem Zusammenhang spannendere Anwendung von neuronalen Netzen, ist die Bestimmung von Modellparametern unter Verwendung von Echtzeit-Messdaten. Dies ist auch unter dem Namen hybride neuronale Netze bekannt, siehe hierzu [2] und [3].

2 Fallbeispiel: Springender Ball

Diese Kapitel präsentiert ein akademisches Beispiel eines hybriden Systems. Die beiden eingeführten Modellierungsmethoden, das neuronale Netz und die hybride Modellierung, werden anhand des springenden Balls, erläutert. Die physikalische Beschreibung des springenden Balls beschränkt sich grundsätzlich auf die Differentialgleichung und den Anfangsbedingungen in (2).

$$\begin{aligned} \ddot{h}(t) &= -g \\ h(0) &= h_0, \dot{h}(0) = v_0 \end{aligned} \quad (2)$$

Diese Differentialgleichung mit den gegebenen Anfangsbedingungen beschreibt den Zusammenhang zwischen dem Zeitpunkt t und der Höhe des Balls. Diese Formulierung kann man als Zustandsraumdarstellung (3) umformulieren

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ -g \end{pmatrix} \quad (3)$$

wobei hier $x(t)$ den Vektor beschreibt, der die Höhe und die Geschwindigkeit des Balls angibt.

Man könnte sich nun fragen, da es sich um ein Differentialgleichungssystem handelt, warum dies ein hybrides Modell darstellt. Die Systemänderung tritt ein, sobald der Ball am Boden auftrifft. Durch den Energieverlust, der mit dem Deformieren des Balls einhergeht, verliert er

an Geschwindigkeit und ändert seine Flugrichtung. Die Beschreibung des Systems bleibt somit gleich, aber der Parameter des Systems ändert. Für die Formulierung des hybriden Systems muss dieser Systemwechsel an sich als auch die zu erfüllende Bedingung definiert werden.

$$\begin{aligned} \{ (h(t), \dot{h}(t)) : h(t) = 0 \wedge \dot{h}(t) = v(t) \leq 0 \} \\ J(v(t_e)) = -\lambda v(t_e), \lambda \in (0,1) \end{aligned} \quad (4)$$

Die erste Zeile in Gleichung (4) beschreibt jene Punkte an denen das Ereignis eintritt. In der zweiten Zeile wird die Parameteränderung des Systems definiert.

Die Tatsache, dass dieses System eine vergleichsweise einfache Struktur aufweist und zusätzlich noch eine analytische Lösung besitzt, ermöglicht eine umfassende Analyse der unterschiedlichen Simulationsmethoden. Die unterschiedlichen Gesichtspunkte der Analyse werden im Abschnitt 4 erläutert.

3 Implementierung des Fallbeispiels auf zwei Arten

Wie in Abschnitt 2 erwähnt, besitzt dieses System eine analytische Lösung, siehe (5).

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -\frac{g}{2}t^2 + c_2t + c_1 \\ -gt + c_2 \end{pmatrix} \quad (5)$$

Diese Lösung kann nun direkt für die Implementierung des Systems verwendet werden und dient in zur Modellvalidierung.

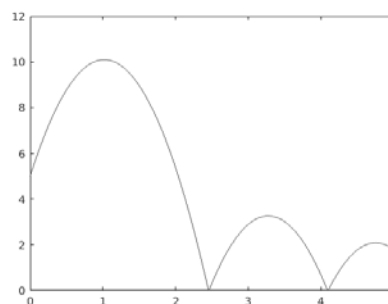


Abbildung 3: Die Simulationsergebnisse des „springenden Balls“ sind hier für das Zeitintervall $t \in [0,12]$ abgebildet.

Abbildung 3 stellt einen exemplarischen Verlauf des springenden Balls dar, wobei $c_2 = v_0$ der Anfangsgeschwindigkeit und $c_1 = h_0$ der Starthöhe des Balles entspricht.

3.1 Hybrides Modell

Die mathematische Formulierung des hybriden dynamischen Modells wurde im Absatz 2 erläutert. Die Implementierung dieses Systems kann nun in unterschiedlichen Simulationsumgebungen bewerkstelligt werden. In diesem Fall wurde das Modell sowohl in MATLAB als auch in Simulink realisiert. In der MATLAB Implementierung kann man einerseits die Differentialgleichung direkt unter Verwendung der Anfangsbedingungen durch vorimplementierte Solver lösen lassen. Mit Hilfe einer Abfrage lässt sich leicht die formulierte Bedingung aus (4) überprüfen und der Parameterwechsel durchführen. Im Hinblick auf die MATLAB Implementierung ist es aber auch möglich die Event Funktion zu verwenden um das Systemverhalten zu simulieren.

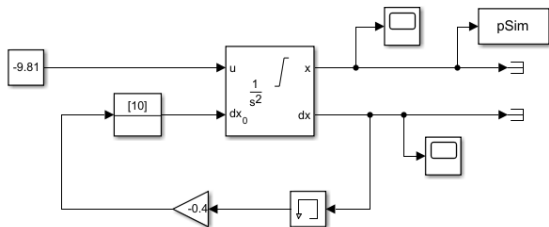


Abbildung 4: Eine der möglichen Implementierungen des springenden Balls unter Verwendung von Simulink und dem Memory-Block.

Mit SIMULINK lässt sich natürlich eine ähnliche Herangehensweise realisieren. Hier formuliert man die Differentialgleichung mittels vorimplementierten Blöcken um verwendet ebenfalls einen If-Block um die Bedingung aus (4) zu überprüfen. Eine weitere Möglichkeit stellt das Einbinden des sogenannten Memory Blocks dar, siehe Abbildung 4.

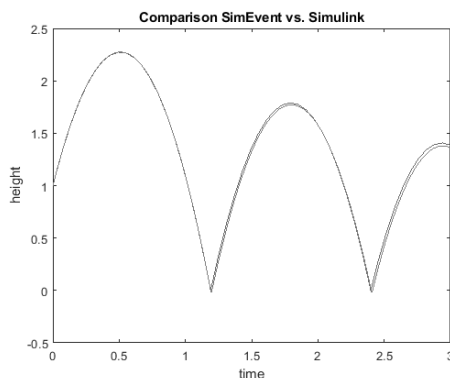


Abbildung 5: Vergleich die Ergebnisse der Simulink Implementierung mittels Memory Block und die Event Funktion in MATLAB.

Alle diese Implementierung stellen im Vergleich zur

analytische Lösung eine gute Approximation des Systemverhaltens da. Die verschiedenen Ansätze im direkten Vergleich liefern ebenfalls zufriedenstellende Ergebnisse. Abbildung 5 zeigt den Vergleich der numerischen Lösung mittels Event Funktion und der Simulink Implementierung mit Hilfe des Memory Blocks.

3.2 Neuronales Netz

Im Hinblick auf die Gestaltung des neuronalen Netzes gibt es nun unterschiedliche Punkte, die festgelegt werden müssen, bevor Ergebnisse verglichen werden können. Um das neuronale Netz trainieren zu können und die Gewichte der Verbindungslinien zu ermitteln benötigt man eine große Anzahl an Datensätze, die mittels der analytischen Lösung aus (5) einfach und schnell erzeugt werden können. Bezüglich der Daten stellt sich nun aber die Frage, welche Parameter können in diesem Beispiel als Input bzw. Output verwendet werden. Zu den möglichen Inputparametern gehören zunächst jedenfalls die Anfangsbedingungen der Differentialgleichung (2) und der Geschwindigkeitsverlust nach dem Aufprall. Betrachtet man Abbildung 3 und definiert den Zeitverlauf des Balls als gewünschten Output des neuronalen Netzes, liegt es nahe eine Diskretisierung der Zeitachse als Input und die dazugehörigen Werte der Simulation als Output zu definieren. In diesem Falle wäre der dazugehörige Output die Höhe des Balls zu den gewählten Zeitpunkten t_i . Die einfachste Implementierung eines neuronalen Netzes, ähnlich zu Abbildung 2, besteht lediglich aus einer einzigen Neuronenschicht. Nehme man hierfür die oben definierten Input- und Outputdaten würde das Modell aus einer einzigen sehr umfangreichen Schicht bestehen. Dementsprechend wäre eine Vielzahl an Datensätzen notwendig um die unzähligen auftretenden Gewichte zu bestimmen. Dies ist weder effizient noch zielführend.

Der gewünschte Ausgang des neuronalen Netzes ist ausschlaggebend für die Struktur des Modells und beeinflusst zeitgleich die möglichen Schichten und ihren Aufbau. Um nun sinnvollere Datensätze zu definieren kann man sich das simulierte Modellverhalten näher ansehen. Einerseits kann ein Ziel des neuronalen Netzes sein, das Systemverhalten zum nächstgelegenen Zeitpunkt vorherzusagen, anstatt den gesamten Verlauf zu prognostizieren. Eine andere Möglichkeit wäre es den Zeitpunkt der nächsten Parameteränderung zu ermitteln. In beiden Fällen reduzieren sich natürlich auch die notwendigen Datenpunkte auf die Anfangsbedingungen als Eingang und

die ausgewählten Punkte, mit eindeutiger Charakterisierung, als Output.

Die Anfangsbedingungen und Parameter des Modells stellen drei Inputparameter des Systems dar. Die Anzahl der benötigten Gewichte reduziert sich dadurch auf ein Minimum. Nun definieren wir noch die passenden Outputparameter. Das Ziel des neuronalen Netzes ist es nun den Zeitpunkt sowie die maximale Höhe des Balls als auch Zeitpunkts des nächsten auftretenden Events zu approximieren.

Abbildung 6 zeigt die Ergebnisse der Implementierung eines neuronalen Netzes mit den eben beschriebenen Input- und Output. Hierbei zeigen die roten Sterne in der Abbildung die approximierten Werte des Netzes an. Die blaue Kurve stellt die analytische Lösung der Differentialgleichung mit denselben Anfangsbedingungen und gleichem Geschwindigkeitsverlust dar.

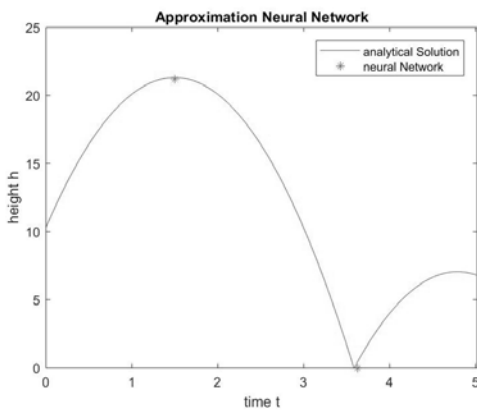


Abbildung 6: Ein Auszug der Ergebnisse des neuronalen Netzes für das Beispiel des springenden Balls im Vergleich mit der analytischen Lösung.

4 Methodenvergleich

Dieser Abschnitt widmet sich nun dem Vergleich der beiden Modellansätze. Worin bestehen die Unterschiede und welche Ähnlichkeiten weisen Sie auf.

4.1 Vergleich der Modelle

Für die numerischen Methoden war keine zusätzliche mathematische Modellbildung notwendig. Die Differentialgleichung, welche den springenden Ball definiert, ist wohlbekannt und einleuchtend. Lediglich die Formulierung der Bedingungen des Events war notwendig. Die Umsetzung in den Simulationsumgebungen ist mit Hilfe dieser mathematischen Beschreibung keine große Kunst. Die vorimplementierten Funktionen, Differentialgleichungslöser sowie Simulink Blöcke vereinfachen die Implementation.

Die Ergebnisse, wie auch Abbildung 5, zeigen, dass die unterschiedlichen Simulationen zuverlässige Ergebnisse liefern.

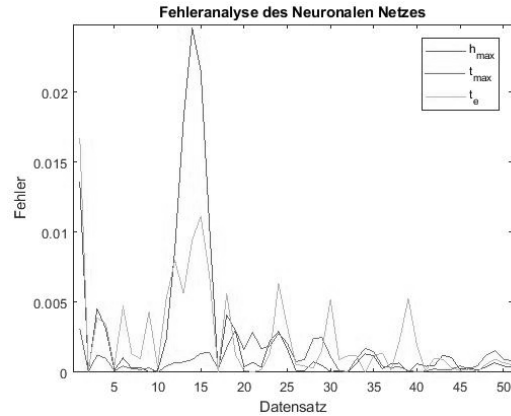


Abbildung 7: Diese Graphik zeigt den Fehlerverlauf der Ergebnisse des neuronalen Netzes bezogen auf den gesamten Datensatz.

Die Ergebnisse des neuronalen Netzes bilden das Systemverhalten wie in Abbildung 6 ersichtlich ebenfalls hinreichend genau ab. In Tabelle 1 sind die Ergebnisse für einen Datensatz ersichtlich. Abbildung 7 zeigt den Verlauf des Fehlers für den gesamten Datensatz. Die Abweichungen der Parameter unterschieden sich wesentlich. Der Fehler im Extremum der Flugbahn zeigt die größte Abweichungen. Die Approximation des Zeitpunkts für den Höhepunkt des Balls ist hier am stabilsten.

	Input Daten		Output Daten	Output neuronales Netz
h_0	10.2	h_{max}	21.3641	21.3686
v_0	14.8	t_{max}	1.51	1.5112
c	0.8	t_e	3.6	3.6039

Tabelle 1: Diese Tabelle vergleicht bei selben Eingangswerten die zugehörigen Ausgangswerte der analytischen Lösung mit der des neuronalen Netzes.

Die Fehler der Outputwerte ist maximal 0.0245 und ist selbst für h_{max} im Mittel kleiner als 0.0027. Dies zeigt, dass das hier verwendete neuronale Netz Teile des Systemverhaltens gut darstellen kann.

4.2 Spezielle Modellverhalten

Hybride Modelle zeichnen sich nicht nur durch die Kombination von diskreten und kontinuierlichen Modell-

beschreibungen aus, sondern können auch den sogenannten Zeno Effekt erzeugen. Dieser Effekt beschreibt das Phänomen unendlich viele Ereignisse in einer endlichen Zeitspanne zu erzeugen. Dies ist auch im Fall vom springenden Ball möglich.

Im Zuge der Implementierung wurde dieses Phänomen bisher vernachlässigt. Dieses Problem ist rein mathematisch nachbildbar, siehe [4]. Nun ist aber die Frage wie es sich bei der Simulation dieses Effekts verhält. Im Simulink Modell führt die richtige Wahl der Parameter auf eine Fehlermeldung, dass die Anzahl an möglichen Events überschritten sei. Im Hinblick auf die neuronalen Netze stellt dieses Phänomen eine noch viel größere Herausforderung dar. Die bisherigen Ergebnisse legen allerdings nahe, dass dieser Effekt vom neuronalen Netz nur ansatzweise dargestellt werden kann, falls dies auch im Datensatz enthalten war. Das Modell muss diesen Fall gelernt haben ansonsten führt die Ausführung des neuronalen Netzes zu völlig falschen Werten.

5 Zusammenfassung und Ausblick

Das hier gewählte Fallbeispiel ist von einfacherer Struktur. Abbildung 6 aber, dass man das Systemverhalten nicht nur mit Hilfe der Differentialgleichungen, sondern auch mit einer Implementierung eines neuronalen Netzes abbilden kann. Dies alleine ist natürlich noch kein Nachweis dafür, dass neuronale Netze in der hybriden Modellbildung eine sinnvolle Herangehensweise darstellen. Es belegt aber, dass gewisse hybride Modelle durch neuronale Netze gänzlich oder aber zumindest teilweise realisiert werden können, wie auch in [3] gezeigt wird.

Bevor man der Frage nach der Ersetzbarkeit von hybriden Modellen durch neuronalen Netzen anhand von Analysen komplexerer Modelle nachgehen kann, gibt es weitere Punkte, die mit dem Beispiel des springenden Balles untersucht werden können. Genauere Untersuchungen des Zeno Effekts mit den beschriebenen numerischen Methoden ist einer der nächsten Schritte.

Im Hinblick auf die neuronalen Netze, lässt sich zusammenfassen, dass es möglich ist, hybride Modelle gänzlich bzw. teilweise durch diese Methode darzustellen. Es ist dabei sehr wichtig sich genau zu überlegen, welche der verfügbaren Daten für die Festlegung der Struktur des neuronalen Netzes sinnvoll sind. Eine Analyse der unterschiedlichen Netzstrukturen im Zusammen-

hang mit der Genauigkeit der Ergebnisse und der Geschwindigkeit des Trainingsalgorithmus anhand des hier erklärten Beispiels werden in der zukünftigen Arbeit berücksichtigt. Zusätzlich wird sich eine weitere Implementierung mit dem im Abschnitt 4.2 erwähnten Zeno Effekt auseinandersetzen. Hier ist aber auch die Frage, ob ein herkömmliches neuronales Netz dieses Phänomen überhaupt abbilden kann.

Die bisherigen Evaluationen beschränkten sich auf Eingangsdaten im Bereich der zur Verfügung gestellten Messdaten. Eine weitere Untersuchung beschäftigt sich damit, die Genauigkeit der Ergebnisse unter Verwendung von Eingangsdaten außerhalb des Trainingsbereichs zu nutzen. Arbeiten von G. Martinus und C.H. Lampert [5] beschäftigen sich mit dieser Fragestellung indem sie den Aufbau und der Funktionalität neuronaler Netze eine neue Richtung geben. Sie untersuchen die Möglichkeit Netze gleichungsbasierten Beschreibungen des Systemverhaltens durch neuronale Netze zu erzeugen. Dies ermöglicht ihnen, das Verhalten des Systems vorherzusagen, ohne die notwendigen Datensätze im Training zu inkludieren. Diese Erkenntnisse könnte auch in der Bewältigung des Zeno Effekts eine Hilfestellung darstellen.

Zusammenfassend lässt sich sagen, dass diese Arbeit am Anfang der Untersuchung der Verwendbarkeit von neuronalen Netzen im Zusammenhang mit der hybriden Modellierung darstellt.

References

- [1] A. Körner, S. Winkler und F. Breiteneker. *Possibilities in State Event Modelling of Hybrid Systems*. 9th EURO-SIM Congress on Modelling and Simulation, Oulu, pp. 542-543, 2016.
- [2] D. C. Psychogios und L. H. Ungar. *A Hybrid Neural Network-First Principles Approach to Process Modeling* AIChE Journal vol. 38, No. 10, pages 1499-1511, 1992.
- [3] W. Lu, P. Zhu und S. Ferrari. *A Hybrid-Adaptive Dynamic Programming Approach for the Model-Free Control of Nonlinear Switched Syses*. IEEE Transactions on Automatic Control vol. 61, No. 10, pages 3203-3208, 2016.
- [4] Or Y., Teel A. R., 2010. Using the Set-Values Bounding Ball for Bounding Zeno Solution of Lagrangian Hybrid Systems. In Proceedings: 8th IFAC Symposium on Nonlinear Control Systems, pp 801-806.
- [5] Martinus G., Lampert C.H., 2016. Extrapolation and learning equations. 2016. arXiv preprint <https://arxiv.org/abs/1610.02995>.

Erstellung eines mikroskopischen Analogons zu einem makroskopischen Fishing Model

Dennis Leser^{1*}, Martin Bicher¹, Niki Popper²

¹Institute for Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *dennis.leser@gmx.at

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

Abstract. Diese Arbeit befasst sich mit der modelltheoretischen Fragestellung: Wie kann man bestehende, alt bewährte makroskopische Modelle in eine mikroskopische Beschreibungssprache "übersetzen", ohne deren Validität zu verlieren? Dazu wollen wir zu einem bestehenden makroskopischen Modell ein mikroskopisches Analogon erzeugen. Zum einen kann man sich hiermit die Vorteile der mikroskopischen Darstellung zunutze machen. Zum anderen hat man auf diese Art ein Paar von zwei validen Modellen, welches man beliebig vergleichen und für die Validierung nutzen kann.

Einleitung

Gerade in den letzten Jahren hat in der Modellbildung und Simulation der Einsatz von mikroskopischen Modellen stark zugenommen. Die Gründe dafür liegen einerseits in der rasanten Weiterentwicklung von Prozessorleistung und Arbeitsspeicher von Computern. Während die computerunterstützte Simulation solcher Modelle vor wenigen Jahrzehnten noch eine große informatische Herausforderung war, ist es heutzutage möglich agentenbasierte Modelle mit mehreren Millionen Agenten in Minutenschnelle zu simulieren [1]. Zum anderen bieten mikroskopische Modellansätze, im direkten Vergleich mit klassischen makroskopischen Modellansätzen große Vorteile, was Flexibilität und ihre Fähigkeit komplexes Verhalten abzubilden anbelangt [2]. Diese Entwicklung wirft auch Modelltheoretisch neue Fragestellungen auf, wovon wir eine in dieser Arbeit näher betrachten werden: Wie kann man bestehende, alt bewährte makroskopische Modelle in eine mikroskopische Beschreibungssprache "übersetzen", ohne deren Validität zu verlieren?

In dieser Arbeit wollen wir zu einem bestehenden makroskopischen Modell ein mikroskopisches Analogon erzeugen, um somit ein Paar von zwei validen Mo-

dellen zu erhalten, welches man beliebig vergleichen und für die Validierung nutzen kann (vgl. Bernhard Zeigler - "Why should we develop Simulation Models in Pairs?"[3]). Einerseits ist dieser Prozess invers zu den auf die klassische Molekularfeld Theorie (Mean-Field Theory) basierenden Ansätzen [4, 5], welche stets ein mikroskopisches durch ein makroskopisches Modell zu beschreiben versuchen. Andererseits wählen wir den Zugang auch deutlich weniger Abstrakt als in [6] vorgeschlagen und lassen uns sehr wohl vom Realsystem inspirieren, um auf jeden Fall ein sinnvolles mikroskopisches Modell zu erhalten.

1 Das untersuchte System

Da der Fokus der Arbeit auf dem Erarbeiten einer neuen mikroskopischen Modellbeschreibung liegt, wollen wir uns bei der Systembeschreibung und der makroskopischen Modellierung auf Vorarbeiten stützen. Auch ein Validierungsprozess mit einem tatsächlichen Referenzsystem wird nicht durchgeführt und das Modell mit einer beispielhaften Parametrisierung simuliert. Wir folgen hierfür [7] und untersuchen eine homogene, beschränkte Fischerzone in der nur eine Fischart lebt

Parameter	Bedeutung
α	jährliche Fangrate pro Fischerboot
β	Lebensdauer eines Fischerbootes
η	durchschn. Instandhaltungskosten eines Fischerbootes pro Jahr
κ	max. Kapazität der Fischerzone
p_b	Preis eines neuen Fischerbootes
p_f	Verkaufspreis eines Fisches

Tabelle 1: Zur Modellierung herangezogene Systemparameter.

und Fischer ihrer Arbeit nachgehen. Die für den Modellierungsprozess herangezogenen Systemparameter sind der Tabelle 1 zu entnehmen.

2 Das makroskopische Modell

Die zugehörigen Modellgleichungen lauten wie folgt

$$\underbrace{\dot{X}(t) = X(t) \left(1 - \frac{X(t)}{\kappa} \right) - \frac{X(t)Y(t)\alpha}{100}}_{=: (1a)}, \quad (1)$$

$$\dot{Y}(t) = \frac{p_f X(t)Y(t)\alpha - 100Y(t)\eta}{100p_b} - \frac{Y(t)}{\beta}, \quad (2)$$

wobei $X(t)$ die Anzahl der Fische und $Y(t)$ die Anzahl der Fischerboote zum Zeitpunkt t repräsentieren. Die Gleichung (1a) beschreibt das makroskopische Modell, wenn nicht gefischt wird und wird im Folgenden als makroskopisches Fish Model bezeichnet. Die Gleichungen (1) und (2) beschreiben die Dynamik des makroskopischen Fishing Models. Die Abbildungen 1 und 2 zeigen das Verhalten der makroskopischen Modelle über einer Zeitspanne T , für unterschiedlichen Modellparameter und numerisch gelöst mit MATLAB's ode45 Methode.

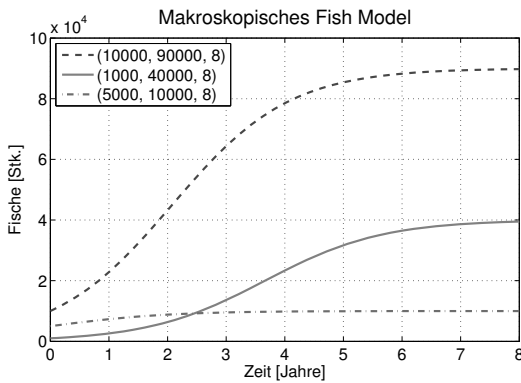


Abbildung 1: Makroskopisches Fish Model mit drei verschiedenen Parametersätzen (x_0, κ, T) .

3 Das mikroskopische Modell

3.1 Simulation des mikroskopischen Fish Models

Im mikroskopischen Fish Model wird jeder Fisch durch einen Fischagenten repräsentiert. Als Umgebung wählen wir ein quadratisches, zweidimensio-

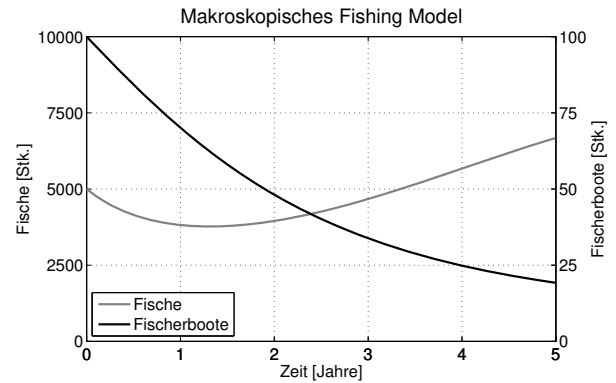


Abbildung 2: Makroskopisches Fishing Model mit den Parametern: $x_0 = 5000$, $y_0 = 100$, $\kappa = 10000$, $\alpha = 1$, $\beta = 15$, $\eta = 50000$, $p_f = 500$, $p_b = 100000$, $T = 5$;

nales Gitter mit festen Grenzen und einer Moore-Nachbarschaft, welches die untersuchte homogene, beschränkte Fischerzone, in der nur eine Fischart lebt, darstellt. Pro Gitterzelle kann sich höchstens ein Fischagent aufhalten. Da beim mikroskopischen Modell jeder Fisch einzeln betrachtet wird, führen wir einen Modellparameter $\mu \in \mathbb{R}^+$ ein, von dem das Wachstum des Fischbestandes abhängt. Die Fischagenten besitzen jeweils nur einen Zustand $z \in (0, 1]$, welcher eine uniform verteilte Zufallszahl ist und quasi seine Bereitschaft zur Vermehrung repräsentiert. Des Weiteren befolgen sie eine einfache Regel: Falls sich in der Moore-Nachbarschaft eines Fischagenten mindestens ein freier Platz befindet und der zugehörige Zustand $z < r_{and}\mu$ ist, nimmt ein neuer Fischagent einen dieser freien Plätze ein und wird mit einer neuen uniform verteilten Zufallszahl $z_{neu} \in (0, 1]$ initialisiert, wobei r_{and} ebenfalls eine uniform verteilte Zufallszahl aus $[0, 1]$ ist. Für die Simulation implementieren wir dieses agentenbasierte mikroskopische Fish Model in MATLAB und kalibrieren den Modellparameter μ wie folgt.

3.2 Kalibrierung des Modellparameters μ

Um den Modellparameter μ als Parameterfunktion von x_0 und κ zu kalibrieren, suchen wir eine einfache Abbildungsvorschrift $g : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$, die für beliebige Parameter x_0 und κ den kalibrierten Modellparameter μ liefert. Dazu berechnen wir zunächst für ein festes Tupel (x_0, κ) und $\mu = 0.1, 0.2, \dots, 10$ das zugehörige arithmetische Mittel $\bar{\delta}_f$ der relativen Fehler δ_k^f , mit $k = 1, 2, \dots, T$, bzgl. des

Fischbestandes zwischen den Fish Models, wobei

$$\bar{\delta}_f = \frac{1}{T} \sum_{k=1}^T \delta_k^f = \frac{1}{T} \sum_{k=1}^T \frac{|X_{makro}(t_k) - X_{mikro}(t_k)|}{X_{makro}(t_k)}. \quad (3)$$

Hierbei bezeichnen $X_{makro}(t_k)$ bzw. $X_{mikro}(t_k)$ die Anzahl der Fische des makroskopischen bzw. des mikroskopischen Modells zum Zeitpunkt $t_k = k$, mit $k = 1, 2, \dots, T$. Der kalibrierte Modellparameter μ für dieses feste Tupel (x_0, κ) ist somit jener Wert, bei dem $\bar{\delta}_f$ sein Minimum annimmt. In dem wir diese Vorgehensweise für verschiedene Tupel (x_0, κ) wiederholen, erhalten wir das Verhalten des kalibrierten Modellparameters μ in Abhängigkeit der Parameter x_0 und κ und approximieren dieses durch die einfache Abbildungsvorschrift

$$g : (x_0, \kappa) \mapsto \exp\left(-\frac{x_0}{\kappa}\right) \frac{\kappa}{2x_0}. \quad (4)$$

3.3 Simulation des mikroskopischen Fishing Models

Das mikroskopische Fishing Model wird durch ein Multi-Agent Model realisiert, indem jeder Fisch durch einen Fischagenten und jedes Fischerboot durch einen Fischerbootagenten repräsentiert wird. Wie beim mikroskopischen Fish Model wählen wir als Umgebung ein quadratisches, zweidimensionales Gitter mit festen Grenzen und pro Gitterzelle kann sich wieder höchstens ein Agent aufhalten. Als Nachbarschaft für sämtliche Agenten wird eine erweiterte Moore-Nachbarschaft mit einer bestimmten Reichweite gewählt, da ein Fischerbootagent mit einer einfachen Moore-Nachbarschaft pro Zeitschritt (Jahr) maximal 8 Fischagenten fangen könnte. Dies würde bei weitem nicht ausreichen, um die Instandhaltungskosten η eines Fischerbootes pro Zeitschritt (Jahr) zu decken. Da beim mikroskopischen Modell jeder Fisch und jedes Fischerboot einzeln betrachtet wird, müssen wir die Modellparameter $\mu \in [0, 1]$, $\theta \in [0, 1]$ und $\varepsilon \in \mathbb{N}$ einführen. Wobei μ das Wachstum des Fischbestandes, θ wie viele Fische ein Fischerboot pro Zeitschritt (Jahr) fängt und ε die Reichweite der erweiterten Moore-Nachbarschaft bestimmt. Die Fischagenten besitzen analog zum mikroskopischen Fish Model jeweils nur einen Zustand $z \in (0, 1]$. Sie befolgen eine einfache Regel: Falls sich in der erweiterten Moore-Nachbarschaft eines Fischagenten $k \in \mathbb{N}$ freie Plätze befinden und der zugehörige Zustand $z > r_{and}$ ist, dann werden $n \in \mathbb{N}$ neue Fisch-

agenten in dieser erweiterten Moore-Nachbarschaft erstellt, wobei $n = \lceil \mu k \rceil$. Die Fischerbootagenten besitzen den Zustand $u \in \mathbb{R}^+$, der angibt, wie alt das Fischerboot ist, und den Zustand $w \in \mathbb{R}^+$, der für das aktuelle Kapital eines Fischerbootes steht. Diese Zustände werden jedem Fischerbootagenten bei seiner Initialisierung durch $u = r_{and}(\beta - 1)$ und $w = r_{and}\eta$ zugewiesen. Des Weiteren halten sie sich an folgende Regeln:

- Falls sich $k \in \mathbb{N}$ Fischagenten in der erweiterten Moore-Nachbarschaft eines Fischerbootagenten befinden, fängt dieser $m \in \mathbb{N}$ Fischagenten, wobei $m = \lceil \theta k \rceil$.
- Falls für einen Fischerbootagenten $w < \eta$ gilt, wird dieser Fischerbootagent entfernt. Ansonsten werden die Instandhaltungskosten η von dem Kapital w dieses Fischerbootagenten abgezogen.
- Falls für einen Fischerbootagenten $w \geq p_b$ gilt, wird ein neuer Fischerbootagent in einer zufälligen freien Gitterzelle erstellt und der Preis eines neuen Fischerbootes p_b von dem Kapital w abgezogen.
- Falls für einen Fischerbootagenten $u \geq \beta$ gilt, wird dieser Fischerbootagent entfernt.

Um dieses agentenbasierte mikroskopische Fishing Model zu simulieren, implementieren wir es in MATLAB und kalibrieren die Modellparameter μ , θ und ε .

3.4 Kalibrierung der Modellparameter μ , θ und ε

Im Unterschied zum mikroskopischen Fish Model kalibrieren wir die Modellparameter μ , θ und ε des mikroskopischen Fishing Models nur für die Parameter $x_0 = 5000$, $y_0 = 100$, $\kappa = 10000$, $\alpha = 1$, $\beta = 15$, $\eta = 50000$, $p_f = 500$, $p_b = 100000$ und $T = 5$. Dazu implementieren wir in MATLAB einen evolutionären Algorithmus, der abbricht, falls ein Individuum der aktuellen Population $\bar{\delta}_f \leq \tau_f$ und $\bar{\delta}_b \leq \tau_b$ erfüllt, oder eine maximale Anzahl an Durchgängen j_{max} erreicht wird. Wobei τ_f und τ_b die Toleranzen bzgl. des Fisch- und Fischerbootbestandes angeben. Das arithmetische Mittel $\bar{\delta}_b$ der relativen Fehler δ_k^b , mit $k = 1, 2, \dots, T$, bzgl. des Fischerbootbestandes zwischen den Fishing Models ist analog wie $\bar{\delta}_f$ in der Gleichung (3) definiert. Schließlich liefert der evolutionäre Algorithmus für $\tau_f = 0.2$, $\tau_b = 0.3$ und $j_{max} = 25$ die kalibrierten Modellparameter $\mu = 0.027$, $\theta = 1$ und $\varepsilon = 8$.

4 Modellresultate

Das makroskopische und das mikroskopische Fish Model werden für die Parameter $x_0 = 5000$, $\kappa = 10000$ und $T = 8$ miteinander verglichen. Dazu betrachten wir das Maximum $\delta_{max}^f = 0.09$ und das arithmetische Mittel $\bar{\delta}_f = 0.02$ der relativen Fehler δ_k^f , mit $k = 1, 2, \dots, T$, bzgl. des Fischbestandes zwischen den Fish Models. Die Abbildung 3 zeigt den graphischen Vergleich der beiden Fish Models.

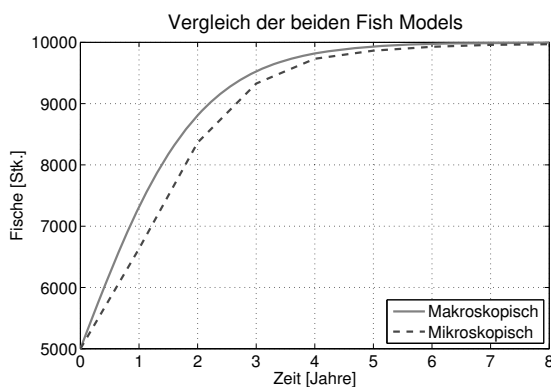


Abbildung 3: Vergleich der beiden Fish Models mit den Parametern $x_0 = 5000$, $\kappa = 10000$ und $T = 8$.

Die beiden Fishing Models werden für die Parameter, mit denen das mikroskopische Fishing Model im Abschnitt 3.4 kalibriert wird, miteinander verglichen. Für den Vergleich betrachten wir die Maxima $\delta_{max}^f = 0.21$ und $\delta_{max}^b = 0.46$, sowie die arithmetischen Mittel $\bar{\delta}_f = 0.14$ und $\bar{\delta}_b = 0.27$ der relativen Fehler bzgl. des Fisch- und Fischerbootbestandes zwischen den Fishing Models. Die Abbildung 4 illustriert den graphischen Vergleich der beiden Fishing Models.

5 Conclusio

In dieser Arbeit präsentierten wir anhand eines ökonomischen Beispiels eine Methode, wie man ein mikroskopisches Analogon zu einem makroskopischen Fishing Model erstellen kann. Das Konzept lässt sich gut für andere, ähnlich strukturierte Differentialgleichungsmodelle verallgemeinern und kann langfristig dazu beitragen, das Verhalten mikroskopischer Modelle besser zu verstehen.

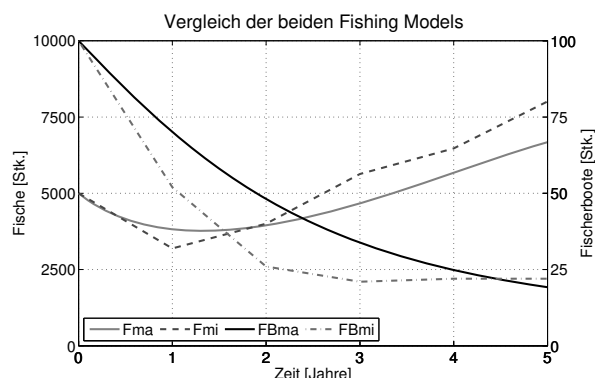


Abbildung 4: Fma bzw. Fmi bezeichnen die Fisch- und FBma bzw. FBmi die Fischerbootbestände des makroskopischen sowie des mikroskopischen Fishing Models.

Literatur

- [1] Bicher M, Glock B, Miksch F, Schneckenreither G, Popper N. Definition, Validation and Comparison of Two Population Models for Austria. In: *Proceedings of 4th UBT Annual International Conference on Business, Technology and Innovation*. Durres, Albania: UBT - Higher Education Institution. 2015; .
- [2] Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*. 2002; 99(Supplement 3):7280–7287.
- [3] Zeigler B. Why should we develop simulation models in pairs? 2017.
- [4] Boudec JYL, McDonald D, Mundinger J. A Generic Mean Field Convergence Result for Systems of Interacting Objects. In: *Proceedings of the 4th International Conference on the Quantitative Evaluation of Systems*. IEEE. 2007; pp. 3–18.
- [5] Bicher M, Popper N. Mean-Field Approximation of a Microscopic Population Model for Austria. In: *Proceedings of the 9th EUROSIM Congress on Modelling and Simulation*. Oulu, Finland. 2016; pp. 544–545.
- [6] Bicher M. Classification of Microscopic Models with Respect to Aggregated System Behaviour. Dissertation, TU Wien, Vienna, Austria. 2017.
- [7] Whelan J. Building the fish banks model and renewable resource depletion: system dynamics education project. *Course material-MIT Sloan School of Management*. 1994;p. 70.

Optimale Steuerung und theoretische Analyse eines mathematischen zwei Serotypen Dengue-Fieber Modells

Gaby Albrecht^{1*}, Kurt Chudej¹

¹Lehrstuhl für Wissenschaftliches Rechnen, Universität Bayreuth,
95440 Bayreuth, Germany; *gaby.albrecht@uni-bayreuth.de

Abstract. Dengue Fieber ist eine Krankheit, die durch die Infektion mit dem Denge-Virus hervorgerufen wird. Dieser Virus wird durch den Stich einer infizierten Asiatischen Tigermücke übertragen. Die sehr robuste und anpassungsfähige Stechmückenart stammt ursprünglich aus Südostasien, jedoch ist durch Warentransporte und Tourismus eine zunehmende (Wieder-)Besiedlung in Europa zu beobachten. Seit wenigen Jahren sind erste stabile Populationen in Süddeutschland detektiert worden. Mit der Ausbreitung der Asiatischen Tigermücke ist auch eine zunehmende Gefahr des Dengue Fiebers einhergehend. Die WHO schätzt die jährliche Infektionsrate auf ca. 390 Millionen Infektionen Weltweit. Der Dengue Virus existiert mit 4 verschiedenen Serotypen. Jeder Serotyp löst das Dengue Fieber aus. Bei einer Zweitinfektion kann ein hämorrhagische Dengue Fieber bis hin zu einem Dengue Schock Syndrom auftreten. Aufgrund dessen wird in diesem Paper ein Zwei-Serotypen-Modell betrachtet, charakteristischen Größen berechnet und numerischen Ergebnissen diskutiert. Erstmals wird eine temporäre Immunität bei der Zweitinfektion berücksichtigt und zusätzlich europäische Mückenparameter in den numerischen Simulation und der optimalen Steuerung benutzt.

Einleitung

Seit einigen Jahren sind Funde erwachsener Tigermücken, bzw. noch schlimmer eigenständige Tigermückenpopulationen, in einigen Orten und Städten Süddeutschlands bekannt [1, 2, 3, 4]. Diese unglückliche Szenerie wird durch die Klimaänderung begünstigt [5, 6]. Zudem sind 708 Dengue Erkrankungen in Deutschland im Jahr 2015 beobachtet worden, die wohl alle durch Fernreisen eingeschleppt wurden

[7]. Denguefieber ist daher eine vor der Tür stehende Krankheit [8, 9, 10, 11, 12]. Dies motiviert die vorliegende Forschung an mechanistischen Kompartimentmodellen für Denguefieber.

Es gibt bereits eine Reihe von Dengue-Fieber-Modellen, die sowohl die Mücken als auch die Menschen mitmodellieren, siehe z.B. die Ein-Serotyp-Modelle von [13, 14, 15] bzw. [16, 17] sowie z.B. die Zwei-Serotyp-Modelle von [18, 19, 20]. Neu in diesem Modell ist die Berücksichtigung einer temporären Immunität bei der Zweitinfektion (WHO 2018) und die Benutzung neuer europäischen Mückenparameter.

1 Mathematisches Modell

Das dargestellte Zwei-Serotypen-Modell beschreibt ein Modell für das Dengue-Fieber repräsentativ für zwei der vier möglichen Serotypen. Wird ein Infizierbarer aus dem Kompartiment S_h mit einem Serotyp i infiziert, wandert dieser in das Kompartiment der Infizierten I_h^i . Neu in diesem Modell ist nun die Einführung einer temporären Kreuzimmunität. Diese beschreibt, dass nach einer Genesung des Infizierten mit Serotyp i eine kurzweilige Immunität gegen beide Serotypen besteht. Das bedeutet, dass der vorher Infizierte mit der Genesungsrate η_h in das Kompartiment R_h^i mit einer temporären Kreuzimmunität übergeht. Die temporäre Kreuzimmunität lässt jedoch mit der Rate χ_h nach und es findet ein Übergang in das Kompartiment S_h^{ij} statt, in welchem eine Empfänglichkeit für eine Infizierung mit Serotyp j besteht. Mit der Genesung gegen den zweiten Serotypen geht dann eine lebenslange Immunität gegen beide Serotypen einher. Bei der Mückendynamik wird unterschieden zwischen den infizierbaren Mücken S_v , den infizierten Mücken I_v^i und den aquatischen Mücken A_v . Letztere stehen repräsentativ für alle Mückenstadien bis zur ausgewachsenen, stechfähigen

gen Mücke. Die graphische Darstellung dieses Zwei-Serotypen-Modells ist in Abb. 1 dargestellt.

Die Modellierung dieses Kompartimentmodells resultiert in die nichtlinearen Differentialgleichungen für die menschliche Population

$$\begin{cases} \frac{dS_h(t)}{dt} = \mu_h N_h - \left(\frac{\beta_1 I_v^1(t) + \beta_2 I_v^2(t)}{N_v} + \mu_h \right) S_h(t), \\ \frac{dI_h^i(t)}{dt} = \beta_i \frac{I_v^i(t)}{N_v} S_h(t) - (\eta_h + \mu_h) I_h^i(t), \\ \frac{dR_h^i(t)}{dt} = \eta_h I_h^i(t) - (\chi_h + \mu_h) R_h^i(t), \\ \frac{dS_h^{ij}(t)}{dt} = \chi_h R_h^i - \left(\beta_{ij} \frac{I_v^j(t)}{N_v} + \mu_h \right) S_h^{ij}(t), \\ \frac{dI_h^{ij}(t)}{dt} = \beta_{ij} \frac{I_v^j(t)}{N_v} S_h^{ij}(t) - (\eta_h + \mu_h) I_h^{ij}(t), \\ \frac{dR_h(t)}{dt} = \eta_h (I_h^{12}(t) + I_h^{21}(t)) - \mu_h R_h(t), \end{cases}$$

und analog für die Mückenpopulation

$$\begin{cases} \frac{dA_v(t)}{dt} = \varphi \left(1 - \frac{A_v(t)}{\alpha k N_h} \right) (S_v(t) + I_v^1(t) + I_v^2(t)) - (\eta_A + [\mu_A + c_A]) A_v(t), \\ \frac{dS_v(t)}{dt} = \eta_A A_v(t) - \left(\tilde{\beta} \frac{I_h^1(t) + I_h^2(t) + I_h^{21}(t) + I_h^{12}(t)}{N_h} + [\mu_v + c_v] \right) S_v(t), \\ \frac{dI_v^i(t)}{dt} = \tilde{\beta} \frac{I_h^i(t) + I_h^{ji}(t)}{N_h} S_v(t) - [\mu_v + c_v] I_v^i(t). \end{cases}$$

Das Modell enthält weiter eine zeitabhängige Bekämpfung der aquatischen Mücken mit einem Larvizid $c_A(t)$, eine zeitabhängige Bekämpfung der erwachsenen Mücken mit einem Insektizid $c_v(t)$ und eine mechanische Kontrolle $\alpha(t)$, die die Trägerkapazität der aquatischen Mücken beeinflusst.

2 Theoretische Analyse, Numerische Simulation und Optimale Steuerung

In diesem Kapitel werden die Gleichgewichtspunkte ausgerechnet und ihre Stabilität beschrieben. Als wichtiger biologisch-medizinischer Parameter wird dann die Basisreproduktionszahl des Modells bestimmt. Im Anschluss werden numerische Simulationen für das Einschleppen von erkrankten Menschen in das Kompartimentmodell mit temporärer Kreuzimmunität mit den denen ohne temporärer Kreuzimmunität

verglichen. Abschliessend wird eine optimale Bekämpfungsstrategie mit zeitabhängigen Steuerungen präsentiert.

2.1 Theoretische Analyse

Für die Analyse der Basisreproduktionszahl und der Gleichgewichtspunkte erfolgt die Annahme von festen Werten für die Kontrollparameter c_A, c_v und α . Die Berechnungen werden in der Menge Ω durchgeführt, mit

$$\Omega = \{ (S_h, I_h^1, I_h^2, R_h^1, R_h^2, S_h^{12}, S_h^{21}, I_h^{12}, I_h^{21}, R_h; A_v, S_v, I_v^1, I_v^2) \in \mathbb{R}_+^{14} \mid S_h + S_h^{12} + S_h^{21} + I_h^1 + I_h^2 + I_h^{12} + I_h^{21} + R_h^1 + R_h^2 + R_h \leq N_h; A_m \leq k N_h, S_v + I_v^1 + I_v^2 \leq m N_h \}.$$

Die Gleichgewichtspunkte für die Menge Ω ergeben sich zu

- trivialer GGP

$$E_t = (N_h, 0, 0, 0, 0, 0, 0, 0, 0, 0 \mid 0, 0, 0, 0)$$

- krankheitsfreier GGP

$$E_{DFE} = (N_h, 0, 0, 0, 0, 0, 0, 0, 0, 0 \mid A_v^*, M_v^*, 0, 0)$$

- endemischer GGPI

$$E_1 = (S_h^*, I_h^*, 0, R_h^*, 0, \tilde{S}_h^*, 0, 0, 0, 0 \mid A_v^*, S_v^*, I_v^*, 0)$$

- endemischer GGP2

$$E_2 = (S_h^*, 0, I_h^*, 0, R_h^*, 0, \tilde{S}_h^*, 0, 0, 0 \mid A_v^*, S_v^*, 0, I_v^*)$$

- endemischer GGP3

$$E_c = (S_h^c, I_h^c, I_h^c, \hat{R}_h^c, \hat{R}_h^c, \hat{S}_h^c, \hat{S}_h^c, \hat{I}_h^c, \hat{I}_h^c, R_h^c \mid A_v^c, S_v^c, I_v^c, I_v^c),$$

$$\text{mit } A_v^c = A_v^*$$

Zur Berechnung der Basisreproduktionszahl werden die Gleichungen der Infizierten Kompartimente umgeschrieben in die Zu- und Abflüsse des jeweiligen Kompartiments. Es ergibt sich folgende Darstellung:

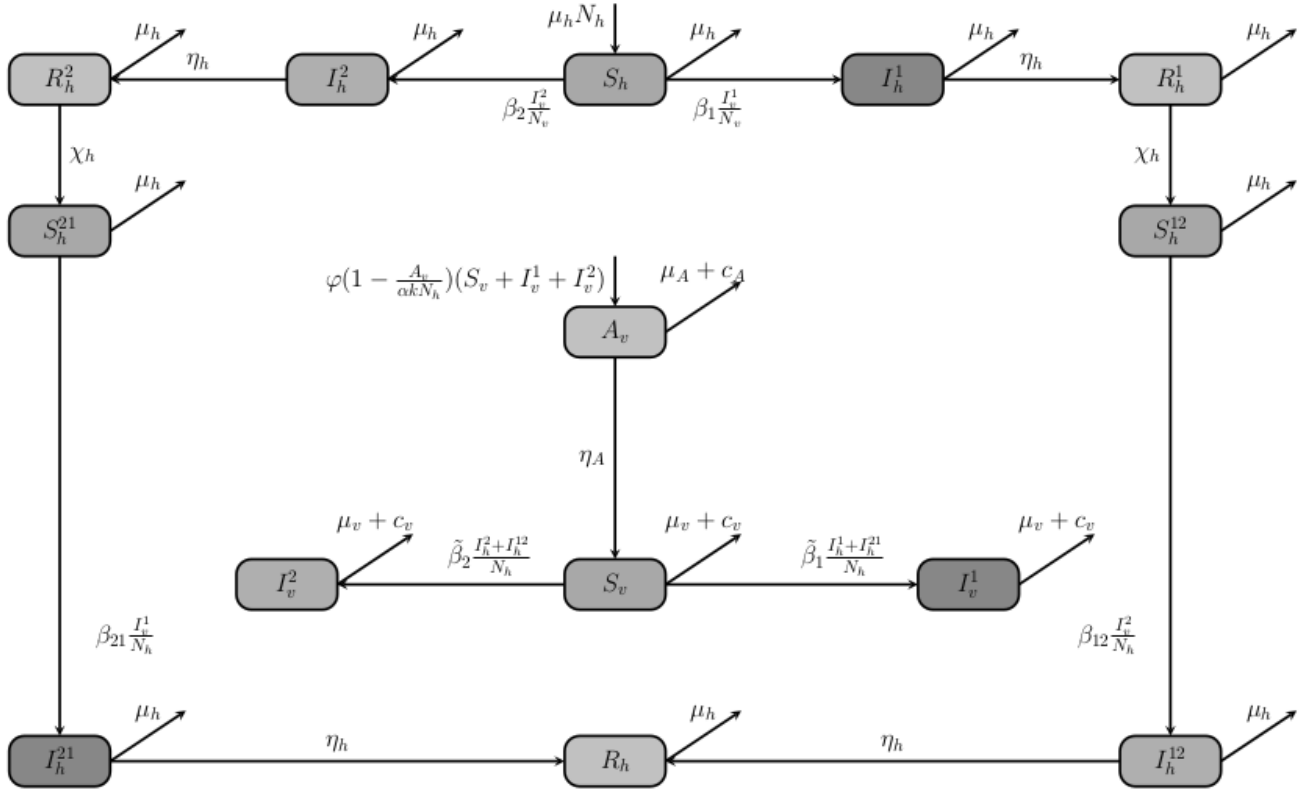


Figure 1: Zwei-Serotyp Dengue Fieber Modell mit temporärer Kreuzimmunität bei Erstinfektion

$$\begin{pmatrix} j_h^1 \\ j_h^2 \\ j_h^{12} \\ j_h^{21} \\ j_v^1 \\ j_v^2 \end{pmatrix} = \begin{pmatrix} \beta_1 \frac{I_v^1}{N_h} S_h \\ \beta_2 \frac{I_v^2}{N_h} S_h \\ \beta_{12} \frac{I_v^1}{N_h} S_h^{12} \\ \beta_{21} \frac{I_v^2}{N_h} S_h^{21} \\ \tilde{\beta}_1 \frac{I_h^1 + I_h^{21}}{N_h} S_v \\ \tilde{\beta}_2 \frac{I_h^2 + I_h^{12}}{N_h} S_v \end{pmatrix} - \begin{pmatrix} (\eta_h + \mu_h) I_h^1 \\ (\eta_h + \mu_h) I_h^2 \\ (\eta_h + \mu_h) I_h^{12} \\ (\eta_h + \mu_h) I_h^{21} \\ (\mu_m + c_v) I_v^1 \\ (\mu_m + c_v) I_v^2 \end{pmatrix}$$

Um die Basisreproduktionszahl zu berechnen, werden die Jacobi-Matrizen des Zuflusses, bzw. der Neuinfektionen und des Abflusses, bzw. den Transferen in andere Kompartimente benötigt. Die Jacobi Matrix der Neuinfektionen wird mit F , die der Transfere mit V bezeichnet. Die Basisreproduktionszahl ist dann definiert, als der Spektralradius der Nächste-Generations-Matrix FV^{-1} , ausgewertet am krankheitsfreien Gleichgewichtspunkt E_{DFE} . Damit ergibt sich die

Basisreproduktionszahl \mathcal{R}_0 zu:

$$\mathcal{R}_0 = \sqrt{\frac{S_{hDFE} \beta_i}{N_h (\eta_h + \mu_h)}} \cdot \sqrt{\frac{S_{vDFE} \tilde{\beta}_i}{N_v (\mu_v + c_v)}}$$

Mit $S_{hDFE} = N_h$ und $S_{vDFE} = M_v^* = \frac{\alpha k N_h \mathcal{M}}{\varphi (\mu_v + c_v)}$, $\mathcal{M} = \varphi \eta_A - (\eta_A + \mu_A + c_A) (\mu_v + c_v)$ kann die Basisreproduktionszahl \mathcal{R}_0 auch wie folgt geschrieben werden:

$$\mathcal{R}_{0,i} = \sqrt{\frac{k \alpha \beta_i \tilde{\beta}_i \mathcal{M}}{\varphi m (\mu_v + c_v)^2 (\eta_h + \mu_h)}}$$

Für die Stabilität des krankheitsfreien Gleichgewichtspunkt E_{DFE} ist entscheidend, ob die Basisreproduktionszahl \mathcal{R}_0 den Schwellwert 1 unterschreitet. Gilt $\mathcal{R}_0 \leq 1$ so ist der krankheitsfreie Gleichgewichtspunkt lokal asymptotisch stabil. Für $\mathcal{R}_0 > 1$ ist dieser instabil.

2.2 Numerische Simulation

Die numerische Simulation erfolgt mit festen Werten für die Kontrollparameter der Larvizide, Adultizide und

mechanische Kontrolle. Die Berechnung erfolgt mittels Matlab mit dem solver ode15s. Es wird angenommen, dass die Werte für β^i, β^{ij} und $\tilde{\beta}^i$, welche die Bissrate und die Übertragungswahrscheinlichkeit kombiniert repräsentieren, für beide Serotypen gleich ist. Weiter werden folgende Parameter für die Simulationen verwendet:

$$N_h = 383\,000, \beta^i = \frac{3}{10}, \mu_h = \frac{1}{80 \cdot 365}, \eta_h = \frac{1}{3},$$

$$\mu_A = \frac{1}{4}, \eta_A = \frac{8}{100}, \mu_v = \frac{1}{12}, \varphi = 6, m = 3, k = 3.$$

Die Gleichgewichtspunkte errechnen sich damit zu folgenden Werten:

- Für die Kompartimente der Gleichgewichtspunkte E_{DFE}, E_1 und E_2 :
 $S_h^* \doteq 99\,571, I_h^* \doteq 29, R_h^* \doteq 436, \tilde{S}_h^* = 282\,964$
 $A_v^* \doteq 1\,037\,765, M_v^* \doteq 673\,145, S_v^* \doteq 673\,021,$
 $I_v^* \doteq 124,$
- Für die Kompartimente des endemischen Gleichgewichtspunktes der Koexistenz:
 $S_h^c \doteq 57\,281, I_h^c \doteq 17, \hat{R}_h^* \doteq 251,$
 $\tilde{S}_h^c \doteq 42\,307, \hat{I}_h^c \doteq 12, R_h^c \doteq 240\,545$
 $A_v^c \doteq 1\,037\,765, S_v^c \doteq 672\,897, I_v^c \doteq 124$

Mit $\mathcal{R}_0 = 1.38$ ist der krankheitsfreie Gleichgewichtspunkt instabil und die endemischen Gleichgewichtspunkte lokal stabil. Das bedeutet, dass bereits wenige Infizierte Menschen einen großen Ausbruch des Dengue Fiebers verursachen können. Es wird daher die Entwicklung des Modells betrachtet für den Fall, dass ein mit Serotyp 1 und elf mit Serotyp 2 infizierte Menschen (z.B. durch Auslandsreisen) in die Referenz-population eingehen. Die Mückenpopulation wird als noch gering angenommen, womit sich folgende Startwerte ergeben:

$$S_h(0) = N_h - 12, I_h^1(0) = 11, I_h^2(0) = 1, I_h^{12}(0) = 0,$$

$$I_h^{21}(0) = 0, S_h^{21}(0) = 0, S_h^{12}(0) = 0$$

$$R_h^1(0) = 0, R_h^2(0) = 0, R_h(0) = 0;$$

$$A_v(0) = 500, S_v(0) = 300, I_v^1(0) = 0, I_v^2(0) = 0.$$

Als untersuchter Zeitraum werden 365 Tage gewählt. Die graphische Darstellung ist in Abb. 2 zu sehen. Wie ersichtlich wächst die Mückenpopulation rasant auf ihr Gleichgewichtsniveau an und bleibt stabil bis, mit einem Versatz von etwa 90 Tagen, ein Ausbruch des Dengue Fiebers mit Serotyp 1 zu registrieren ist. Ein Ausbruch des Dengue Fiebers mit Serotyp 2 ist

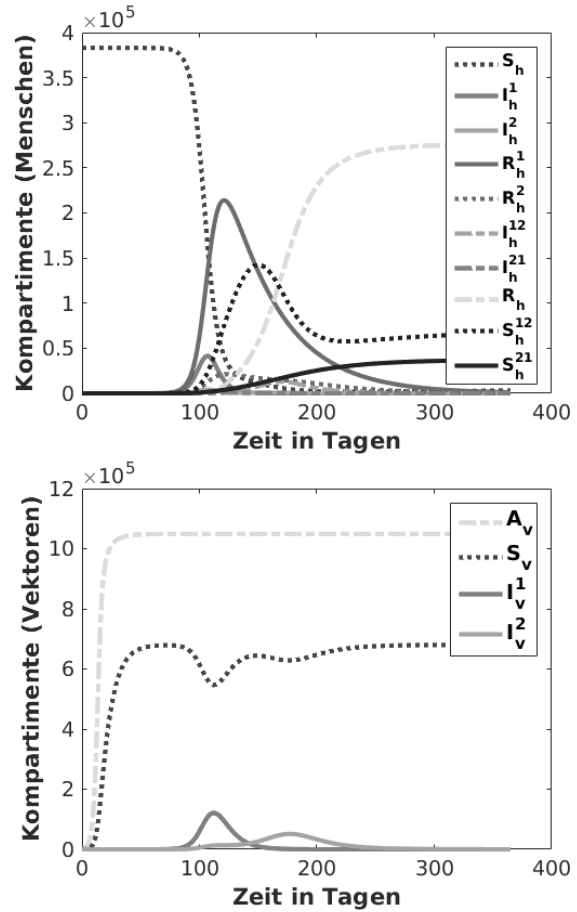


Figure 2: Numerische Simulation des Verlaufs bei Einbringung weniger Infizierter Menschen mit aktiven Kontrollparametern der Mücken in ein Zwei-Serotypen-Modell mit temporärer Kreuzimmunität, oben die Kompartimente der Menschen, unten die Kompartimente der Mücken

verzögert und abgeschwächt nach etwa 160 Tagen zu verzeichnen. In Absolutwerten entspricht dies rund 41 600 infizierte Menschen mit Serotyp 1 und 4 000 infizierte Menschen mit Serotyp 2. Bei den Zweitinfektionen sind es 12 920 die als zweites mit Serotyp 2 infiziert und 850 die als zweites mit Serotyp 1 infiziert wurden.

Wird der zeitliche Ausbruch der infizierten Menschen mit Serotyp 2 beobachtet, kann im Vergleich zu einem Zwei-Serotypen-Modell ohne die temporäre Kreuzimmunität, wie in Abb. 3 dargestellt, eine zeitliche Verschiebung des Ausbruchs beobachtet werden. Viel interessanter ist noch die Auswirkung auf die Anzahl der infizierten Menschen. Bei einem Zwei-Serotypen-Modell

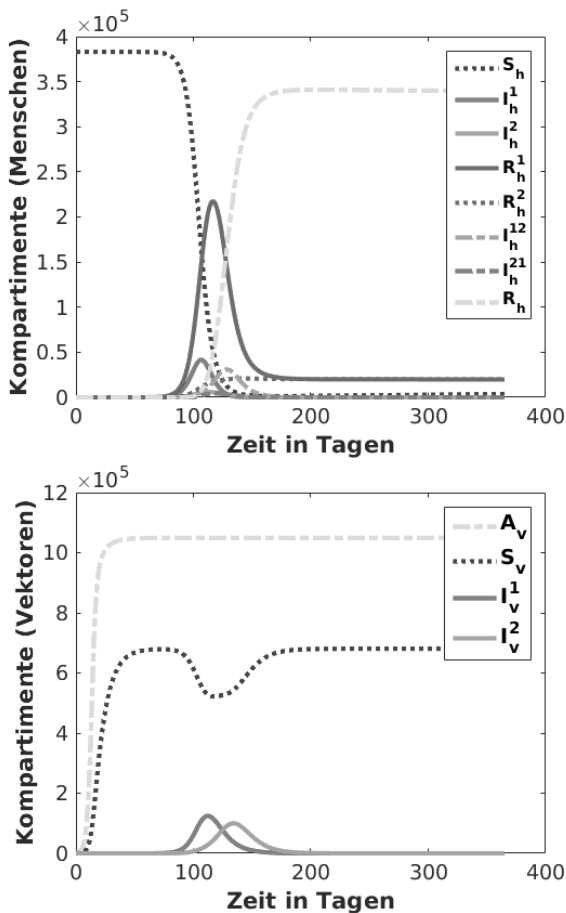


Figure 3: Numerische Simulation des Verlaufs bei Einbringung weniger Infizierter Menschen mit aktiven Kontrollparametern der Mücken in ein Zwei-Serotyp Modell ohne temporäre Kreuzimmunität, oben die Kompartimente der Menschen, unten die Kompartimente der Mücken

ohne temporärer Kreuzimmunität ergeben sich rund 41 550 mit Serotyp 1 infizierte Menschen und 59 500 mit Serotyp 2 infizierte Menschen. Bei den Zweitinfektionen sind es 31 050 die als zweites mit Serotyp 2 infiziert und 4 500 die als zweites mit Serotyp 1 infiziert wurden. Es zeigt sich, dass durch die realitätsnähere Modellierung mit einer temporären Kreuzimmunität vor allem eine Reduktion der Zweitinfektionen zu verzeichnen ist. Dies kann auf die Wechselwirkung der zeitlichen Einwirkung und der Dynamik der Mückenpopulation zurückgeführt werden.

2.3 Optimale Steuerung

Für die optimale Steuerung wird nun folgendes Kostenfunktional betrachtet.

$$J = \int_0^{t_f} \left[\gamma_1 \cdot \left(\frac{\sum (I_h^i(t) + I_h^{ij})}{N_h} \right)^2 + \gamma_2 \cdot c_v(t)^2 + \gamma_3 \cdot c_A(t)^2 + \gamma_4 \cdot (1 - \alpha(t))^2 \right] dt$$

Dabei werden die nichtlinearen Differentialgleichungen diskretisiert und mittels dem impliziten Euler Verfahren modelliert. Das hieraus resultierende nichtlineare Differentialgleichungssystem wird mit der Modellsprache AMPL und dem Löser IPOPT gelöst. Es werden drei verschiedene Fälle für die Gewichtung der einzelnen Terme verglichen, Tab. 1.

	Gewichte	Kosten
Fall A	$\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.25$	0.097862
Fall B	$\gamma_1 = 0.70, \gamma_2 = 0.10, \gamma_3 = 0.10, \gamma_4 = 0.10$	0.172594
Fall C	$\gamma_1 = 0.10, \gamma_2 = 0.30, \gamma_3 = 0.30, \gamma_4 = 0.30$	0.044721

Table 1: Übersicht der Gewichte und der Kosten des Kostenfunktionals für die drei Fälle

Es werden hierbei unterschiedliche Szenarien betrachtet. Fall A stellt die Referenz dar, in dem alle Gewichte und damit die relativen Kosten gleich bewertet sind. Fall B beschreibt die Situation, in welcher die Behandlung der infizierten Menschen am teuersten ist. Der dritte Fall (Fall C) zeigt hingegen die Situation, wenn die Kosten für die Mücken- und Larvenbekämpfung die der Behandlung der infizierten Menschen übersteigt.

Sowohl bei den Steuerparametern, als auch bei den Entwicklungen der infizierten Kompartimente ist der Einfluss der Einbringung von unterschiedlich vielen Infizierten deutlich erkennbar.

Bei den Steuerparametern der Larvizide, der Adultizide und der mechanischen Kontrolle ist deutlich zu sehen, dass es zu Beginn sehr steil steigt, dann abfällt und wenn der zweite Peak in der Infizierten Welle einsetzt nochmals steigt. Dabei ist zu beachten, dass der zweite Peak nicht mehr so stark ausfällt wie der erste. Dieser Verlauf ist bei allen drei Fällen zu beobachten. Für den Fall mit den hohen Kosten für die Kontrollparameter flachen beide Maxima ab und der zweite lokale Hoch-

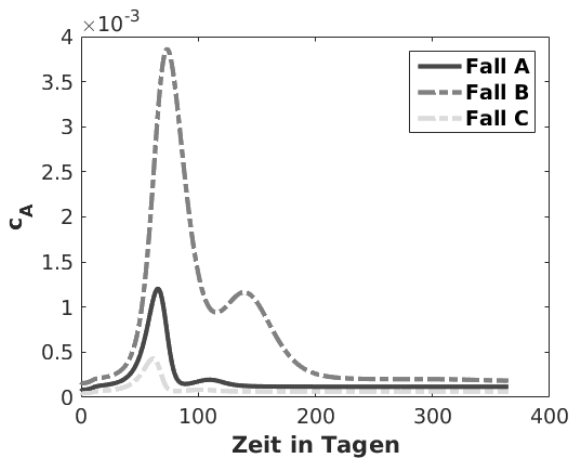


Figure 4: optimaler Steuerparameter des Larvizids c_A , bei einem Zwei-Serotyp-Modell mit temporärer Kreuzimmunität

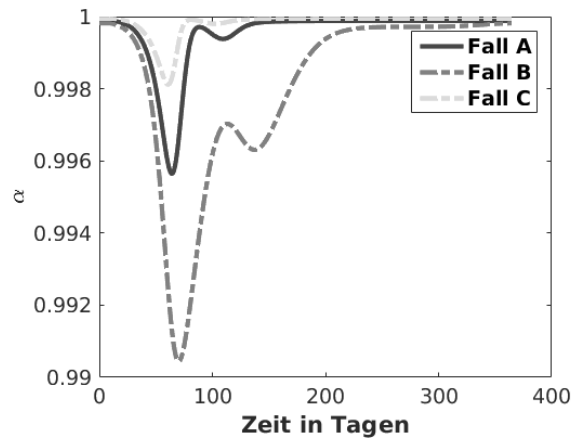


Figure 6: optimaler Steuerparameter der mechanischen Kontrolle α , bei einem Zwei-Serotyp-Modell mit temporärer Kreuzimmunität

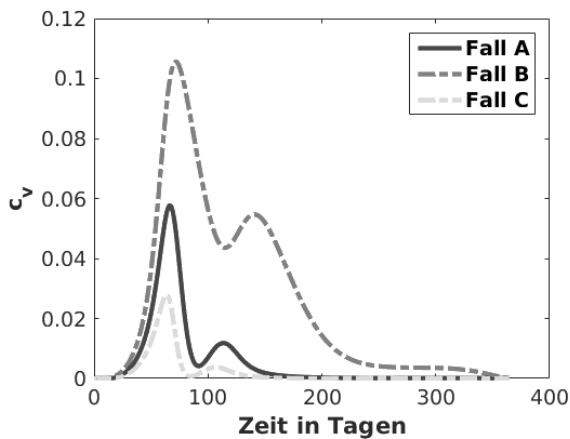


Figure 5: optimaler Steuerparameter des Adultizids c_v , bei einem Zwei-Serotyp-Modell mit temporärer Kreuzimmunität

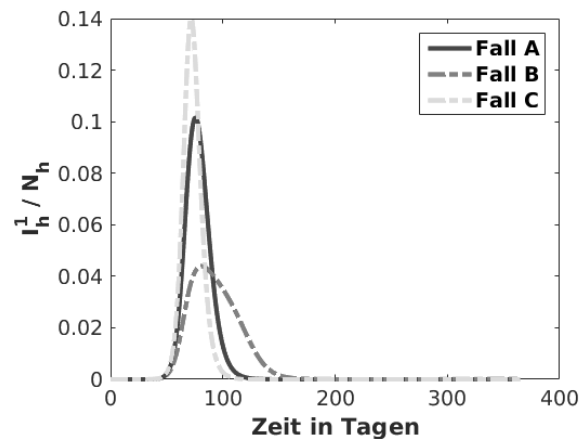


Figure 7: Vergleich der Ausschlagshöhe der mit Serotyp 1 infizierten Menschen in den verschiedenen Fällen, bei einem Zwei-Serotyp-Modell mit temporärer Kreuzimmunität

punkt verschwindet fast gänzlich.

In Bezug auf die infizierten Kompartimente zeigt sich ein komplexeres Verhalten. Obwohl die Infizierung mit Serotyp 1 als gleich wahrscheinlich wie die Infizierung mit Serotyp 2 angenommen wird, weist vor allem Fall C, in dem alle Steuerungen gegen die Mücken teurer als die Behandlung der infizierten sind, den höchsten Ausschlag für die infizierten Menschen auf. Am wenigsten Infizierte treten in Fall B auf, in welchem die Kosten der Infizierten die der Steuerparameter der Mücken übersteigt. Hier erwähnenswert ist ein deutlicher zweiter Ausbruch der Zweit-

infektionen mit Serotyp 1. Damit ist zu erkennen, dass eine einmal ausgebrochene Krankheitswelle nicht vor einer weiteren schützt. Dies unterstreicht die Komplexität des Systems, in welchem das Zusammenspiel der Parameter und die Verteilung der Infizierten signifikante Auswirkungen auf die Entwicklung der Krankheit haben. Werden diese Ergebnisse nun mit einem Zwei-Serotypen-Modell ohne temporärer Kreuzimmunität verglichen, kann festgestellt werden, dass die zwei Maxima der Steuerparameter c_A, c_v und α näher zusammenrücken. Gleichzeitig wird der absolute

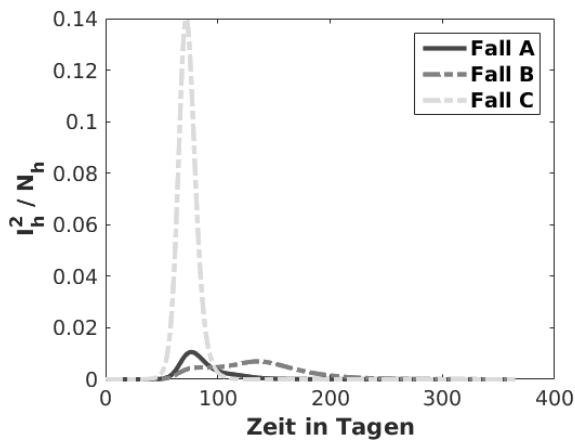


Figure 8: Vergleich der Ausschlagshöhe der mit Serotyp 2 infizierten Menschen in den verschiedenen Fällen, bei einem Zwei-Serotyp-Modell mit temporärer Kreuzimmunität

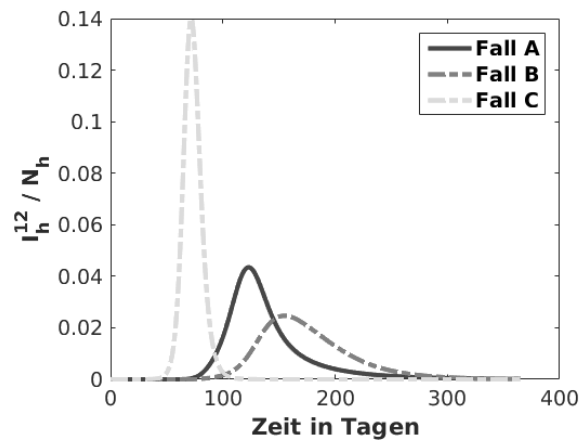


Figure 10: Vergleich der Ausschlagshöhe der mit Serotyp 2 zweitinfizierten Menschen in den verschiedenen Fällen, bei einem Zwei-Serotyp-Modell mit temporärer Kreuzimmunität

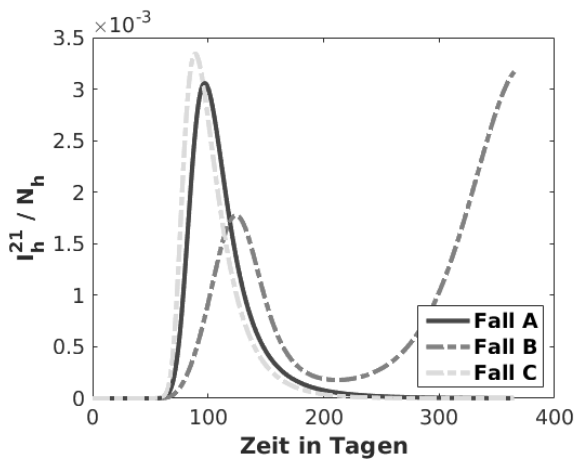


Figure 9: Vergleich der Ausschlagshöhe der mit Serotyp 1 zweitinfizierten Menschen in den verschiedenen Fällen, bei einem Zwei-Serotyp-Modell mit temporärer Kreuzimmunität

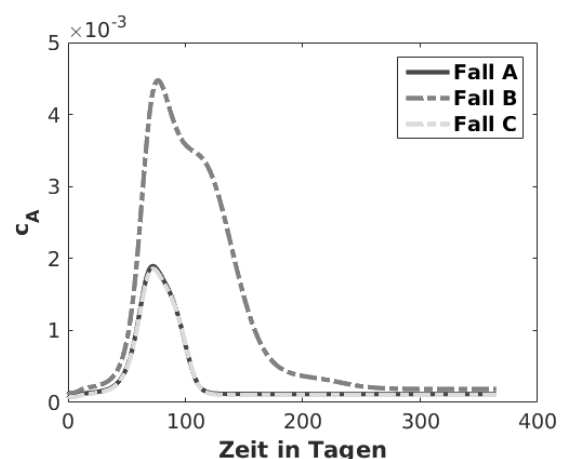


Figure 11: optimaler Steuerparameter des Larvizids c_A , bei einem Zwei-Serotyp-Modell ohne temporärer Kreuzimmunität

Ausschlag größer. Damit lässt sich feststellen, dass das Einbeziehen einer temporären Kreuzimmunität Einfluss auf den Auftrittszeitpunkt und die Ausschlagshöhe hat.

Bei Betrachtung der infizierten Kompartimente ist auch hier eine Verschiebung und eine Veränderung der Absolutzahlen ersichtlich. Sind es den Kompartimente der Erstinfektionen noch geringe Unterschiede, so zeigt der Vergleich der Zweitinfektion bei Berücksichtigung der temporären Kreuzimmunität deutliche Unterschiede. Die maximal Werte sind in Tab. 2 zusammen-

gefasst dargestellt.

Abschließend kann damit festgehalten werden, dass das Einbeziehen einer temporären Kreuzimmunität merklichen Einfluss auf den Verlauf und auf die Absolutwerte der infizierten Kompartimente nimmt. Weiter ist auch eine Auswirkung der temporären Kreuzimmunität auf die Kontrollparamtern der Mücken evident. Es sind daher weiterführende Untersuchung notwendig, um detailliertere Zusammenhänge zu identifizieren.

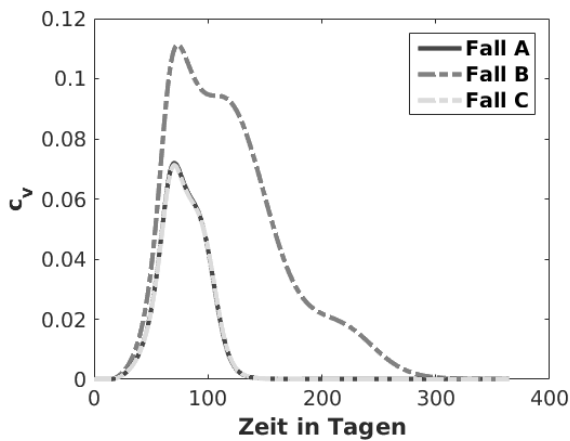


Figure 12: optimaler Steuerparameter des Adultizids c_v , bei einem Zwei-Serotyp-Modell ohne temporärer Kreuzimmunität

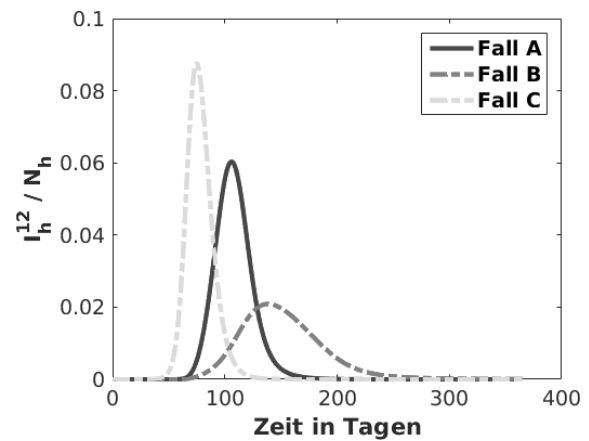


Figure 14: Vergleich der Ausschlagshöhe der mit Serotyp 2 zweifinfizierten Menschen in den verschiedenen Fällen, bei einem Zwei-Serotyp-Modell ohne temporärer Kreuzimmunität

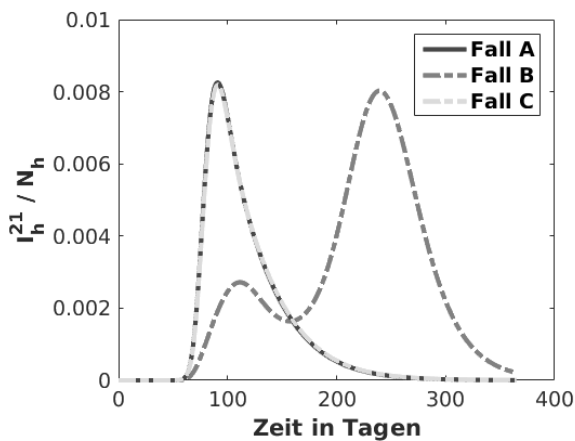


Figure 13: Vergleich der Ausschlagshöhe der mit Serotyp 1 zweifinfizierten Menschen in den verschiedenen Fällen, bei einem Zwei-Serotyp-Modell ohne temporärer Kreuzimmunität

Kompartiment	Fall	Wert in %
I_h^{12}	Fall A _{mit Kreuzim.}	4.35
I_h^{12}	Fall A _{ohne Kreuzim.}	6.03
I_h^{21}	Fall A _{mit Kreuzim.}	0.31
I_h^{21}	Fall A _{ohne Kreuzim.}	0.83
I_h^{12}	Fall B _{mit Kreuzim.}	2.47
I_h^{12}	Fall B _{ohne Kreuzim.}	2.09
I_h^{21}	Fall B _{mit Kreuzim.}	0.32
I_h^{21}	Fall B _{ohne Kreuzim.}	0.80
I_h^{12}	Fall C _{mit Kreuzim.}	4.57
I_h^{12}	Fall C _{ohne Kreuzim.}	6.00
I_h^{21}	Fall C _{mit Kreuzim.}	0.33
I_h^{21}	Fall C _{ohne Kreuzim.}	0.82

Table 2: Vergleich der infizierten Kompartimente der Zweitinfektionen bei einem Zwei-Serotyp-Modell mit und ohne temporärer Kreuzimmunität in den Fällen A, B und C

References

- [1] Becker N, Schön S, Klein A, Ferstl I, Kizgin A, Tannich E, Kuhn C, Pluskota B, Jöst A. First mass development of *Aedes albopictus* (Diptera: Culicidae)—its surveillance and control in Germany. *Parasitology Research*. 2017;116(3):847–858.
- [2] Kampen H, Kronefeld M, Zielke D, Werner D. Further specimens of the Asian tiger mosquito *Aedes albopictus* (Diptera, Culicidae) trapped in southwest Germany. *Parasitology research*. 2013;112(2):905–907.
- [3] Werner D, Kampen H. *Aedes albopictus* breeding in southern Germany, 2014. *Parasitology research*. 2015; 114(3):831–834.
- [4] Thomas SM, Tjaden NB, Frank C, Jaeschek A, Zipfel L, Wagner-Wiening C, Faber M, Beierkuhnlein C, Stark K. Areas with High Hazard Potential for Autochthonous Transmission of *Aedes albopictus*-Associated Arboviruses in Germany. *International*

- journal of environmental research and public health.* 2018;15(6).
- [5] Becker N, Krüger A, Kuhn C, Plenge-Bönig A, Thomas S, Schmidt-Chanasit J, Tannich E. Stechmücken als Überträger exotischer Krankheitserreger in Deutschland. *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz.* 2014;57(5):531.
- [6] Thomas SM, Tjaden NB, van den Bos S, Beierkuhnlein C. Implementing cargo movement into climate based risk assessment of vector-borne diseases. *International journal of environmental research and public health.* 2014;11(3):3360–3374.
- [7] Zur Situation bei wichtigen Infektionskrankheiten – Reiseassoziierte Krankheiten 2015. *Epidemiologisches Bulletin – Robert Koch-Institut.* 2016;(39):431–446.
- [8] Frank C, Faber M, Hellenbrand W, Wilking H, Stark K. Wichtige, durch Vektoren übertragene Infektionskrankheiten beim Menschen in Deutschland. *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz.* 2014;57(5):557.
- [9] Scholte EJ, Dijkstra E, Blok H, De Vries A, Takken W, Hofhuis A, Koopmans M, De Boer A, Reusken C. Accidental importation of the mosquito *Aedes albopictus* into the Netherlands: a survey of mosquito distribution and the presence of dengue virus. *Medical and veterinary entomology.* 2008;22(4):352–358.
- [10] Takken W, Knols BG. *Emerging pests and vector-borne diseases in Europe*, vol. 1. Wageningen Academic Pub. 2007.
- [11] Stanaway JD, Shepard DS, Undurraga EA, Halasa YA, Coffeng LE, Brady OJ, Hay SI, Bedi N, Bensenor IM, Castañeda-Orjuela CA, et al. The global burden of dengue: an analysis from the Global Burden of Disease Study 2013. *The Lancet infectious diseases.* 2016; 16(6):712–723.
- [12] Benelli G, Mehlhorn H. Declining malaria, rising of dengue and Zika virus: insights for mosquito vector control. *Parasitology research.* 2016;115(5):1747–1754.
- [13] Esteva L, Vargas C. Analysis of a dengue disease transmission model. *Mathematical Biosciences.* 1998; 150(2):131 – 151.
- [14] Pinho STRd, Ferreira CP, Esteva L, Barreto FR, e Silva VM, Teixeira M. Modelling the dynamics of dengue real epidemics. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences.* 2010;368(1933):5679–5693.
- [15] Albrecht G, Fischer A, Chudej K. Analyse, Simulation und Optimale Steuerung eines mathematischen Dengue-Fieber Modells mit Impfung. In: *Tagungsband Workshop 2018 ASIM/GI-Fachgruppen in Hannover*, edited by Loose T. ARGESIM Verlag Wien. 2018; pp. 223–229.
- [16] Rodrigues H. Optimal Control and Numerical Optimization Applied to Epidemiological Models. PhD, Universidade de Aveiro, Portugal. 2012.
- [17] Rodrigues HS, Monteiro MTT, Torres DF, Zinober A. Dengue disease, basic reproduction number and control. *International Journal of Computer Mathematics.* 2012;89(3):334–346.
- [18] Feng Z, Velasco-Hernández JX. Competitive exclusion in a vector-host model for the dengue fever. *Journal of Mathematical Biology.* 1997;35(5):523–544.
- [19] Esteva L, Vargas C. Coexistence of different serotypes of dengue virus. *Journal of Mathematical Biology.* 2003;46(1):31–47.
- [20] Chudej K, Fischer A. Optimal vaccination strategies for a new dengue model with two serotypes. *IFAC-PapersOnLine.* 2018;51(2):13–18.

Analysis of Practical Reasons for Training Simulators in Manufacturing

Benjamin Knoke^{1*}, Christian Gorltd¹, Klaus-Dieter Thoben¹

¹ BIBA – Bremer Institut für Produktion und Logistik GmbH at the University of Bremen, Hochschulring 20, 28359 Bremen, Germany; *kno@biba.uni-bremen.de

Abstract. The objective of this research is to identify the reasoning behind the use of training simulators for manufacturing processes and to discuss potential applications. Training Simulators are a tool that is successfully applied to vocational training processes within multiple areas, such as medicine, aviation or military. Research from these areas provides practical reasons for the applications of training simulators that are used to structure a discussion on potential applications in manufacturing. The practical reasons are taken from literature and evaluated towards existing applications in manufacturing.

Introduction

The ongoing transformation of production environments towards cyber-physical systems creates increasing demands towards the skills of workers and the vocational training [1]. The desired skillsets encompass technical and functional expertise, as well as application- and process-oriented skills [2]. Training simulations have been proven to be a viable approach to create and improve these skills, and are commonly applied in multiple areas [3], [4], [5].

Simulation-based training has become an integral part in diverse sectors, such as medicine [6], aviation [7], or firefighting [8], but only few applications exist for manufacturing processes [9]. The scope of this paper is to analyse if the reasons for the integration of simulations-based from these areas can be applied to manufacturing.

1 Related Work

The research is based on the reasons towards the use of training simulators from the aviation and medical areas. These reasons are used to structure a discussion of exist-

ing applications in manufacturing with the aim to identify the practical reasons for the integration of training simulators in manufacturing processes on a general level. The term practical reasons encompasses those reasons that result in a measurable benefit and excludes subjective benefits such as a more innovative image.

1.1 Reasoning for the application of training simulations

The following practical reasons why simulation-based training can be preferable compared to training within the real system have been extracted from literature on simulation-based training in aviation [10]:

1. Training within the real system is too dangerous,
2. The desired conditions are not available,
3. Training on the real system is too expensive,
4. There is insufficient time available on the real system,
5. Required circumstances do not occur frequently enough,
6. Environmental regulations restrict training possibilities,
7. Safety regulations prohibit training of certain tasks.

Lateef (2010) lists the following benefits of simulation-based training for medical education [3]:

1. Deliberate practice with feedback
2. Exposure to uncommon events
3. Reproducibility
4. Opportunity for assessment of learners
5. The absence of risks to patients

When comparing both sources, it appears that the latter focusses on educational benefits and does not include expenses, but adds an assessment and feedback system to the list.

1.2 Training simulations for manufacturing processes

Training simulators can be categorized according to the type of skills that are focussed during training. One established taxonomy uses the following categories [11], [12]:

- Psychomotor: Technical and functional expertise training.
- Cognitive: Problem-solving and decision-making skills.
- Affective: Interpersonal and communications skills or team-based competencies.

This sub-section discusses two existing applications of training simulators within the industrial area in order to determine, if the previously described reasons apply. While a welding simulator focusses on psychomotor skills, a CNC machining simulator trains mostly cognitive skills.

Simulations for interpersonal and communication skills, such as business games, are less specific to manufacturing processes and are not within the scope of this research.

Welding simulator

The current key simulation technologies in welding training are Virtual Reality and Augmented Reality. During the past decade these technologies were applied to create welding simulators and made the transition towards practical application [13], [14].

These simulators rely on optic measurement to capture the position and movement of a welding torch or electrode [15]. The measured characteristics include stick out, work angle, travel angle, travel path, and travel speed. The available welding simulators differ greatly in physical and functional fidelity, as well as in price.

The applied simulator system is the Soldamatic training simulator for welders (Fig. 1), which focusses on manual welding processes and is a commercial product. The system has been observed in two case studies, a plant for the construction of car chassis of a car manufacturer and an organisation for joint training of metal processing SMEs [16]. Both organisations resort to simulation-based training for the first 20-35% of their welding courses.

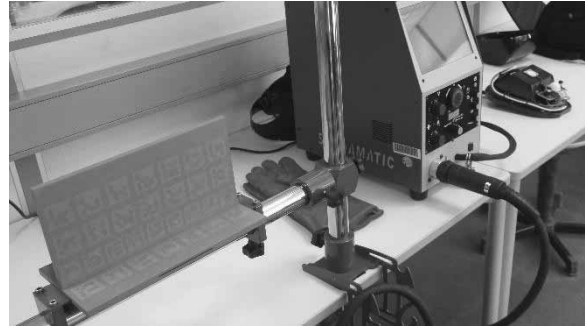


Figure 1: Soldamatic welding simulator

In comparison to the reasons for simulation-based training extracted from literature (section 1.1), the following reasons were expressed within the case studies:

1. Training within the real system is too dangerous
The ability to conduct training in a controlled environment is considered to be one of the key reasons for simulation-based training in welding. This stems from the high temperatures and radiation hazards that are emitted in the real system and that are eliminated in the simulators.
2. The desired conditions are not available:
This issue does not seem to apply to welding training in either of the case studies, because the training on the real system is conducted in welding booths with relatively stable conditions.
3. Training on the real system is too expensive
When compared to training on the real system, the simulation-based training does consume significantly less resources. Training on the real system relies on metal, inert gas, and other welding consumables. Although the simulator requires a significant investment, it is expected to amortise within 1-3 years, depending on the utilisation ratio and other financial factors.
4. There is insufficient time available on the real system
Access to welding gear was not stated to be a limitation to training in both case studies.
5. Required circumstances do not occur frequently enough
Welding training in both case studies encompasses support processes to prepare work pieces for the following exercise, which includes cutting or breaking and grinding. The duration of these activities depends great-

ly on the availability of tools and takes up a significant amount of time. Opposed to this, the simulation-based training can be restarted within a few seconds. This leads to much shorter training cycles.

6. Environmental regulations restrict training possibilities

Environmental regulations were not a hindering factor to welding training in either case study.

7. Safety regulations prohibit training of certain tasks

The use of welding tools requires protection from hazards such as high temperatures, fumes, and radiation. This prohibits applications in unfitted classroom or fair settings. Here, a simulator can be used and the absence of the typical hazards also allows the modulation of welding parameters to demonstrate the outcome without risk.

In line with the benefits of simulation-based training for medical education (section 1.1), an additional issue was extracted from the case studies that is roughly related to safety regulations:

8. Objective feedback is not provided by the real system
Welding training is typically performed in an enclosed welding booth and the trainee receives feedback from the trainer after the weld is performed based on the resulting workpiece. The received criticism can be a burden to the relationship in some cases. When simulation-based training is applied, the feedback is supported by objective sensor data that is usually accepted by the trainee and can be visualized while the weld is performed. The hazard-free training also allows for the trainer to correct the trainee's posture and teach basic ergonomic principles during the training.

CNC Machining simulator

Simulations for machining tools are designed to train the required skills for the real system without the need of an actual machining process or a complete machining centre. Simulators usually focus on machining centres under automatic control that utilize a computerised numeric control (CNC). Training simulators for these devices typically consist of an input panel or control console that can also be found on an actual machining centre and a display to visualise the simulation. However, there are also examples of simulations that are solely consisting of PC based software or web applications [17], [18].

In regard of the reasoning behind the use of training simulators, the following statements can be extracted from literature.

1. Training within the real system is too dangerous

Training on CNC machine tools is not considered to be particularly dangerous, but the elimination of any injury risk is mentioned as a benefit in some cases [19].

2. The desired conditions are not available

Machining processes are usually conducted in factories and are designed to not depend on external conditions.

3. Training on the real system is too expensive

Cost reduction can be considered to be the main reason for the virtualisation of CNC machine training [20], [21], [22].

4. There is insufficient time available on the real system

In the case of a CNC machine centre, limitation of time is not a stand-alone issue. The time that is available for training is mostly limited through costs of machine hours and loss of production, which links it to the reason of being too expensive. Machine downtimes do not seem to impede training.

5. Required circumstances do not occur frequently enough

Most CNC machining simulators include a fast-forward option during the machining process or a direct jump to the finished product (e. g. CNCsimulator1). This is expected to allow for more time efficient training sessions.

6. Environmental regulations restrict training possibilities

There are no indications of environmental regulations as an argument against training on the real system.

7. Safety regulations prohibit training of certain tasks

Safety regulations are no significant barrier to training on a CNC machining centre.

8. Objective feedback is not provided by the real system

The training simulation can be designed to constantly provide feedback on the outcome, even while changes

¹ <http://cncsimulator.info>

are made [23], [24]. This is a benefit over training on the real system and facilitates the acquisition of problem-solving and decision-making skills.

1.3 Summary of the reasoning behind both applications

The reasoning to prefer training simulations over training solely on the real system is summarized in table 1 for both applications. The degree to which the reasoning behind flight simulators applies is rated for both applications with low (-), medium (o), or high (+).

Issue	Welding	CNC Machining
Training within the real system is too dangerous	+	o
The desired conditions are not available	-	-
Training on the real system is too expensive	o	+
There is insufficient time available on the real system	-	-
Required circumstances do not occur frequently enough	+	+
Environmental regulations restrict training possibilities	-	-
Safety regulations prohibit training of certain tasks	+	-
Objective feedback is not provided by the real system	+	o

Table 1: Applicability of the reasoning behind flight simulators for applications in manufacturing

The deviations between both applications can likely be attributed to the different skill focus. Safety concerns are more apparent in processes that rely on psychomotor skills than in those with a focus on cognitive skills. Additionally, automatized machines, such as machining centres are more expensive to acquire and operate than tools for manual labour, such as welding equipment.

2 Applicability of Training Simulators in Manufacturing

The applicability of training simulators for manufacturing processes is discussed in the following, based on the

reasoning that has been derived from literature in section 1.1.

2.1 Training within the real system is too dangerous

When compared to aviation, mistakes made within manufacturing processes certainly have a less fatal impact. However, accidents happen and especially in areas that rely on manual labour and psychomotor skills. The most dangerous activities include climbing on structures or scaffolding, or the handling of powered tools, such as saws [25]. Depending on the severity of these hazards, it can be preferable to train inexperienced workers in a simulated environment.

2.2 The desired conditions are not available

Contrary to a pilot, who needs to train a certain task under specific conditions (i. e. crosswind landing), industry workers are rarely required to perform under environmental conditions that would require particular training. Exceptions may include emergency protocols, such as in cases of a sudden change in the weather (i. e. emergency evacuation during offshore wind farm assembly), or technical failures (i. e. faulty air conditioning system in a paint factory).

2.3 Training on the real system is too expensive

Cost reduction is a major reason for the application of training simulators. Distributors of welding simulators, as well as CNC machining simulators, strongly advertise with a long-term benefit through lower running costs. When the manufacturing task is simulated, the costs of work pieces and other resources are greatly reduced or eliminated completely.

The running costs of vocational training determine the investment in simulation-based training that commercially operating organisations are willing to take. While the costs of professional flight simulators can reach up to multiple million euros, training simulators in manufacturing may cost significantly less.

2.4 Insufficient time available on the real system

Time can be a limiting factor to training on the real system when necessary tools or machines are not available due to maintenance downtimes, high workload, or

because they have not been acquired yet.

2.5 Required circumstances do not occur frequently enough

Both industrial simulators that have been described aim to speed up activities that do not directly contribute to the improvement of desired skillsets in order to increase the rate of training sessions.

The welding simulator is an example of a training simulator that focuses on manual labour during the process and skips set-up and post-processing.

CNC machining simulators, on the other hand, are designed to improve programming skills that are applied during the set-up phase. Therefore, many simulations include an option to fast-forward through the actual machining process or to skip directly to the result.

The impact of this factor rises with waiting time or time taken for ancillary tasks.

2.6 Environmental regulations restrict training possibilities

Both applications from manufacturing appear to characterize environmental regulations as a minor issue regarding training possibilities. Exceptions may include processes that require a significant amount of resources, or involve the handling or creation of hazardous waste, noise or radiation.

2.7 Safety regulations prohibit training of certain tasks

Safety regulations are no hard barrier to training in manufacturing. However, many jobs involve mandatory personal protective equipment that complicates training in specific settings such as classrooms or demonstrations on job fairs. Training simulators can be designed to allow for training without the strict need for these safety measures.

2.8 Objective feedback is not provided by the real system

The ability to receive objective feedback while performing the training is one of the major reasons for simulation-based training in both described applications from manufacturing.

Direct feedback is possibly more important in processes that require psychomotor skills to correct the posture or the handling of a tool. However, immediate

and sensor-based feedback has been observed to support communication between trainer and trainee [16], as well as provide an educational benefit [3].

3 Conclusion and Outlook

The analysis has shown that the reasons for simulation-based training in manufacturing differ from those in aviation or medicine. The following observations have been made:

- When compared to aviation or medicine, resource orientation is more important in manufacturing. System availability, the simulation of environmental conditions, the avoidance of life-threatening situations, or barring regulations are lesser factors.
- The type of skills that are trained impacts the reasoning towards simulation-based training. While processes that rely on psychomotor skills appear to be more dangerous on average, those that rely on cognitive skills tend to apply more expensive machines.
- The ability to simulate manufacturing processes eliminates the necessity of personal protective equipment and enables demonstrations in classroom settings or at job fairs.

The advance of technologies for human-machine interaction (HMI) occasionally enables new applications of simulation-based training, such as the described welding simulator [16]. Therefore, a comparison of the identified benefits of training simulators in manufacturing with the characteristics of HMI technologies and industrial requirements will be investigated in future work in order to identify potential applications and research gaps.

References

- [1] Gorecky, D., Schmitt, M., Loskyll, M., & Zühlke, D. (2014). Human-machine-interaction in the industry 4.0 era. Proceedings of the 12th IEEE International Conference on Industrial Informatics (INDIN 2014), pp. 289-294.
- [2] Ahrens, D., & Spöttl, G. (2015). Industrie 4.0 und Herausforderungen für die Quali-fizierung von Fachkräften. In Hirsch-Kreinsen, H., Ittermann, P., & Niehaus, J. Digitalisierung industrieller Arbeit. Nomos Verlagsgesellschaft mbH & Co. KG, pp. 184-205.
- [3] Lateef, F. (2010). Simulation-based learning: Just like the real thing. Journal of Emergencies, Trauma and

- Shock, 3(4), pp. 348-352.
- [4] Larnpotang, S., Lizdas, D., Rajon, D., Luria, I., Gravenstein, N., Bisht, Y., Schwab, W., Friedman, W., Bova, F., & Robinson, A. (2013). Mixed simulators: augmented physical simulators with virtual underlays. *Proceedings of the IEEE Virtual Reality (VR) 2013*, pp. 7-10.
- [5] Sohmer, B., Hudson, C., Hudson, J., Posner, G. D., & Naik, V. (2014). Transesophageal echocardiography simulation is an effective tool in teaching psychomotor skills to novice echocardiographers. *Canadian Journal of Anesthesia/Journal canadien d'anesthésie* 61(3), pp. 235-241.
- [6] Cook, D. A., Hatala, R., Brydges, R., Zendejas, B., Szostek, J. H., Wang, A. T., & Hamstra, S. J. (2011). Technology-enhanced simulation for health professions education: a systematic review and meta-analysis. *Jama* 306(9), pp. 978-988.
- [7] Sullivan, J., Yang, J. H., Day, M., & Kennedy, Q. (2011). Training simulation for helicopter navigation by characterizing visual scan patterns. *Aviation, space, and environmental medicine* 82(9), pp. 871-878.
- [8] Backlund, P., Engstrom, H., Hammar, C., Johannesson, M., & Lebram, M. (2007). Sidh-a game based firefighter training simulation. *Proceedings of the 11th International Conference on Information Visualization, 2007 (IV '07)*, pp. 899-907.
- [9] Mujber, T. S., Szecsi, T., & Hashmi, M. S. (2004). Virtual reality applications in manufacturing process simulation. *Journal of materials processing technology*, 155-156, pp. 1834-1838.
- [10] Farmer, E., Van Rooij, J., Riemersma, J., Jorna, P. & Moraal, J. (1999). *Handbook of simulator-based training*. Farnham: Ashgate.
- [11] Lateef, F. (2010). Simulation-based learning: Just like the real thing. *Journal of Emergencies, Trauma, and Shock*, 3(4), pp. 348-352.
- [12] Larnpotang, S., Lizdas, D., Rajon, D., Luria, I., Gravenstein, N., Bisht, Y., Schwab, W., Friedman, W., Bova, F. and Robinson, A. (2013). Mixed simulators: augmented physical simulators with virtual underlays. *Conference Proceedings IEEE Virtual Reality 2013*, in Orlando, Florida, pp. 7-10.
- [13] Fast, K., Gifford, T. & Yancey, R. (2004). Virtual training for welding. *Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004)* in Arlington, pp. 298-299.
- [14] White, S., Prachyabrued, M., Baghi, D., Aglawe, A., Reiners, D., Borst, C. & Chambers, T. (2009). Virtual welder trainer. *Proceedings of the IEEE Virtual Reality Conference 2009* in Lafayette, pp. 303-303.
- [15] Okimoto, M. L. L., Okimoto, P. C., & Goldbach, C. E. (2015). User experience in augmented reality applied to the welding education. *Procedia Manufacturing* 3, pp. 6223-6227.
- [16] Knoke, B. & Thoben, K.-D. (2017). Integration of Simulation-based Training for Welders. *Simulation Notes Europe* 27(1), pp. 37-44.
- [17] Qin, S. F., Harrison, R., West, A. A., & Wright, D. K. (2004). Development of a novel 3D simulation modeling system for distributed manufacturing. *Computers in industry*, 54(1), pp. 69-81.
- [18] Dalef, H. H., & Aziz, F. A. (2017). Augmented Reality Application to Monitor Wireless Data Process in Manufacturing Operation. *Journal of Computational and Theoretical Nanoscience*, 14(9), pp. 4143-4157.
- [19] Chen, F. (2007). Operation Training SimulationSystem for Computer Numerical Control (CNC) Machine. U.S. Patent Application No. 11/526,622.
- [20] Wasfy, T. M., Wasfy, A. M., El-Mounayri, H., & Aw, D. (2005, January). Virtual training environment for a 3-axis CNC milling machine. In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 1111-1120.
- [21] Xiaoling, W., Peng, Z., Zhifang, W., Yan, S., Bin, L., & Yangchun, L. (2004, June). Development an interactive VR training for CNC machining. In *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, pp. 131-133.
- [22] Lee, W. B., Li, J. G., & Cheung, C. F. (2002). Development of a virtual training workshop in ultra-precision machining. *International Journal of Engineering Education*, 18(5), pp. 584-596.
- [23] Li, J., Yao, Y., & Wu, J. (2011). CNC partner: a novel training system for NC machining. *Computer Applications in Engineering Education*, 19(3), pp. 466-474.
- [24] Fang, X. D., Luo, S., Lee, N. J., & Jin, F. (1998). Virtual machining lab for knowledge learning and skills training. *Computer Applications in Engineering Education*, 6(2), pp. 89-97.
- [25] United States Department of Labor. (2015). Incidence rate and number of nonfatal occupational injuries by selected industries and ownership, 2015. Online resource: <https://www.bls.gov/news.release/osh.t05.htm>

Ein neuer Ansatz zur Modellbildung und Simulation mit VR- und AR- Brillen am Beispiel der Fabrikplanung

Uwe Bracht¹, Marc Schlegel^{1*}

¹Institut für Maschinelle Anlagentechnik und Betriebsfestigkeit – Abteilung für Anlagenprojektierung und Materialflusslogistik (IMAB), Technische Universität Clausthal, Leibnizstr. 32, 38678 Clausthal-Zellerfeld, Deutschland; *marc.schlegel@imab.tu-clausthal.de

Abstract. Virtual- und Augmented-Reality-Brillen haben signifikante Fortschritte in ihrer technischen Leistungsfähigkeit gemacht. Dieser Fachbeitrag zeigt eine Methode, wie diese Brillen zur Unterstützung der Modellbildung und Simulation eingesetzt werden können. Dies wird anhand eines Praxisfalls in der Fabrikplanung demonstriert.

Motivation und Ausgangslage

Kürzere Produktlebenszyklen, steigende Kundenerwartungen und zunehmender Konkurrenzdruck durch die Globalisierung [1] zwingen Unternehmen dazu, ihr Produktportfolio immer häufiger zu aktualisieren [2]. Dementsprechend müssen Fabriken in immer kürzerer Zeit an die geänderten Anforderungen angepasst werden. Der Einsatz von Virtual Reality (VR) und Augmented Reality (AR) kann dabei helfen, den Prozess der Um- und Neuplanung zu beschleunigen und Planungskosten zu reduzieren [3, 4].

Die bisherige Anwendung von VR und AR in der Fabrikplanung liegt bisher weniger in der Planung oder Bearbeitung von Maschinenlayouts, sondern beschränkt sich zumeist auf die reine Visualisierung von bereits vorbereiteten Planungsvarianten [5]. Hauptgrund hierfür ist, dass es sich bei bisherigen VR- und AR-Systemen oftmals um Speziallösungen handelt, welche teuer in der Anschaffung sind und von den gängigen Materialflusssimulations – und Fabrikplanungsprogrammen nicht unterstützt werden [1,6]. Planungs- und Überprüfungsaufgaben erfordern daher einen hohen Aufwand in der Vor- und Nachbereitung.

Jetzt verfügbare VR- und AR-Brillen sind bspw. die „Vive“ von HTC sowie die „Mixed Reality“ Modelle

als auch die HoloLens von Microsoft. Diese Brillen unterscheiden sich von bisherigen VR- und AR-Lösungen durch ihre vergleichsweise geringen Anschaffungskosten und ihren hohen Verbreitungsgrad. Für diese Brillen wurden am IMAB der TU Clausthal Einsatzmöglichkeiten entwickelt und getestet [7]. Es zeigte sich, dass diese Brillen in ihrem aktuellen technischen Entwicklungsstand sowie durch ihre Unterstützung durch bestehende Soft- und Hardwareumgebungen für Planungsaufgaben als einsetzbar angesehen werden können. Eine Integration in Materialflusssimulations- und Fabrikplanungssoftware steht jedoch noch aus.

Basierend auf diesen Voruntersuchungen wurde am Institut eine Methode entwickelt, mit der diese Brillen zur Modellbildung und Unterstützung von Simulationen eingesetzt werden können. Anhand eines Praxisbeispiels aus der Fabrikplanung wird diese Methodik demonstriert.

1 Anforderungen und Entwicklung der Methodik

Die entwickelte Methodik für die Modellbildung und Simulation mit VR- und AR- Brillen soll den gesamten Planungs- und Umplanungsprozess abdecken. Beginnend mit der 3D-Datenerfassung bereits existierender Anlagen, über die Integration von 3D-Modellen in ein Simulationsprogramm und die Umplanung in VR, hin zur Überprüfung und Modifikation von Planungsvarianten am Einsatzort mit Hilfe von AR.

Im Folgenden, werden entsprechende Anforderungen aufgestellt die bei der Entwicklung der Methodik von Relevanz sind.

- Schnelle Datenaufnahme
- Hoher Wiedererkennungswert der Modelle
- Wenige Arbeitsschritte bei der Nachbereitung
- Hardware günstig in Anschaffung und Betrieb
- Kurze Einarbeitungszeit der Mitarbeiter
- Verwendung von Standardformaten
- Verbindung zwischen Simulationssoftware und dem VR- sowie dem AR- Framework

Die basierend auf diesen Anforderungen entwickelte Methodik umfasst vier Schritte (Abbildung 1).



Abbildung 1: Methodik zur smarten Datenerfassung und Umplanung

Zu Beginn steht die Datenerfassung und Aufbereitung. Für den Einsatz von VR- und AR- Brillen wird in diesem Schritt die Aufnahme aller benötigten 3D-Raumstrukturen und Maschinen sowie weiterer planungsrelevanter Objekte vorgenommen. In der nachfolgenden Datenaufbereitung werden diese 3D-Modelle von Störkonturen befreit und anschließend in das Simulationsprogramm sowie zwei verschiedene Programme zur Steuerung der VR- und AR-Brillen eingebunden (VR-AR-Frameworks).

Nach erfolgter Materialflussoptimierung im Simulationsprogramm werden Typ und Position der im Materialflussmodell verwendeten Objekte über eine bidirektionale Schnittstelle [8] in das VR-Framework übertragen. Diese bildet aus besagten Informationen eine VR-Umgebung und ermöglicht die Entwicklung eines Maschinenlayouts durch einen Fabrikplaner. Schließlich lässt sich der finale planungsstand durch die Verwendung von AR-Brillen vor Ort in der Realität überprüfen und anpassen.

2 Anwendung in der Praxis

2.1 Datenerfassung und -aufbereitung

Zur Aufnahme räumlicher Strukturen als Grundlage der Planung und virtuellen Begehung werden bisher das Aufmaß oder 3D-Laserscanner verwendet.

Das Aufmaß ist günstig in der Anschaffung benötigter Messgeräte, erfordert allerdings die vollständige Modellierung aller 3D-Modelle von Hand. Gerade bei komplexen Strukturen kann der Aufwand in einem

Maße steigen, dass es nicht möglich ist, die umzuplanende Halle mit allen notwendigen Details zu modellieren. 3D-Laserscanner wiederum liefern sehr gute Ergebnisse welche jedoch aufwändig aufbereitet werden müssen. Weiterhin sind Laserscanner in Anschaffung und Betrieb so teuer, dass sich ihr Einsatz nur für Unternehmen mit entsprechend regelmäßigem Bedarf lohnt.

Die bisherigen Untersuchungen am IMAB ergaben eine Eignung der AR-Brille Microsoft HoloLens für die Datenaufnahme von räumlichen Strukturen. Verwendet wird hierbei die Sensorik der Brille zur Positionierung von virtuellen Objekten in der Realität. Um diese Objekte zu positionieren, erstellt die Brille eigenständig ein Raummodell der Umgebung, welches sich auslesen lässt. Für die Aufnahme einer vollständigen Halle ist die einmalige Begehung notwendig. Vorteilhaft ist, dass durch die Mobilität der AR-Brille auch verdeckte Bereiche gezielt begangen und erfasst werden können. Versuchsweise wurde eine 50m x 10m große Halle zur Rohstoffaufbereitung aufgenommen. Das entsprechende 3D-Modell ist in Abbildung 2 zu sehen. Die Auflösung der erhobenen Daten liegt deutlich unter denen eines Laserscanners, ist für Fabrikplanungszwecke jedoch ausreichend.

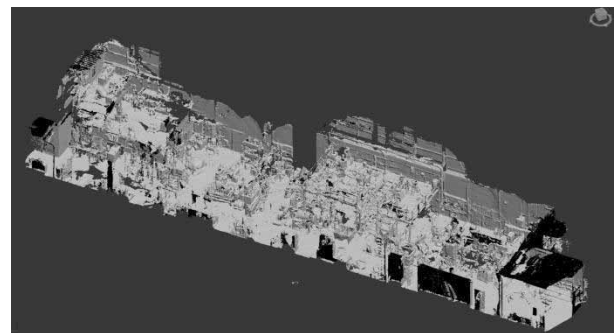


Abbildung 2: Aufnahme einer 50m x 10m Versuchs-halle zur Rohstoffaufbereitung

Zu erkennen sind alle für eine Umplanung relevanten räumlichen Strukturen wie Wege, Türen, Treppenaufgänge, Träger und Podeste sowie Bodunenregelmäßigkeiten und Wasserrinnen. Weiterhin sind die Positionen der 23 in der Halle befindlichen Maschinen zu erkennen. Die Dauer der gesamten Aufnahme betrug ca. eine Stunde. Die exportierten Modelle sind bisher einfarbig. Eine automatische Texturierung würde den Erkennungswert deutlich steigern. Programme im Prototypenstadium, die dies leisten können, sind bereits erhältlich.

Für die Aufnahme einzelner Maschinen ist ein höherer Detailgrad notwendig. Ein System, welches 3D-Modelle mit hoher Auflösung und automatischer Texturierung erzeugt, ist der iSense StructureSensor. Es eignet sich für kleinere Objekte wie Maschinen, für die Aufnahme sehr großer 3D-Objekte wie Raumstrukturen ist es jedoch nicht geeignet. Der Scan des in Abbildung 3 zu sehenden Metallschredders dauert ca. zwei Minuten.

Zur weiteren Verwendung werden die aufgenommenen 3D-Daten aus den HoloLens sowie aus dem ISense-Scanner als .obj-Datei exportiert und in einem CAD-Programm von Störkonturen bereinigt. Dieser Prozess dauert jeweils nur wenige Minuten. Anschließend kommen die gewonnenen 3D-Modelle sowohl im Simulationsprogramm „Tecnomatix Plant Simulation“ sowie in den VR- und AR-Frameworks zum Einsatz.



Abbildung 3: 3D-Detaillaufnahme eines Metallschredders

2.2 Integration in die Simulationsumgebung

Die aufgenommenen Raummodelle aus der Microsoft HoloLens werden im .wrl-Format in Plant Simulation importiert. Sie bilden so die Basis der Umplanung. Weitere Objekte, wie die separat erhobenen Maschinen, wie der Metallschredder, oder auch am PC konstruierte CAD-Modelle, werden auf gleiche Weise hinzugefügt. Abbildung 4 demonstriert die Integration der aufgenommenen Modelle in PlantSimulation.

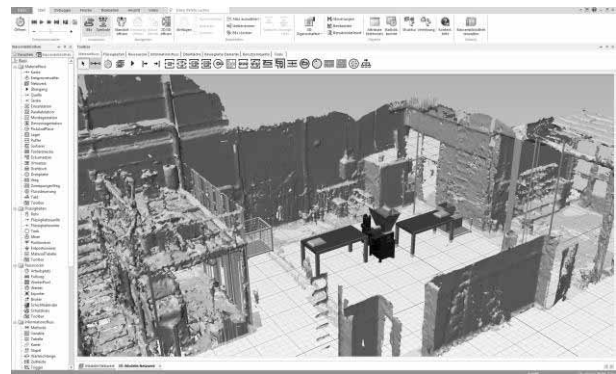


Abbildung 4: Integration der aufgenommenen 3D-Modelle in die Simulationsumgebung

Durch die Verwendung von Originalobjekten aus der umzuplanenden Halle ist es bei Bedarf möglich, in Plant Simulation nicht nur die Materialflusssimulation durchzuführen, sondern auch das Maschinenlayout maßstabsgerecht zu erstellen.

2.3 Umplanung in VR

Nach der Simulation des Modells kann für die Feinplanung des Layouts eine VR-Brille zur Hilfe genommen werden. Der hohe Grad an Immersion dieser Brillen kann zu einer entsprechend besseren Planung führen da auf reale Gegebenheiten der Halle, wie Tore, Türen, Feste Wege aber auch Abflussrinnen usw. Rücksicht genommen werden kann. Plant Simulation unterstützt derartige Brillen nativ momentan noch nicht.

Der am IMAB gegangene Weg zur Überbrückung dieses Defizits ist die bidirektionale Verbindung von Plant Simulation mit einem VR-Brillen fähigen Framework. Abbildung 5 stellt im Diagramm die Verbindung zwischen Plant Simulation und den VR- und AR-Brillenframeworks dar.

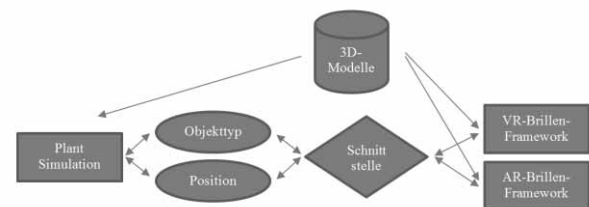


Abbildung 5: Bidirektionale Verbindungen zwischen PlantSimulation und den VR- und AR-Brillenframeworks

Über eine Schnittstelle werden die im Simulationsmodell befindlichen Objekttypen und deren Positionskordinaten übertragen. Auf Basis der aufgenommenen 3D-Modelle ist das Framework dazu in der Lage das Simulationsmodell automatisiert nachzubilden.

Ein Export von 3D-Modellen aus Plant Simulation sowie eine entsprechende Konvertierung und Nachbearbeitung für das VR-Framework entfällt somit vollständig. Abbildung 6 das VR-Layout in besagtem Framework.



Abbildung 6: Bearbeitung des 3D-Layouts in der VR-Umgebung durch den Fabrikplaner

Neben der Eigenschaft zur Visualisierung mittels einer VR-Brille ist das Framework mit zusätzlichen Funktionen ausgestattet. Diese bestehen u.a. aus der Möglichkeit, die detaillierten Maschinenmodelle mit Hilfe der Controller der VR-Brille frei im Raum anzuordnen. Anschließend lassen sich die veränderten Maschinenpositionen an Plant Simulation zurückübertragen oder, bei Bedarf, als FBX-Datei exportieren und in anderen CAD-Programmen weiterverwenden.

2.4 Überprüfung durch AR

Zur Überprüfung der Planung lassen sich die Detailmodelle an ihren neuen Standorten mittels der AR-Brille in die reale Halle projizieren (Abbildung 7).



Abbildung 7: Virtuelle Überprüfung der geplanten Position des Metallschredders mit Hilfe der Microsoft HoloLens

Grundlage ist hier die Verbindung zu einem AR-Framework über die Schnittstelle sowie die automatische Verwendung der aufgenommenen 3D-Modelle. Durch den Einsatz mehrerer AR-Brillen im Verbund ist es möglich, den Planungsstand vor Ort im Team zu besprechen und zu modifizieren. Die Maschinenmodelle lassen sich dabei von allen beteiligten Personen per Fingerzeig verschieben. Die neuen Positionen werden dann automatisch in das Simulationsmodell zurückübertragen. So lassen sich eventuelle Fehler frühzeitig erkennen und korrigieren aber auch Änderungswünsche integrieren.

3 Zusammenfassung

Die vorgestellte Methodik zeigt, am Beispiel der Fabrikplanung, wie Modellbildung und Simulation durch VR- und AR-Brillen unterstützt werden können. Zum einen lassen sich die Eigenschaften zur 3D-Datenaufnahme nutzen um räumliche Strukturen zu erhalten. Zum anderen lassen sich, durch den hohen Grad an Immersion und Interaktion von VR-Brillen, die Ergebnisse der Simulation in Layouts übertragen, die den realen räumlichen Gegebenheiten gerecht werden. Gleiches gilt für die Verwendung von AR-Brillen, wo nun die Erkenntnisse aus der Simulation direkt in die Realität projiziert werden. Somit kann durch die Integration von VR- und AR- Brillen in die Modellbildung und Simulation, schon während der Planungsphase, mit relativ einfachen Mitteln verblüffend realitätsnah und wesentlich fehlerfreier agiert werden.

References

- [1] Bracht, U.; Geckler, D; Wenzel, S.: *Digitale Fabrik – Methoden und Praxisbeispiele, Basis für Industrie 4.0. 2.* aktualisierte und erweiterte Auflage; Springer-Verlag, Berlin (2018).
- [2] Kampker, A.; *Innovative Fabriken interdisziplinär planen.* wt Werkstattstechnik online 102 (2012) 4. Internet: www.werkstattstechnik.de; Düsseldorf: Springer-VDI Verlag, S. 186–192.
- [3] Schönheit, M.; Kuhnert, T.: *Smart Factory – Die Planung der intelligenten Fabrik.* wt Werkstattstechnik online 107 (2017) 4, Internet: www.werkstattstechnik.de; Düsseldorf: Springer-VDI Verlag, S. 219–224.
- [4] Hoffmeyer, A.: *Integration komplexer dynamischer Systeme in Augmented Reality-Anwendungen im Fabriklebenszyklus und in der Fabrikplanung.* docupoint, München (2013).
- [5] Brosch, P.: *Smarte digitale Layoutplanung – Neue virtuelle mobile Ansätze für Umplanungen.* Dissertation. TU Clausthal, IMAB – Innovationen der Fabrikplanung und -organisation (Band 32), Hrsg.: Prof. Dr.-Ing. Uwe Bracht, Shaker Verlag, Aachen (2014).
- [6] Enderle, R.: How virtual reality will impact the enterprise. Cio.com, (2015), Internet: <http://www.cio.com/article/2902749/it-industry/how-virtual-reality-will-impact-the-enterprise.html>, Zugriff am 24.06.2017.
- [7] Bracht, U.; Schlegel, M.: *AR- und VR-Brillen in der Fabrikplanung.* wt Werkstattstechnik online 108 (2018) 4, Internet: www.werkstattstechnik.de; Düsseldorf: Springer-VDI Verlag, S. 251–256.
- [8] Berssenbrügge, J.; Stöcklein, J.; Köchling, D.: *Interactive VR-based Visualization for Material Flow Simulations.* VAMR 2016 (2016), Basel: Springer International Publishing Switzerland, S. 587–596.

Die lernende Lernfabrik – eine intelligente Lehr-Lernumgebung zur Energie- und Ressourceneffizienz (ILehLe)

Mathias Bös^{1*}, Bastian Thiede², Gerrit Posselt², Christoph Herrmann²

¹ SDZ GmbH, Hauert 20, 44227 Dortmund, Germany; **mathias.boes@sdz.de*

² Technische Universität Braunschweig, Institut für Werkzeugmaschinen und Fertigungstechnik (IWF), Nachhaltige Produktion & Life Cycle Engineering, Langer Kamp 19 b, 38106 Braunschweig, Germany, **b.thiede@tu-braunschweig.de*

Abstract. Mit dem bundesgeförderten Forschungsprojekt ILehLe (Die intelligente Lehr-Lernumgebung) werden herkömmliche Lernfabriken innoviert und zu neuartigen, selbstlernenden Lernfabriken weiterentwickelt - sogenannten intelligenten Lehr-Lernumgebungen. Eine wichtige Komponente spielt hierbei das virtuelle Modell der Lernfabrik, welches ein digitales Abbild der gesamten Lernumgebung anstrebt, um neue Handlungsmöglichkeiten zu bieten und gleichzeitig für die Lernenden einen transparenteren Zugang zu den vermittelten Themen schafft. Das virtuelle Modell (Simulationsmodell) ergänzt dazu eine physische Lernfabrik um zusätzliche Aspekte einer Fabrik die aus Kosten- oder Platzgründen nicht vorhanden sind, für das Lernziel jedoch wichtig sind. Das virtuelle Modell ist mit einer physischen Lernfabrik sowie mit dem adaptiven Learning Management System (LMS) als weiterer Komponente von ILehLe gekoppelt. Das virtuelle Modell übernimmt Systemzustände der physischen Lernfabrik und kann mit dieser in Wechselwirkung treten. Für die Umsetzung dieser Aspekte werden im Vorhaben bereits existierende Standards der Simulation um spezifische Anforderungen von ILehLe erweitert. Dazu werden Bausteine für das virtuelle Abbild der Lernfabrik entwickelt sowie Bausteine für die Integration des virtuellen Modells mit den übrigen Komponenten von ILehLe.

1 Motivation

Um Prozesse des lebenslangen Lernens als permanenter Bestandteil der allgemeinen Arbeitsaktivitäten in Unternehmen zu verankern, wird eine ganzheitliche Lernkultur in modernen Lehr-Lernumgebungen wie Lernfabriken benötigt, die sich durch die Entwicklung zentraler Kompetenzen und der Fähigkeit zur kritischen Reflexion auszeichnet [1, 2]. Insbesondere vor dem Hintergrund von schnellen und anhaltenden Innovationen im Rahmen von Industrie 4.0 müssen Unternehmen kontinuierlich das Wissen und die Kompetenzen ihrer Angestellten in ihrem neuen digitalisierten Arbeitsplatz fördern [3, 4]. Lernfabriken stellen ein realitätsnahes

virtuelles oder physisches Modell einer Fabrik dar, kombinieren dieses mit ausgewählten Methoden der Didaktik zu einer handlungsorientierten Lehr-Lernumgebung [5] und ermöglichen so informelles Lernen durch Probehandeln. Sie sind neben der Wissensvermittlung insbesondere zur Aneignung von beruflichen Handlungskompetenzen geeignet [6]. Demgegenüber steht jedoch der erforderliche Aufwand bei der Betreuung, Kontrolle und Steuerung des Lernprozesses.

Um den Lernenden eine eigenständige Aneignung von Wissen und Kompetenzen in einer Lernfabrik zu ermöglichen – unabhängig von einzelnen Schulungen oder Workshops – sind ein intelligentes Lernmanagementsystem (LMS) sowie umfangreiche Feedbackinformationen für den Lernenden nötig. Hierfür muss die Lernfabrik mit entsprechenden Sensoren sowie Aktoren ausgestattet sein und sich mittels eines virtuellen Modells auswerten und steuern lassen, um (teil-) automatisiert gesteuerte Lernprozesse zu ermöglichen. Bestehende Konzepte und Umsetzungen von Lernfabriken weisen hier Defizite auf und erreichen die Anforderungen nur unzureichend. Lernende sind damit für die Nutzung einer Lernfabrik auf einen entsprechenden Dozenten oder Trainer angewiesen und an spezielle Zeitfenster gebunden. Als Weiterbildungsinfrastruktur, die sich für ein lebenslanges Lernen flexibel in den jeweiligen Arbeitsalltag integriert, können Lernfabriken damit bisher nicht dienen.

Lernfabriken schließen zudem die Lücke zwischen Aus- und Weiterbildung und betrieblicher Praxis momentan nur unzureichend. Die Problemstellungen in der ingenieurwissenschaftlichen Ausbildung weichen nach wie vor signifikant von Problemstellungen aus der Praxis ab. Die Praxis ist oftmals geprägt durch Unübersichtlichkeit, Komplexität und Zielkonflikte. Entscheider sind oftmals konfrontiert mit einer Vielfalt an po-

enziellen Lösungswegen vor dem Hintergrund einer unzureichenden Informationslage und nicht-technischen Einflussfaktoren [7]. Um diesen Umstand zu entsprechen, muss der enge inhaltliche Fokus zugunsten eines ganzheitlicheren und damit zugleich realistischeren Ansatzes erweitert werden. Dazu zählt auch die Integration weiterer Kennzahlen aus dem unternehmerischen Kontext. Die Mehrheit der bestehenden Lernfabriken konzentriert sich auf die klassischen Zielgrößen der Produktion wie Zeit, Kosten und Qualität. Im Zuge einer nachhaltigen Entwicklung sind jedoch weitere Zielgrößen relevant, insbesondere hinsichtlich der Umweltwirkungen von Produktionsprozessen. Erste Konzepte zur Verankerung von Energie- und Ressourceneffizienz bestehen bereits, bleiben zumeist jedoch auf Einzelprozesse beschränkt oder erfordern komplexe und sehr teure Infrastrukturen.

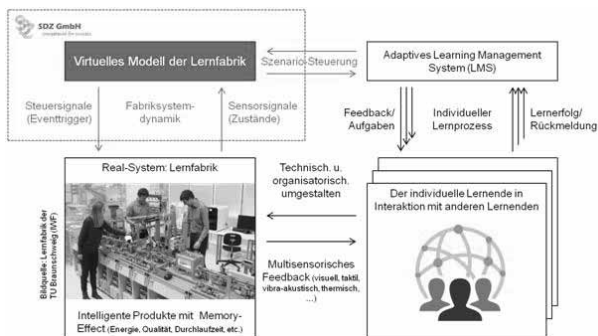


Abbildung 1: Die intelligente Lehr-Lernfabrik ILehLe (in Anlehnung an [8])

Ziel von ILehLe ist es daher, das Lernen im Rahmen der Lernfabrik effizient und kognitiv vielseitig auszurichten und mit einem didaktischen Konzept sinnvoll zu komplementieren [8]. Die intelligente Lehr-Lernumgebung verfügt dazu über die nachfolgend aufgeführten Komponenten (siehe Abbildung 1):

- Die reale (physische) Lernfabrik,
- ein virtuelles Modell als Abbild und Erweiterung der Lernfabrik (Simulation),
- ein Lern-Management-System (LMS) sowie
- die Lernenden als Teil des Gesamtsystems.

Aufbauend auf einer am IWF bestehenden Lernfabrik [<http://www.dielernfabrik.de/experimentierfabrik>; 9, 10] wird im Rahmen des Verbundprojektes ILehLe eine intelligente Lehr-Lernumgebung am Beispiel des Themenfeldes Energie- und Ressourceneffizienz konzipiert, die durch den Lernenden eigenständig genutzt werden kann und diesen auf der Basis von Feedbackmethoden zur adaptiven Lernprozesssteuerung interaktiv in seinem

individuellen Lernprozess unterstützt. Die Wirkungsweise dieser modifizierten Lernfabrik wird zudem wissenschaftlich evaluiert.

2 Das virtuelle Modell der Lernfabrik

2.1 Zielstellung

Ziel des Teilvorhabens *Virtuelles Modell der Lernfabrik* ist die Ergänzung des Real-Systems (Modell einer physischen Fabrik als Lernfabrik) um eine Systemumgebung, die fehlende Aspekte bzw. Komponenten des Real-Systems virtuell bereitstellt. Dadurch wird die virtuelle Einbindung der Lernfabrik in übergeordnete Strukturen der Fabrikumgebung erreicht, ergänzende Systemzustände können visualisiert werden oder eine Erweiterung des Untersuchungsraums kann den Lernenden zur Verfügung gestellt werden.

Diese Idee der virtuellen Erweiterung adressiert die bereits im Projekt identifizierten Defizite physischer Lernfabriken. Hierbei sind zeitliche, technische und finanzielle Limitationen vordergründig [11]. Zeitliche Limitationen können einerseits organisatorisch begründet sein, beispielsweise durch lange Rüstzeiten oder Umbauten an Maschinen und Anlagen ohne didaktischen Mehrwert, oder durch die erforderliche Zeitspanne, bis sich gewünschte Effekte einstellen. Technische und finanzielle Limitationen können durch die Betrachtung des wirtschaftlichen Betriebs der Lernfabrik entstehen, bei der gegebenenfalls technisch anspruchsvolle Umsetzungsmaßnahmen zulasten von Lehrinhalten vermieden werden.

Das virtuelle Modell der Lernfabrik besitzt eine bidirektionale Hardwareschnittstelle zur physischen Lernfabrik. Diese ermöglicht es, sowohl den Status der physikalischen Systemkomponenten mittels vorhandener Sensortechnik zu bestimmen als auch Steuerungssignale an Komponenten zu übertragen. Szenario-basiert können so Kennwerte und Zielerreichungsgrade berechnet und daraufhin Ereignisse ausgelöst werden. Auf diese Weise können System-Effekte realisiert werden, die für eine Gamifizierung der Lernumgebung notwendig sind. Das Simulationsmodell ermöglicht es ebenfalls, eine Skalierung von kleinen technischen und organisatorischen Umgestaltungseffekten in der physikalischen Lernfabrik durch den Lernenden in gleichem Maße wie in realen Fabrikssystemen vorzunehmen. Durch die Realisierung spezifische Schnittstellen von der Lernfabrik

zur Simulation kann eine echtzeitnahe Kopplung von virtueller und realer Welt erzeugt werden.

Die Lernfabrik richtet sich insbesondere auf Fragen der Energie- und Ressourceneffizienz in der Produktion. In modernen Produktionssystemen sind entscheidende Prozess- und Produkteigenschaften hinsichtlich Energie- und Ressourceneffizienz für den Menschen nur indirekt erfahrbar. Anlagen und Herstellungsprozesse sind von außen nicht zugänglich und geben keine direkten Informationen preis. Um diese Aspekte in die Lernfabrik zu integrieren, wird ein erweitertes virtuelles Modell erstellt. Über diese Komponente, die über eine direkte Kopplung zum physischen Modell der Lernfabrik verfügt, wird für die Lernenden der Zugang zu den vorab aufgeführten Aspekten geschaffen. So können über die Kopplung von physischem und virtuellem Modell Produkteigenschaften (z. B. Material, Herstellungskosten, Einsatzbereiche) sowie Produkt- und Systemzustände (z. B. Energieverbrauch, Anlagenzustände, Produktivität) dargestellt und visualisiert werden. Über das Virtuelle Modell können Vorgaben des LMS zur Steuerung des physischen Modells empfangen, verarbeitet (aufbereitet) und für die Ausführung in der physischen Fabrik an diese weitergeleitet werden. Dem virtuellen Modell kommt zudem die Aufgabe zu, die Abläufe im physischen Modell, insbesondere für die Energie- und Ressourceneffizienz der Produktion, zu steuern, zu protokollieren, zu dokumentieren und in Form von Kennzahlen zu visualisieren. Im Gegensatz zu den bisherigen Ansätzen physischer Lernfabriken können durch das virtuelle Modell neue Zustände der Fabrik geschaffen und individuell an die Lernenden angepasst werden und so einen effektiveren Lernprozess ermöglichen. Der Lehrende wird zudem an dieser Stelle von operationalen Tätigkeiten entlastet und kann die Lernenden besser unterstützen.

Die Entwicklungen im Teilvorhaben *Virtuelles Modell der Lernfabrik* umfassen daher im Wesentlichen zwei Aspekte. Zum einen handelt es sich um die Entwicklung von Bausteinen für die eigentliche virtuelle Abbildung der Lernfabrik. Die zu entwickelnden Bausteine gewährleisten, dass ein virtuelles Abbild der Lernfabrik realisiert wird. Dies umfasst auch die individuelle Konfiguration des virtuellen Modells über die Parametrierung der Bausteine. Der zweite Aspekt umfasst die Integration des virtuellen Modells in die Systemumgebung der intelligenten lernenden Lernfabrik. Dadurch wird die Kommunikation des virtuellen Mo-

dells mit dem physischen Modell der Lernfabrik ermöglicht, sodass Systemzustände (Sensorsignale) an das virtuelle Modell übertragen werden können und Steuerungssignale vom virtuellen Modell an das physische Modell übermittelt werden können. Über das Integrationskonzept wird zudem die Kopplung mit adaptiven LMS gewährleistet. Das LMS stellt im Gesamtsystemkonzept die Steuerungsinstanz der lernenden Lernfabrik dar. Darüberhinausgehend werden Bausteine entwickelt, die auch eine Analyse der Abläufe in der Lernfabrik ermöglichen. Dazu werden Zustände über definierbare Zeiträume ausgewertet und in Form von aussagefähigen und interpretierbaren Kennzahlen dem Lernenden zur Verfügung gestellt.

2.2 Das Integrationskonzept von physischer und virtueller Lernfabrik

Für die Kommunikation aller Komponenten der intelligenten Lehr-Lernumgebung lLehLe wird unter Berücksichtigung der Anforderung aller Partner eine gemeinsame Kommunikationszentrale entwickelt, das sogenannte Gateway.

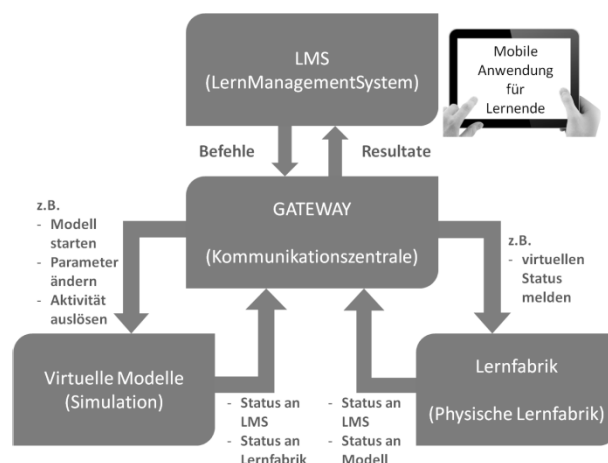


Abbildung 2: Gateway lLehLe

Das Gateway übernimmt die Aufgabe, die Kommunikation zwischen allen Modulen zu steuern (vgl. Abbildung 2). Es übernimmt Befehle oder Signale von einer Komponente und übermittelt diese an den adressierten Empfänger. Das virtuelle Modell empfängt so über das Gateway Informationen zur Änderung von Parametern oder Befehle, die Aktivitäten im virtuellen Modell auslösen. Auf der anderen Seite sendet das virtuelle Modell Statusmeldungen an das Gateway, welches diese an die Empfänger übermittelt (die physische Lernfabrik bzw. das LMS).

Herausfordernd hat sich im Projekt die Interprozess-Kommunikation in einem heterogenen Netz mit verschiedenen Beteiligten, die auch räumlich getrennt sein können, gezeigt. Aus diesem Grund werden mehrere unterstützende Protokolle und Kommunikationstechniken eingesetzt. Das Gateway muss zwischen den unterstützten bzw. erwarteten Protokollen der angeschlossenen Komponenten vermitteln. Das Gateway der intelligenten Lehr-Lernumgebung ILehLe nutzt folgende Protokolle bzw. Techniken:

- HTTP (Hypertext Transfer Protocol): Standardprotokoll im Internet
- OPC/UA (OPC Unified Architecture): standardisiertes Automatisierungsprotokoll für Kommunikation zwischen Anlagenkomponenten
- WebSocket: Bidirektionale Kommunikation auf einer TCP-Verbindung (hier eingesetzt zur online-Übertragung von Daten zwischen Gateway und Modulen).
- COM (Component Object Model): Microsoft Windows Technologie zur Kommunikation

Das Gateway wird als Hintergrunddienst auf einem Server-Rechner betrieben. Auf diesem laufen sowohl das Gateway als auch das virtuelle Modell. Für Letzteres werden folgende Services entwickelt und bereitgestellt:

- Entgegennahme von Befehlen vom Gateway und Verarbeitung in der Simulation,
- Lieferung von Inhalten an Gateway (Inhalte in der Simulation erzeugen und versenden) sowie
- Verwaltung der Simulatorinstanzen

DOSIMIS-3 (siehe Abschnitt 2.3) ist dazu um Funktionen erweitert, um Befehle vom Gateway zu übernehmen und diese an das Simulationsmodell weiterzugeben. Entsprechende Erweiterungen sind notwendig für die Generierung und Weitergabe von Statusmeldungen von DOSIMIS-3 an das Gateway. Hierzu werden die zuvor aufgeführten Protokolle und Techniken im Simulationssystem erweitert und für die spezifischen Anforderungen von ILehLe angepasst. Innerhalb der Bausteine von DOSIMIS-3 sind Funktionalitäten implementiert, um die übernommenen Befehle interpretieren zu können und resultierende Aktionen abzuleiten. Zudem sind spezifische Statistiken implementiert, um die benötigten Informationen im virtuellen Modell zu generieren und an die konsumierenden Module weiterleiten zu können.

Da es sich beim Real-System um eine echte Maschi-

ne mit entsprechend realen Gefahren für die Lernenden handelt, sind alle genannten Steuersignale keine direkten Befehle an die Aktorik der Maschine, sondern lediglich Anweisungen zur Ausführung. Die jeweiligen Steuerungen der Prozessschritte evaluieren kontinuierlich, ob die Anweisung im aktuellen Zustand tatsächlich sicher ist und zur Ausführung gebracht werden kann.

2.3 Das Simulationssystem DOSIMIS-3

Als Basis für die Entwicklung des virtuellen Modells dient das Simulationssystem DOSIMIS-3 der Firma SDZ GmbH. DOSIMIS-3 ist ein ereignisdiskreter Simulator (DES – discrete event simulation) und zählt zu den etablierten Simulationstools für Anwendungen in Produktion und Logistik. Seit über 30 Jahren wird DOSIMIS-3 erfolgreich in Industrie und Handel eingesetzt.

Die Ursprünge von DOSIMIS-3 reichen bis Anfang der 80er Jahre zurück, als das damals revolutionäre Konzept von DOSIMIS-3 von einer visionären Gruppe um Professor Bernd Noche entworfen und umgesetzt wurde. Diese Pioniere der Simulation in Produktion und Logistik hatten sich zum Ziel gesetzt, ein interaktiv zu bedienendes Simulationssystem zu entwickeln, dass von Planern und Ingenieuren angewandt werden konnte. Dazu wurde nicht nur erstmalig eine grafische Oberfläche für ein Simulationssystem entwickelt. Die eigentliche Innovation bestand in einer Bausteinbibliothek mit vorgegebenen Modulen, die eine Abbildung sämtlicher Komponenten logistischer oder von Produktionssystemen ermöglichten und die über Parameter für die jeweils konkret vorgesehene Komponente spezifiziert werden konnten. Mittlerweile orientieren sich zahlreiche Hersteller von Simulationssystemen an diesem Konzept von DOSIMIS-3.

Mit DOSIMIS-3 kommen technisch orientierte Anwender schnell und einfach zu sicheren Ergebnissen, weshalb DOSIMIS-3 auch als Taschenrechner des Planers bezeichnet wird. Schnörkellos und ergonomisch in der Handhabung garantiert DOSIMIS-3 einen schnellen und sicheren Einsatz der ereignisdiskreten Simulation und somit höchste Effizienz bei der Durchführung von Simulationsstudien. Der Anwender kann mit DOSIMIS-3 innerhalb von Minuten die Produktion eines Jahres, die gesamte Logistik einer Fertigung oder die Lagerprozesse vom Wareneingang über die Kommissionierung, Nachschubanforderung bis hin zum Warenausgang simulieren.

DOSIMIS-3 verfügt bis heute über die umfangreichste und funktional stärkste Bausteinbibliothek der am Markt etablierten Simulationssysteme, sodass DOSIMIS-3 einen vergleichsweise schnellen und validen Modellaufbau ermöglicht. Dabei wird auch die Qualitätssicherung der Simulationsmodelle im Rahmen der Validierung methodisch von DOSIMIS-3 unterstützt und abgesichert. Die hohe Performance bei der Durchführung von Simulationsläufen ermöglicht die Analyse zahlreicher Varianten und Szenarien in kürzester Zeit, für DOSIMIS-3 über umfangreiche Auswertungsmethoden verfügt. Ergänzt wird dies durch die virtuelle Animation der Simulationsmodelle.

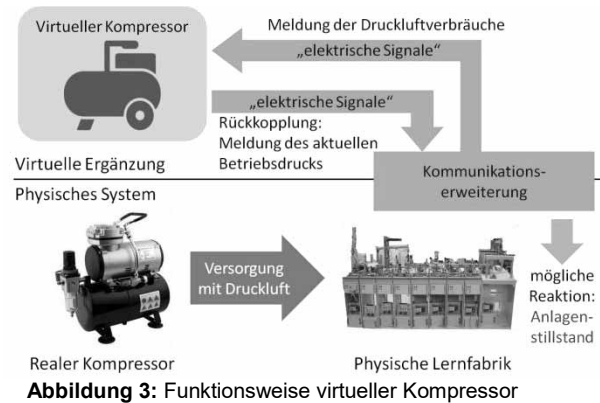
Die praxisnahe und anwendungsorientierte Entwicklung von DOSIMIS-3 war der Grund für die Auswahl dieses Simulationssystems für das Vorhaben. Neben der Nutzung bereits zahlreicher Bausteine für die virtuelle Erweiterung der physischen Lernfabrik konnten sowohl weitere Bausteine als auch die notwendigen Kommunikationserweiterungen recht einfach umgesetzt werden.

2.4 Umsetzung am Beispiel eines virtuellen Kompressors

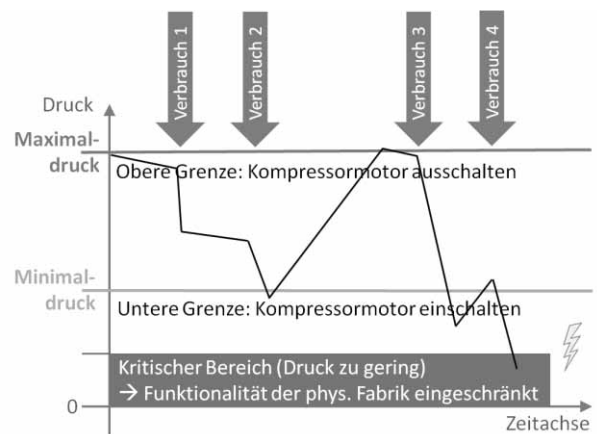
Das virtuelle Modell der Lernfabrik umfasst unterschiedliche Komponenten, mit denen die physische Lernfabrik virtuell erweitert wird. Im ersten Schritt ist dazu ein virtueller Druckluft-Kompressor vorgesehen.

Der Kompressor versorgt die physische Lernfabrik mit Druckluft, die diese für den Betrieb benötigt. In der Realität erzeugt ein Kolbenkompressor mit einem Druckluftbehälter den benötigten Überdruck gegenüber der Umgebung. Während des Betriebs wird dieser Druckunterschied genutzt, um beispielsweise Handhabungsaufgaben durchzuführen. Diese Entnahme von Druckluft führt zu einer Verringerung des vorherrschenden Drucks. Der Kompressor ist mittels einer Hysterese geregelt: Bei Unterschreiten eines definierten Minimaldrucks beginnt der Kompressor mit der Kompression der Luft und füllt einen Druckluftbehälter erneut bis zum definierten Maximaldruck.

Der virtuelle Kompressor ist über elektrische Signale mit der physischen Lernfabrik verbunden. Er bildet die Charakteristik eines realen Kompressors nach. Über Parametrierung kann die Leistung des Kompressors variiert werden und das resultierende Betriebsverhalten der physischen Lernfabrik ermittelt werden.



Die physische Lernfabrik meldet über Signale kontinuierlich den Druckluftverbrauch in Form eines Zählerstandes an die Steuerung des virtuellen Kompressors. Dieser meldet aufgrund der Charakteristik des Kompressors den aktuellen (virtuellen) Betriebsdruck der Druckluft an die physische Lernfabrik (siehe Abbildung 3). Wie der reale Kompressor baut der virtuelle Kompressor den Druck entsprechend den physikalischen Gesetzmäßigkeiten wieder auf, wenn dieser Minimaldruck unterschreitet. Sollte der Betriebsdruck zu niedrig sein, beispielsweise aufgrund eines zu leistungsarm konfigurierten Kompressors, kann dies zum Stillstand der physischen Lernfabrik führen, wie in Abbildung 4 dargestellt. Die Reaktion auf den zu geringen Druck ist entsprechend in einem Programmbaustein in den Steuerungen der Prozessschritte hinterlegt, welche den virtuellen Druck von der Simulation gemeldet bekommen.



Dem Lernenden werden spezifische Kennzahlen (z. B. über den Druckluftverbrauch) zur Verfügung gestellt, so dass die Kompressor-Konfiguration analy-

siert und eine geeignete Konfiguration ermittelt werden kann.

Für unterschiedliche Aufgabenstellungen und Auswertungen können beliebige Konfigurationen für den virtuellen Kompressor vorgegeben werden, um die Wirksamkeit unterschiedlicher Kompressortypen analysieren zu können. So können unter anderem folgende technische Parameter geändert werden:

- Volumen des Druckbehälters,
- Hubraum des Kompressorkolbens
- Umdrehungsgeschwindigkeit des Motors
- Energieverbrauchbedarfe entsprechend des Datenblatts realer Kompressoren

Im virtuellen Modell können Änderungen der Konfiguration über die Variation von Parametern vorgegeben werden. Dies hat den Vorteil, dass Änderungen im laufenden Betrieb, d. h. ohne manuellen oder physischen Eingriff, zur Laufzeit des Systems vorgenommen werden können. Eine Änderung des Kompressors im virtuellen Modell erfordert daher nicht das notwendige Know-how zum eigentlichen Umbau in der physischen Welt. Die Änderungen erfolgen ohne Zeitverzug, verursachen keine Anschaffungskosten für unterschiedliche Kompressoren und auch keine Umbaukosten. In der physischen Lernfabrik verhalten sich die Elemente entsprechend der virtuellen Manipulation, z. B. durch Stillstand bei zu geringem Druck. Für die Lernenden bietet sich hierdurch die Möglichkeit, in sehr kurzen Intervallen die geplanten Änderungen vorzunehmen und den Einfluss auf die reale Lernfabrik ohne zeitliche Verzögerung wahrzunehmen.

3 Anwendungsfälle des virtuellen Modells

Im Vorhaben ILeHLe wird in jeweils einem akademischen und einem industriellen Anwendungsfall unterschieden. Zu dem akademischen Anwendungsfall zählt die virtuelle Erweiterung der physischen Lernfabrik des IWF, zunächst um die Kompressor-Komponente. Weitere geplante Elemente des virtuellen Modells sind virtuelle Erweiterungen der physischen Lernfabrik. Dies beinhaltet sowohl vor- als auch nachgelagerte Prozessschritte der physischen Prozesskette. Aus technischer Sicht stellt dies keine weitere Neuentwicklung dar. Es können die gleichen Kommunikationsprotokolle verwendet werden, um zu Beginn der Prozesskette der physischen Lernfabrik die virtuelle Zuführung von

Teilen zu melden bzw. an deren Ende die Übergabe von physischen Teilen an das virtuelle Modell. So können zum einen komplexe vorgelagerte Prozesse (z. B. der Teilevorfertigung) im virtuellen Modell abgebildet werden, zum anderen aber auch Auswirkungen von nachgelagerten Prozessen (z. B. störungsbedingte Blockaden im nachgelagerten virtuellen Modell) auf das Betriebsverhalten der physischen Lernfabrik ermittelt werden. Aus didaktischer Sicht handelt es sich hierbei um eine optionale Erweiterung, um dem Lernszenario eine größere Tiefe zu verleihen. Zum einen kann hierdurch die Relevanz der durchgeführten Änderungen im Gesamtkontext besser dargestellt werden. So können Maßnahmen zur Steigerung der Energieeffizienz die Eigenschaft besitzen, zu einer Problemverschiebung zu führen. Diese können nur in einer ganzheitlichen Betrachtungsweise erkannt werden. Zum anderen bietet sich die Möglichkeit, durch eine steigende Komplexität den Schwierigkeitsgrad für die Lernenden angepasst an den individuellen Fortschritt erhöhen.

Ein weiterer Aspekt ist die sogenannte Langzeitsimulation. Anhand von statistischen Größen wird ein aktuelles Abbild der physischen und virtuellen Lernfabrik in ein reines Simulationsmodell überführt. Hierzu wird zunächst ein separater Referenzlauf gestartet, in dem die physische Lernfabrik einen vorgegebenen Produktionsplan durchläuft. Während des Referenzlaufs treten keine gegenseitige Beeinflussung der Produkte oder Prozesse untereinander statt. Die individuellen Steuerungen der Lernfabrik können mithilfe eigener Messtechnik die relevanten Kenngrößen wie spezifischer Energieeintrag oder Prozesszeit je Variante ermitteln und notwendige Statistiken ermitteln. Im Anschluss an den Referenzlauf werden diese Informationen an das Simulationsmodell übermittelt, welches in der Folge eine Langzeitsimulation ohne Kopplung der physischen Lernfabrik durchführt. Die Aufgabe der Lernenden ist es, anhand der Ergebnisse der Simulation ein sogenanntes Lastmanagement durchzuführen. Hierfür kann das Simulationsmodell über das LMS ein Gesamtlastprofil der Lernfabrik, bestehend aus physischen und virtuellen Komponenten bereitstellen. Die Lernenden können anhand dieser Informationen und dem zuvor vermittelten theoretischen Wissen geeignete Maßnahmen ableiten und umsetzen.

Im industriellen Anwendungsfall werden spezifische Abläufe aus der Praxis des Industriepartners Volkswagen AG in der intelligenten Lehr-Lernumgebung nach-

gebildet, um damit Mitarbeitende des Industriepartners aus- bzw. weiterzubilden zu können. Neben anderen innovativen Methoden von ILeHLe wird auch dieser Anwendungsfall durch eine Zeitraffersimulation ergänzt. Hierbei werden den Mitarbeitenden die Auswirkungen ihrer Entscheidungen auf das resultierende Betriebsverhalten aufgezeigt. So können in kurzer Zeit unterschiedliche Konfigurationen des nachgebildeten Systems vermittelt und analysiert werden. Langzeiteffekte (z.B. über einen Tag oder sogar ein ganzes Betriebsjahr) die in Lernfabriken üblicherweise nur schwer vermittelbar sind, können durch die Kopplung von virtuellem und physischem Modell vermittelt werden.

4 Zusammenfassung

Die vorgestellte virtuelle Erweiterung der Lernfabrik kann in akademischen wie auch im industriellen Anwendungsfall einen wertvollen Beitrag zur Vermittlung von neuem Wissen und Kompetenzen ermöglichen, indem physische, zeitliche und finanzielle Limitationen physischer Lernfabriken überwunden werden. Die Kopplung der physischen Lernfabrik mit der virtuellen Erweiterung ermöglicht es, Probehandlungen durchzuführen und zu bewerten, obwohl deren Effekt erst lange Zeit nach der eigentlichen Änderung ersichtlich wird. Hierdurch kann die zeitliche Effizienz und die Perspektive des Lernenden gesteigert werden. Zudem ergibt sich, beispielsweise durch das virtuelle Kompressormodell, die Möglichkeit, nicht vorhandene Geräte und Anlagen zu betrachten und diese in das Lernszenario einzubeziehen. Es ergibt sich wiederum ein zeitlicher Vorteil durch die fehlende Neukonfiguration und einen finanziellen Vorteil durch geringere Anschaffungs-, Betriebs-, Rüst- und Wartungskosten. Durch die Entlastung der Lehrenden von diesen Nebentätigkeiten, können sich diese auf die intensivere Betreuung von Problemen im Lernprozess der Lernenden konzentrieren die vom LMS nicht abgedeckt werden konnten. Neben der Anwendung im Rahmen des Projekts ILeHLe konnte die virtuelle Erweiterung zudem zeigen, dass eine Kopplung des Realsystems und der Simulation in Echtzeit möglich ist und die erwarteten gegenseitigen Effekte und Einflussnahmen zeigt.

Projektförderung

Das Projekt *ILeHLe – Die lernende Lernfabrik – eine intelligente Lehr-Lernumgebung zur Energie- und Res-*

*sourceneffizienz wird im Förderschwerpunkt *Erfahrbares Lernen* unter den Förderkennzeichen 16SV7556 und 16SV7555 gefördert vom Bundesministerium für Bildung und Forschung (BMBF).*

Literaturhinweise

- [1] Mezirow, J.: “Transformative Learning: Theory to Practice”, In: *New Directions for Adult and Continuing Education*, Issue 74, June 1997, S. 5-12
- [2] Mikula, R. (2007). *Lebenslanges Lernen - emanzipatorische Potenziale veränderter Lernkulturen*. *Magazin Erwachsenenbildung.at*, 2007(2), 1–11.
- [3] Tisch, M., Hertle, C., Cachay, J., Abele, E., Metternich, J. and Tenberg, R., 2013. A systematic approach on developing action-oriented, competency-based Learning Factories. *Procedia CIRP*, 7, 580–585. doi:10.1016/j.procir.2013.06.036
- [4] Chrysolouris, G., Mavrikios, D. and Rentzos, L., 2016. The Teaching Factory: The Manufacturing Education Paradigm. *Procedia CIRP*, 57, 44–48. doi:10.1016/j.procir.2016.11.009
- [5] Wagner, U., AlGeddawy, T., ElMaraghy, H., & Mÿller, E. (2012). The State-of-the-Art and Prospects of Learning Factories. *Procedia CIRP*, 3, 109–114. doi:10.1016/j.procir.2012.07.020.
- [6] Abele, M., Czajkowski, S., Faatz, L., Metternich, J., & Tenberg, R. (2013). Kompetenzorientiertes Curriculum für Lernfabriken. Ein didaktisch hinterlegtes Konzept für Lernfabriken. *Werkstattstechnik online*, 103(3), 240–245
- [7] Jonassen, D. / Strobel, J. / Beng, L. C.: „Everyday Problem Solving in Engineering: Lessons for Engineering Educators“. In: *Journal of Engineering Education*, Vol. 95 (2), April 2006, S. 139–151
- [8] Posselt, G., Böhme, S., Aymans, S., Herrmann, C., Kaufeld, S. (2016). Intelligent Learning Management by Means of Multi-sensory Feedback. *Procedia CIRP*, 54, 77–82. doi: 10.1016/j.procir.2016.05.061.
- [9] Kaluza, A., Juraschek, M., Neef, B., Pittschellis, R., Posselt, G., Thiede, S., Herrmann, C. (2015). Designing Learning Environments for Energy Efficiency through Model Scale Production Processes. *Procedia CIRP*, 32, 41–46. doi: 10.1016/j.procir.2015.02.114.
- [10] Thiede, S., Juraschek, M., Herrmann, C. (2016). Implementing Cyber-physical Production Systems in Learning Factories. *Procedia CIRP*, 54, 7–12. doi:10.1016/j.procir.2016.04.098.
- [11] Thiede, B., Posselt, G., Kaufeld, S., Herrmann, C. (2017). Enhancing Learning Experience in Physical Action-orientated Learning Factories Using a Virtually Extended Environment and Serious Gaming Approaches. *Procedia Manufacturing*, 9, 238–244. doi:10.1016/j.promfg.2017.04.042.

Zuordnungsmethoden bei Dual-load Fahrzeugen im Horizontalverkehr von Container Terminals

Anne Kathrina Schwientek^{1*}, Fredrik Branding¹, Ann-Kathrin Lange¹, Carlos Jahn¹

¹Institut für Maritime Logistik, Technische Universität Hamburg, Am Schwarzenberg-Campus 4, 21073 Hamburg;
*a.schwientek@tuhh.de

Abstract. Container Terminals dienen als logistischer Knoten für den Umschlag zwischen Seetransport und Weitertransport im Hinterland. Steigende Transportmengen und immer größere Schiffe führen dazu, dass die Umschlagprozesse möglichst schnell und effizient durchgeführt werden müssen. Der Horizontaltransport zwischen Wasserseite und Containeryard spielt dabei eine wesentliche Rolle. Eine effiziente und robuste Organisation der Containertransporte durch eine entsprechende Auftragszuordnung von Containern und Fahrzeugen zwischen Containerbrücken und Yard ist dabei essentiell, um die Containerbrücken optimal betreiben zu können. Die meisten Fahrzeuge des Horizontaltransport sind technisch in der Lage zwei 20'-Standardcontainer zu transportieren. Dies wird in bestehenden Untersuchungen zur Auftragszuordnung im Horizontaltransport kaum berücksichtigt. Daher zielt die vorliegende Simulationsstudie darauf ab, den Einfluss der Fahrzeugkapazität im Horizontaltransport auf die Leistungsfähigkeit von verschiedenen Zuordnungsmethoden zu untersuchen. Dabei werden wesentliche Parameter eines Container Terminals berücksichtigt.

Einleitung

Die aktuelle Schiffsgrößenentwicklung der Containerschiffe wirkt sich im besonderen Maße auf den Umschlagsprozess im Hafen aus. Hier sollen zum einen die Schiffsfliegezeiten für die Reedereien reduziert werden und zum anderen Umschlagsressourcen in begrenzten Zeitfenstern sinnvoll eingesetzt werden (vgl. [1]). Durch die steigende Anzahl an Containern wird es umso wichtiger, die Prozesse auf Container Terminals möglichst effizient zu gestalten.

1 Problemanalyse

Container Terminals können grundsätzlich in die vier Bereiche Wasserseite, Horizontaltransport, Containeryard und Landseite aufgeteilt werden (vgl. [2,3]). Beim Horizontaltransport werden Container zwischen der Kaikante und dem Containeryard umgeschlagen. Vorgelagerte Prozessschritte beziehen sich auf die Entladung durch die

Containerbrücken und nachgelagerte auf den Lagerbereich mit der anschließenden Beladung für Lkw- bzw. Bahn-Verkehre. Den Fahrzeugen zwischen Kaikante und Containeryard werden Aufträge für den Transport zugewiesen. Diese Auftragszuordnung kann durch feste Zuweisung von Strecken, aber auch durch unterschiedliche Zuordnungsmethoden erfolgen.

Allgemein können Fahrzeuge in passives und aktives Equipment unterschieden werden (vgl. [4]). Aktives Equipment wie z. B. Straddle Carrier kann unabhängig von weiteren Umschlagmitteln einen Container hochheben. Passives Equipment wie z. B. Traktor-Trailer-Einheiten (TT) oder Automated Guided Vehicles (AGV) ist beim Umschlag auf die Containerbrücke bzw. die Lagerkrane angewiesen. Wenn eine Containerbrücke auf ein passives Fahrzeug warten muss, ergeben sich Wartezeiten, die den Durchsatz des Terminals senken. Die Transportaufträge müssen dementsprechend so den Fahrzeugen zugeordnet werden, dass keine Wartezeiten an den Engpässen entstehen. Dabei liegt ein besonderer Fokus darauf, einen kontinuierlichen Betrieb der Containerbrücken zu gewährleisten, da diese zum einen das kostenintensivste Equipment auf einem Container Terminal sind und zum anderen bei dem Umschlag an der Kaikante die Haupteinnahmequelle eines Terminals liegt. Gleichzeitig sollen auch zur Optimierung der internen Prozesse die Fahrtwege der Fahrzeuge minimiert und die Anzahl der notwendigen Umstapelvorgänge in den Containerlagern möglichst geringgehalten werden.

Weiterhin erhöht sich die Komplexität bei der Auftragszuordnung dadurch, dass die meisten Transportfahrzeuge eine Kapazität von zwei 20'-Standardcontainern (engl. Twenty-Foot Equivalent Units (TEU)) besitzen. Somit ist es ihnen möglich, zwei 20'- bzw. einen 40'-Container zu laden. Dadurch ergeben sich bei unterschiedlichen Zieldestinationen der Container eine Vielzahl an Varianten für den Transport. Diese wird besonders erhöht, wenn berücksichtigt wird, dass zum Zeitpunkt der ersten Entladung ein dritter Container für die

Beladung zur Verfügung stehen könnte, bevor der zweite Container entladen wird. Möglich wird dies erst durch die Twin-lift-Fähigkeit der Containerbrücken, d.h. die Eigenschaft, dass eine Containerbrücke einen oder zwei hintereinander positionierte 20'- bzw. einen 40'-Container umschlagen kann.

Ziel dieser Arbeit ist es, mittels einer Simulationsstudie zu ermitteln, welchen Einfluss die Fahrzeugkapazität auf die Leistungsfähigkeit der Zuordnungsmethoden der Aufträge im Horizontalverkehr hat. Im Folgenden wird zunächst der Stand der Forschung umrissen und der Aufbau des Simulationsmodells und der Experimentplan beschrieben. Abschließend erfolgt eine kurze Darstellung der zu Ergebnisse sowie das Fazit und ein Ausblick.

2 Stand der Forschung

Die Herausforderungen an Container Terminals hinsichtlich Gestaltung und Produktivität werden in vielen wissenschaftlichen Studien untersucht. Die Publikationen zu diesem Thema sind in einigen Literaturübersichten zusammengefasst. Vis und de Koster [5] wie auch Steenken, Voß und Stahlbock [3] klassifizieren Entscheidungsprobleme auf Container Terminals und bieten jeweils eine breite Übersicht der Literatur vor allem der 1990er. Stahlbock und Voß [6] sowie Carlo, Vis und Roodbergen [7-9] erweitern und aktualisieren diese Übersicht, wobei Carlo, Vis und Roodbergen die drei Bereiche Wasserseite [8], Horizontaltransport [7] und Yard [9] separat untersuchen und das Klassifikationsschema jeweils erweitern.

Kim und Lee [10] legen den Fokus auf Trends bei Entscheidungsproblemen und konzentrieren sich besonders auf die Funktionen von Terminal Operating Systemen, der Software, die zur Entscheidungsunterstützung im Terminalbetrieb genutzt wird. Gharehgozli, Roy und de Koster [11] ergänzen die Literaturübersichten um einen Überblick über innovative Technologien auf Container Terminals und Trends in der Operations Research (OR). Angeloudis und Bell [12] fokussieren auf Simulationsmodelle, die den Betrieb von Container Terminals untersuchen. Stahlbock und Voß [13] betrachten vor allem Routingprobleme von Fahrzeugen auf Container Terminals.

Eine Reihe von Publikationen untersucht Zuordnungsmethoden auf Container Terminals. Dabei wird in der Literatur zwischen Dispatching und Scheduling un-

terschieden ([14], S.4). Beim Dispatching wird eine dynamische Zuordnung von Fahrzeugen und Aufträgen unter Berücksichtigung von kurzfristigen Änderungen vorgenommen. Es ist deswegen sehr flexibel. Im Gegensatz wird sich beim Scheduling mit einer langfristigen (d. h. mehrstündigen) und festen Zuordnung von Aufträgen beschäftigt. Dabei werden voraussichtliche Ankunftszeiten und Operationszeiten verwendet. Dieser Ansatz wird oftmals im Bereich des OR verordnet. Aufgrund der hohen Dynamik und Stochastizität der Abläufe auf Container Terminals (z. B. aufgrund des schwankenden Arbeitsaufkommen abhängig von den Ankunftsdaten von Schiffen, LKWs und Zügen, Ausfällen von Equipment, Verzögerungen beim Handling, schwierige Wetterbedingungen; vgl. [15], S. 592) werden in dieser Ausarbeitung ausschließlich die flexibleren dynamischen Zuordnungsmethoden, also das Dispatching betrachtet.

Eine feste Zuordnung von Fahrzeugen zu Containerbrücken findet sich u. a. in [16–18]. Dynamische Zuordnungsmethoden sind häufig Prioritätsregelverfahren wie „dichtester Auftrag“, „längste Wartezeit“ oder „geringste Fahrzeugauslastung“ und werden u. a. in [14,15,19,20] untersucht. Zuordnungsmethoden mit Vorausschauhorizont (z. B. der spätesten Ankunftszeit an der Containerbrücke) werden u. a. in [21–24] verwendet. Auf der Seite des Scheduling finden sich Publikationen wie [16,25–32], die u. a. ein mathematisches Modell formulieren und es entweder exakt lösen (was für eine Anwendung auf realen Container Terminals eine zu lange Rechenzeit erfordert) oder mit Hilfe von Heuristiken wie genetischen Algorithmen, Tabu Search oder Simulated Annealing eine möglichst gute Annäherung an die optimale Lösung ermitteln.

Eine Übersicht zum Dispatching und Scheduling im Horizontaltransport von Container Terminals findet sich in Schwientek, Lange und Jahn [33]. Ein Großteil der dort vorgestellten Publikationen betrachtet ausschließlich eine Fahrzeugkapazität von einem Container, unabhängig der Containergröße. Eine Kapazität von mehr als einem Container wird lediglich bei fünf Publikationen berücksichtigt.

Grunow, Günther und Lehmann [28] entwickeln einen Prioritätsregel-Ansatz, der auf einem erweiterten Konzept der Fahrzeug-Verfügbarkeit beruht. Zum Vergleich wird auch ein gemischt-ganzzahliges lineares Problem (Mixed-Integer Linear Programming, MILP) formuliert. Beide Ansätze werden anschließend für vier

verschiedene Szenarios hinsichtlich der absoluten Verspätung der AGVs analysiert. Verglichen werden dabei die Anwendung der Prioritätsregel für Single-load und Dual-load sowie die Lösung des MILP für Dual-load AGVs. Die durchschnittliche Verspätung der AGVs ist im Fall der Dual-load AGVs teils deutlich geringer. Weiterhin bringt die Anwendung des MILP Modells bessere Ergebnisse als die Prioritätsregel, jedoch zu Kosten eines deutlich höheren Rechenaufwands.

Grunow, Günther und Lehmann [15] bauen auf [28] auf und entwickeln ergänzend eine Muster-basierte statische Strategie für einen definierten Planungshorizont. Diese wird, basierend auf Experimenten, die Terminalgröße und die stochastische Varianz untersuchen, mit den Ergebnissen der Prioritätsregel verglichen. In allen Fällen ist die statische Strategie erfolgreicher als die dynamische. Eine steigende Stochastizität beeinflusst die Leistungsfähigkeit der Methoden negativ. Der Unterschied der Ergebnisse zwischen Single-load und Dual-load AGVs ist relativ gering, beim statischen Verfahren ist die Dual-load-Variante besser, beim dynamischen jedoch die Single-load-Variante.

Klerides und Hadjiconstantinou [34] erstellen eine mathematische Problemformulierung mit dem Ansatz eines rollierenden Horizonts und vereinfachen dabei die Formulierung von [28]. Zum Testen verwenden sie die Szenarios mit bis zu 18 Aufträgen von [28] und vergleichen drei Fälle: Dual-load AGVs mit einem rollierenden Horizont, Single-load AGVs mit einem rollierenden Horizont sowie Dual-load AGVs mit einem kompletten Zeithorizont. Als Vergleichskriterium dient die minimale Verspätung pro Auftrag. Diese ist bei Single-load mit dem rollierenden Horizont leicht höher und bei Dual-load und einem kompletten Zeithorizont leicht niedriger als bei der Kombination Dual-load und rollierender Horizont.

Nishimura, Imai und Papadimitriou [35] untersuchen die Potenziale von Multi-Trailer-Systemen mit bis zu sechs Trailern pro Zugmaschine. Sie erstellen Problemformulierungen sowohl für das Single-trailer als auch für das Multi-trailer Problem, lösen es mit Hilfe eines genetischen Algorithmus und vergleichen das Ergebnis mit der festen Zuordnung von Multi-Trailern zu jeweils einer Containerbrücke. Die flexible Zuordnung ist kostenseitig immer vorteilhaft, erhöht jedoch die Komplexität der Transporte. Die Nutzung von Multi-Trailern reduziert die gefahrenen Wege.

Petering [36] schlägt verschiedene Kontrollsysteme

für Dual-load Fahrzeuge vor und führen vier verschiedene Experimente mit einem Simulationsmodell durch. Zielgröße ist die *gross crane rate*, also die Produktivität der Containerbrücken in moves/h. Die Ergebnisse zeigen, dass eine bestands-basierte Zuordnungsmethode gegenüber einer zeitbasierten die *gross crane rate* positiv beeinflusst, dass es vorteilhaft ist, alle Fahrzeuge einem Terminal-weiten Pool zuzuordnen statt jeweils einem Schiff oder einer Containerbrücke und dass die Nutzung der Dual-load-Fähigkeit einen kleinen Vorteil bringt.

Grundsätzlich scheint es also vorteilhaft zu sein, die Dual-load-Kapazität der Fahrzeuge zu nutzen. Wie groß der Nutzen ist und ob das die zusätzliche Komplexität im System rechtfertigt, muss noch weiter untersucht werden. Keine der Studien untersucht in diesem Zusammenhang den Einfluss von Twin- und TEU-Faktor, also des Anteils der Twin-lift-Bewegungen der Containerbrücken sowie der 20'-Container.

3 Simulationsstudie

Im folgenden Abschnitt wird das zunächst das Simulationsmodell mit seinen Annahmen beschrieben und anschließend der Experimentplan mit allen variierten Parametern entwickelt.

3.1 Simulationsmodell

Im Rahmen einer Simulationsstudie wird der Entlade- und Beladeprozess in einem Container Terminal analysiert. Der Fokus liegt auf der Problematik der Auftragszuordnung für den Horizontaltransport. Angegliederte Planungsaufgaben wie die Zuordnung der Containerbrücken und Lagerplätze im Zwischenlager werden vereinfacht abgebildet. Für die Abfertigung durch die Containerbrücken ist beispielsweise lediglich relevant, dass eine Brücke nicht direkt neben einer anderen operieren kann. Hierbei muss mindestens der Abstand einer Ladebucht auf dem Schiff eingehalten werden (vgl. [1]). Nachgeordnete Prozesse, wie die Be- und Entladung von Lkw- und Bahnverkehren, werden nicht betrachtet, da sie nicht maßgeblich für die Abfertigungszeit des Schiffs sind. Zusätzlich wird angenommen, dass der Stauplan, also die Position der Container auf dem Schiff, gegeben ist.

Das Layout des Container Terminals ist festgelegt und bleibt während der Simulation statisch. Die Container werden durch Containerbrücken vom Schiff entladen. Je nach Schiffsgröße und Stauplan wird eine bestimmte

Anzahl von Ladebuchten durch die Brücken angesteuert. Der Horizontaltransport erfolgt durch Zugmaschinen mit einer Kapazität von zwei TEU, die je nach Experiment unterschiedlich genutzt wird. Als Senke des Systems ist das Blocklager mit den Lagerkränen zu sehen. Die Beladung des Schiffes erfolgt vice versa. Dabei ist anzumerken, dass jede einzelne Ladebucht des Schiffs zuerst vollständig entladen und anschließend beladen wird. Je nach Größe des Containerschiffs wird diesem eine bestimmte Anzahl von Containerbrücken und Lagerkränen bzw. Lagern zugewiesen.

Da nicht das Zusammenspiel von einer parallelen Abfertigung von mehreren Schiffen untersucht wird, sondern der bewusst gewählte Fokus auf der Auftragszuordnung liegt, wird je Simulation nur ein Schiff abgefertigt. Für dieses Schiff sind die Parameter grundsätzlich variabel, in der vorliegenden Analyse wird sich jedoch ein großes Schiff mit 23 Ladebuchten und insgesamt 16.000 Containern beschränkt. Das Schiff wird von sechs Containerbrücken mit einer Produktivität von 30 moves/h bedient. Im Containeryard gibt es 18 Lagerblöcke, in denen gummibereifte Portalkrane mit einer Produktivität von 15 moves/h die Fahrzeuge be- und entladen. Die Container werden gleichmäßig über das Lager verteilt. Es gibt 30 Fahrzeuge für den Horizontaltransport, die mit einer Geschwindigkeit von 25km/h auf dem Terminalgelände fahren. Die Annahmen für die Simulationsstudie beziehen sich nicht auf ein spezifisches reales Terminal. Sie sind jedoch mittels Literaturstudien und Expertengesprächen als realistisch eingeschätzt worden und werden im Laufe der Studie durch Sensitivitätsanalysen überprüft.

Es wird untersucht, wie sich durch die Veränderung unterschiedlicher Zuordnungsmethoden und Parametern die Liegezeit des Schiffes als wesentliche Ausgabegröße verändert. Weitere Analyse Kriterien sind die Auslastung der Containerbrücken, die Auftragsdauer und die Transportzeit der Fahrzeuge. Abbildung 1 stellt die Input- und Outputgrößen der Simulationsstudie dar.

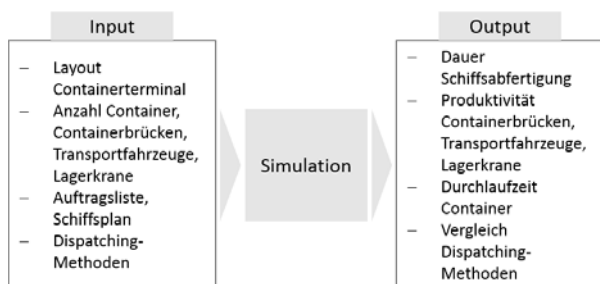


Abbildung 1: Input- und Outputgrößen der Simulation

3.2 Experimentplan

Wie bereits erläutert, verändert sich der Status eines Fahrzeugs durchgehend. Dabei ist es entweder leer, teilweise beladen (mit einem 20'-Container) oder vollständig beladen (mit zwei 20'-Containern oder einem 40'-Container). Die Auftragsvergabe wird nicht durch den Auftrag, sondern durch das Fahrzeug selbst ausgelöst. Sobald es zur Verfügung steht und ein Auftrag zu vergeben ist, wählt es selbstständig einen neuen Auftrag basierend auf einer Zuordnungsmethode aus.

Zuordnungsmethoden. Es werden in dieser Ausarbeitung vier verschiedene Zuordnungsmethoden von Aufträgen zu Fahrzeugen untersucht: Feste Zuordnung von Fahrzeugen zu Containerbrücken, Vergabe der Aufträge nach dem Bestandslevel der Containerbrücken in zwei Versionen und eine hybride Variante.

Die feste Zuordnung von Fahrzeugen des Horizontaltransports zu Containerbrücken ist ein in der Praxis weit verbreitetes Vorgehen. Es handelt sich dabei um eine relativ einfache und für das Personal des Terminals komfortable Methode, die deswegen aber auch deutliche Optimierungspotenziale, besonders zur Reduzierung von Leerfahrten, beinhaltet. Dabei ist eine feste Zahl von Fahrzeugen einer bestimmten Containerbrücke zugeordnet und führt ausschließlich Transporte für diese Brücke durch.

Bei der Vergabe von Aufträgen nach dem Bestandslevel wird für jede Containerbrücke der Bestandslevel ermittelt und für ein freies Fahrzeug der Auftrag ausgewählt, der zur Brücke mit dem geringsten Bestandslevel gehört (vgl. z. B. [14,21]). Als Bestandslevel wird dabei die Menge an Transportaufträgen für eine Containerbrücke verstanden, die bereits Fahrzeugen zugeordnet worden sind. Durch die Einbeziehung des Bestandslevels werden die Containerbrücken unterschiedlich priorisiert und eine Containerbrücke mit weniger Vorlauf an Aufträgen erhält eine höhere Priorität. Dabei wird zwischen zwei Varianten differenziert. In der Variante Bestandslevel (-) werden keine Aufträge zwischengeschaltet, während es bei der Variante Bestandslevel (+) möglich ist, dass ein Fahrzeug einen neuen Auftrag bearbeitet, während es noch einen 20'-Container beladen hat. Unter einer Zwischenschaltung wird also die Situation verstanden, dass ein Auftrag, der sich bereits in der Ausführung befindet, verzögert wird, um zwischendurch einen weiteren Auftrag zu beginnen. Die Variante Bestandslevel (-) ohne Zwischenschaltungen sorgt also dafür, dass ein einmal

begonnener Auftrag immer fertig ausgeführt wird, bevor ein neuer Auftrag ausgewählt wird.

Bei der hybriden Variante wird die Bestandslevel-Methode um weitere Containerbrücken-bezogene Parameter erweitert. Dabei erhält jeder Auftrag einen Prioritätswert basierend auf dem entsprechenden Bestandslevel, dem Auftragsstand der Containerbrücke (je mehr Aufträge zur Ent- und Beladung der jeweiligen Ladebucht anstehen, desto höher der Wert), dem Vorhandensein einer nachfolgend zu ent- und beladenden Ladebucht sowie der möglichen Blockierung einer anderen Containerbrücke.

Varierte Parameter. Der Experimentplan unterteilt sich in zwei Stufen. In der ersten Stufe werden zunächst die genannten Zuordnungsmethoden in Abhängigkeit der Fahrzeugkapazität untersucht. Eine Fahrzeugkapazität von 1 bedeutet dabei, dass immer nur ein 20'- oder 40'-Container pro Fahrt transportiert wird, auch wenn sich rein physikalisch an der Fahrzeuggröße nichts ändert.

Twin-Faktor	TEU-Faktor	Fahrzeug-Kapazität	Zuordnungsmethode			Hybrid
			Feste Zuordnung	Bestandslevel (-)	Bestandslevel (+)	
Stufe 1						
60 %	1,5	1	IF		IB	IH
		2	IIF	IIB(-)	IIB(+)	IIH
Stufe 2						
40 %	1,3	1	F1	B(-)1	B(+)	H1
	1,5		F2	B(-)2	B(+)	H2
	1,7		F3	B(-)3	B(+)	H3
60 %	1,3	2	F4	B(-)4	B(+)	H4
	1,5		F5	B(-)5	B(+)	H5
	1,7		F6	B(-)6	B(+)	H6
80 %	1,3	2	F7	B(-)7	B(+)	H7
	1,5		F8	B(-)8	B(+)	H8
	1,7		F9	B(-)9	B(+)	H9

Tabelle 1: Experimentplan

In der zweiten Stufe wird eine Fahrzeug-Kapazität von 2

betrachtet, aber dabei der Twin-Faktor und der TEU-Faktor variiert. Der Twin-Faktor beschreibt dabei den Anteil der 20'-Container, die von der Containerbrücke im Twinlift-Modus umgeschlagen werden. Ein Twin-Faktor von 60 % besagt also, dass 60 % der 20'-Container zu zweit umgeschlagen werden und 40 % der 20'-Container einzeln umgeschlagen werden. Dabei ist ein Wert von 100 % nicht realistisch, da es u.a. abhängig von Schiffsgröße, Stauplan und Laderaumdesign in jedem Fall Stellplätze für alleinstehende 20'-Container gibt. In den Experimenten wird der Twin-Faktor zwischen 40 % und 80 % variiert. Der TEU-Faktor als zweiter variierte Parameter gibt das Verhältnis zwischen 20'- und 40'-Containern an. Dabei beschreibt z. B. ein TEU-Faktor von 1,7 den Zustand, dass es sich bei 70 % der Container um 40'-Container handelt und bei 30 % der Container um 20'-Container. In den Experimenten wird der TEU-Faktor zwischen 1,3 und 1,7 variiert. Tabelle 1 stellt den Experimentplan mit allen Variationen für beide Stufen dar.

4 Ergebnisse

Im Folgenden werden die Ergebnisse der Simulationsstudie dargestellt. Zunächst wird in Abschnitt 4.1 untersucht, wie sich die Veränderung der Fahrzeugkapazität unter Berücksichtigung der vier genannten Zuordnungsmethoden auf die Leistungsgrößen Schiffsentladezeit, Auslastung der Containerbrücken, Transportzeit und Auftragszeit auswirkt. Abschnitt 4.2 gibt die Ergebnisse der zweiten Stufe des Experimentplans wieder, in der Twin- und TEU-Faktor variiert werden, und analysiert diese ebenfalls hinsichtlich der Leistungsgrößen. Anschließend wird in Abschnitt 4.3 die Verifikation und Validierung durchgeführt. Abschließend wird in Abschnitt 4.4 die Sensitivität des Simulationsmodells auf eine Variation der Parameter Fahrzeuggeschwindigkeit, Fahrzeuganzahl, Produktivität des Umschlagequipments und Lagerzuordnung untersucht.

4.1 Stufe 1: Variation der Fahrzeugkapazität

Grundsätzlich führt eine Erhöhung der Kapazität der Fahrzeuge von 1 auf 2 dazu, dass sich die Liegezeit des Schiffs um durchschnittlich 8 % verkürzt und das System somit leistungsfähiger wird (siehe Abbildung 2). Dies liegt daran, dass es mehr Transportaufträge gibt, die gemeinsam durchgeführt werden, wodurch sich die Transportzeit pro Auftrag verringert. Die Transportzeit der

Container ist bei den Experimenten mit der Fahrzeugkapazität 1 durchschnittlich 9 % länger als bei der Nutzung der Dual-load Kapazität. Ebenso ist die Transportzeit bei der festen Zuordnung von Fahrzeugen zu Containerbrücken länger als bei der Bestandslevel- und der hybriden Methode. Dies liegt unter anderem an einer größeren Anzahl von Leerfahrten.

Weiterhin wird die Zuordnungsmethode *Feste Zuordnung* von den anderen Methoden dominiert, wobei die hybride Methode mit Kapazität 1 sogar leistungsfähiger ist als die feste Zuordnung mit der Kapazität 2. Am besten schneidet bei beiden Kapazitäten die hybride Variante ab, wobei der Abstand zur bestandsbasierten Methode nur etwa 1 % beträgt.

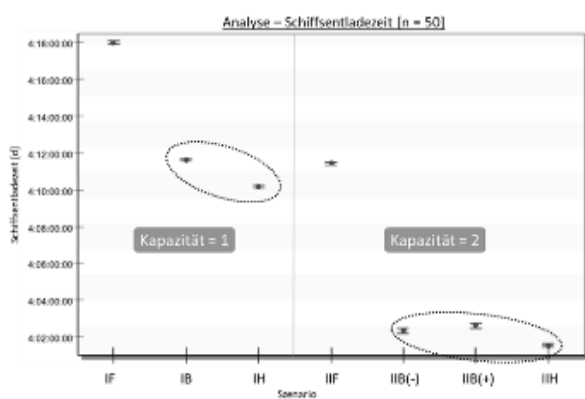


Abbildung 2: Darstellung des Schiffsentladezeit in Abhängigkeit der gewählten Zuordnungsmethode und der Fahrzeugkapazität

Die Auslastung der Containerbrücken ist höher, wenn die Kapazität 1 beträgt. Dies lässt sich damit begründen, dass die Brücken bei einer Fahrzeugkapazität von 1 keine Twin-lift Umschläge durchführen und somit insgesamt mehr Bewegungen haben.

Die Auftragszeit ist bei den bestandsbasierten Methoden am kürzesten, hier wird die hybride Methode klar dominiert. Dies liegt daran, dass diese Zuordnungsmethode stark auf eine Minimierung der Schiffs Liegezeit abzielt und dadurch einzelne Brücken priorisiert, wodurch insgesamt die Wartezeiten der Fahrzeuge an den Brücken ansteigen.

4.2 Stufe 2: Variation von Twin- und TEU-Faktor

Die feste Zuordnungsmethode wird auch bei den Experimenten der zweiten Stufe bezüglich der Liegezeit des

Schiffs von den anderen Methoden dominiert. Zu erkennen ist dies in Abbildung 3 an den drei Linien im oberen Teil der Abbildung. Die hybride Methode (H1-H9) führt bei den meisten untersuchten Szenarios zur kürzesten Abfertigungszeit des Schiffs. Die beiden Varianten der Bestandslevel-Methode sind ähnlich leistungsfähig. Dies zeigt, dass es keinen nennenswerten Nutzen bringt, Zwischenschaltungen von Aufträgen zu erlauben, wenn ein Fahrzeug nur mit einem 20'-Container beladen ist. Teilweise ist sogar die Bestandslevel-Variante ohne Zwischenschaltungen B(-) schneller als jene mit Zwischenschaltungen.

Die dunkelblauen Trendlinien zeigen den Einfluss des TEU-Faktors, der in den Szenarios von 1,3 über 1,5 auf 1,7 ansteigt. Insgesamt verlängert sich die Liegezeit des Schiffs mit steigendem TEU-Faktor. Dabei fällt auf, dass der Anstieg zwischen 1,3 und 1,5 nicht so stark ist wie zwischen 1,5 und 1,7. Die Dreiecke in Abbildung 3 zeigen den Einfluss des Twin-lift-Anteils. Je höher der Anteil der Twin-lifts der Containerbrücken ist, umso geringer ist die Liegezeit des Schiffs.

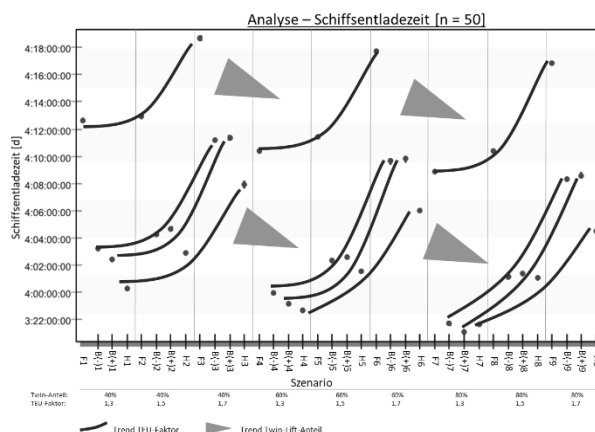


Abbildung 3: Darstellung der Schiffsentladezeit in Abhängigkeit der gewählten Zuordnungsmethode sowie des Twin- und TEU-Faktors

Bezüglich der Brückenauslastung führen, wie in der ersten Stufe, die Bestandslevel-Methoden zu einer höheren Auslastung. Der Unterschied zwischen den Methoden ist bei einem TEU-Faktor von 1,7 nur noch gering. Die Auslastung nimmt bei einer Steigerung des TEU-Faktors von 1,3 auf 1,5 stärker (etwa 5 %) zu als bei einer Steigerung von 1,5 auf 1,7 (etwa 2 %). Bei einer Erhöhung des Twin-lift-Anteils wird die Abfertigung beschleunigt und dadurch sinkt die Auslastung der Brücken.

Die Transportzeit der Fahrzeuge sinkt wie schon in

Stufe 1, wenn statt der festen Zuordnung die Bestandslevel- oder die hybride Zuordnungsmethode gewählt wird. In den meisten Fällen ist die hybride Methode am vorteilhaftesten. Eine Ausnahme stellen die Szenarien mit einem TEU-Faktor von 1,3 dar. Dort ist die Bestandslevel-Variante mit Zwischenschaltungen B(+) am besten. Grundsätzlich sinkt die Transportzeit bei steigenden Twin-lift-Anteil und steigt mit steigendem TEU-Faktor. Bei einem TEU-Faktor von 1,7 liegen die Ergebnisse der Methoden insgesamt dichter beieinander als bei einem geringeren TEU-Faktor.

Bei der Betrachtung der Auftragszeit führen wie in Stufe 1 die Bestandslevel-Varianten zu einer geringeren mittleren Auftragszeit während die hybride Methode klar dominiert wird. Indem einzelne Brücken höher priorisiert werden, erhöhen sich dort die Wartezeiten der Fahrzeuge und damit die Auftragszeiten. Auch hier führen die Zwischenschaltungen bei den Bestandslevel-Varianten zu keinem signifikanten Nutzen. Bei einem höheren TEU-Faktor ist die feste Zuordnung vergleichbar gut wie die Bestandslevel-Methoden. Auch in diesem Fall nimmt durch eine Erhöhung des Twin-Lift-Anteils die Auftragszeit ab.

4.3 Verifikation und Validierung

Das Simulationsmodell bezieht sich nicht auf ein reales Terminal, daher kann im Rahmen der Verifikation und Validierung nicht auf Realdaten zurückgegriffen werden. Im Folgenden wird daher vor allem geprüft, ob die durch das Modell erzeugten Daten realistische Werte aufweisen.

Zunächst werden die Containerbrücken im Modell untersucht. Die Anzahl der bearbeiteten Aufträge je Brücke variiert in einem realistischen Rahmen, die Gesamtzahl über alle Brücken stimmt mit der Zahl der insgesamt umgeschlagenen Container überein. Anhand der Start- und Endzeiten der Containerbrücken an den einzelnen Ladebuchten des Schiffs lässt sich sehen, dass die Ladebuchten jeweils nacheinander abgearbeitet werden und die Verfahrzeiten der Containerbrücken zwischen den Ladebuchten eingehalten werden. Auch Blockierungen der Containerbrücken sind ablesbar, wenn die Bearbeitung einer Ladebucht erst verzögert beginnt, weil an der benachbarten Ladebucht noch eine Containerbrücke aktiv ist.

Auch die Fahrzeugdaten sind plausibel. Die Summe der Aufträge je Fahrzeug stimmt wiederum mit der Gesamtzahl überein. Die Summe der Transportzeiten, der

Wartezeiten an den Ladebuchten bzw. im Containeryard sowie die Umschlagszeiten bewegen sich in einem plausiblen Rahmen.

4.4 Sensitivitätsanalysen

In den Sensitivitätsanalysen werden Variationen der Geschwindigkeit der Fahrzeuge, der Anzahl der Fahrzeuge, der Produktivität des Umschlagequipments und des Lagerzuordnungsverhalten hinsichtlich ihrer Auswirkungen auf die Ergebnisse der Simulationsstudie untersucht.

Geschwindigkeit der Fahrzeuge. Im Rahmen der Sensitivitätsanalyse wird die Geschwindigkeit zwischen 20, 25 und 30 km/h variiert. Mit steigender Geschwindigkeit der Fahrzeuge wird das Schiff insgesamt schneller abgefertigt und die Auslastung der Containerbrücken steigt. Dabei ist der Anstieg von 20 auf 25 km/h größer als von 25 auf 30 km/h. Der gleiche Zusammenhang ist bei der Transportzeit zu sehen. Eine Reduzierung der Fahrzeuggeschwindigkeit führt zu einer Erhöhung der Gesamttransportzeit. Wiederum ist die Differenz zwischen 20 und 25 km/h größer als zwischen 25 und 30 km/h.

Anzahl der Fahrzeuge. Wenn nicht ausreichend Fahrzeuge für Verfügung stehen, werden Aufträge mit Verzögerung durchgeführt und es entstehen Wartezeiten für die Containerbrücken, die letztendlich auch die Liegezeit des Schiffs negativ beeinflussen. In der Sensitivitätsanalyse wird zwischen 24, 30 und 36 Fahrzeugen differenziert, was 4, 5 bzw. 6 Fahrzeugen pro Containerbrücken entspricht. Insgesamt steigt mit der Anzahl der Fahrzeuge auch die Auslastung der Containerbrücken.

Die Schiffliegezeit bewegt sich je nach Experiment zwischen etwas mehr als 5 Tagen (Experiment IIF mit 24 Fahrzeugen) und 3 Tagen und 20 Stunden (Experiment IIH mit 36 Fahrzeugen). Dabei ist die Steigerung von 24 auf 30 Fahrzeuge stärker als von 30 auf 36 Fahrzeuge. Dieser Zusammenhang spiegelt sich auch in der Auftragszeit. Bei einer größeren Anzahl an Fahrzeugen entstehen zwar weniger Wartezeiten für die Containerbrücken, weil die Aufträge schneller zugeordnet werden, es kommt jedoch zu längeren Wartezeiten für die Fahrzeuge an den Brücken und im Containeryard. Beispielsweise beträgt die Auftragszeit bei der festen Zuordnung 9:20 min für 24 Fahrzeuge und 11:35 min für 36 Fahrzeuge. Dies bedeutet, dass der gesamte Prozess der

Schiffsabfertigung schneller vorstättengeht, die einzelnen Transporte jedoch ineffizienter sind.

Produktivität des Umschlagequipments. Die Produktivität des Umschlagequipments ergibt sich aus dem Zeitaufwand, der benötigt wird, um ein Fahrzeug zu be- oder entladen. In den Experimenten wurde ein Zeitaufwand von 120 s angenommen, was einer Produktivität von 30 moves/h entspricht. In der Sensitivitätsanalyse wird der Zeitaufwand um 30 s verkürzt bzw. verlängert, was einer Produktivität von 24 bzw. 40 moves/h entspricht.

Diese Variation hat einen starken Einfluss auf die Simulationsergebnisse. Bei einer festen Zuordnung von Fahrzeugen zu Containerbrücken schwankt die Liegezeit des Schiffs um 15 h. Je langsamer das Equipment, desto länger die Liegezeit, aber desto höher die Auslastung. Bei schnellerem Equipment ist der gegenteilige Effekt zu beobachten. In diesem Fall wäre es sinnvoll, mehr Fahrzeuge einzusetzen, um den Engpass an der Brücke auszugleichen. Weiterhin werden die Abstände zwischen den Ergebnissen der Zuordnungsmethoden geringer, wenn die Containerbrücken langsamer umschlagen.

Lagerzuordnung. In den Experimenten wurde angenommen, dass die Container gleichmäßig über das Lager verteilt werden. Dies hat zur Folge, dass beim Transport von zwei 20'-Containern in der Regel zwei verschiedene Lagerblöcke angefahren werden. In dieser Analyse wird nun untersucht, welchen Einfluss es hat, wenn im Twin-lift-Fall die Container mit einer Wahrscheinlichkeit von 80 % für dasselbe Lager bestimmt sind. Dies ließe sich dadurch erklären, dass zwei Container mit ähnlichen Eigenschaften wie Destination, Größe und Gewicht auf einem Schiff eher nebeneinander gestaut werden und somit für Twin-lifts in Betracht kommen und dann auch eher im Dual-load transportiert werden.

Bei der Liegezeit hat die Variation, dass meistens das gleiche Lager angefahren wird, den Effekt, dass die Liegezeit sich bei allen Zuordnungsmethoden leicht verlängert. Dies hängt damit zusammen, dass sich zwar die durchschnittliche Transportzeit pro Auftrag verkürzt, die durchschnittlich Auftragszeit sich jedoch verlängert. Vor allem die Umschlagszeit und die Wartezeit, die die beiden anderen Komponenten der Auftragszeit darstellen, steigen in diesem Fall an, was nicht durch die eingesparte Transportzeit kompensiert wird. Eine angepasste Lagerzuordnung scheint also vor allem dann sinnvoll zu sein,

wenn im Lager zwei Container gleichzeitig entladen werden können oder die Transportwege sehr weit sind.

5 Fazit

Diese Simulationsstudie verfolgt das Ziel zu ermitteln, welchen Einfluss die Fahrzeugkapazität auf die Leistungsfähigkeit der Zuordnungsmethoden der Aufträge im Horizontalverkehr hat und verschiedene Parameter eines Container Terminals diese Zusammenhänge beeinflussen.

Eine feste Zuordnung von Fahrzeugen zu Containerbrücken führt zu einem hohen Anteil an Leerfahrten. Die Fahrzeuge stehen nicht als Pool zur Verfügung und diese Szenarien werden somit stets von anderen dominiert. Somit ist unabhängig von der Nutzung der Fahrzeugkapazität ein Wechsel zu einer anderen der untersuchten Zuordnungsmethoden vorteilhaft.

Eine erhöhte Kapazität der Fahrzeuge sorgt insgesamt für eine Verbesserung der Produktivität und Beschleunigung der Schiffsabfertigung. Allerdings steigt damit auch die Komplexität. Das betrifft nicht nur den Modellierungsaufwand für die Simulation, sondern auch die Anwendung in der Praxis. Wenn ein zusätzlicher Auftrag bereits angefangen wird, bevor der vorherige komplett abgeschlossen wurde, dann entstehen Zwischenschaltungen. Grundsätzlich haben diese einen positive Effekt auf die Transportzeit, jedoch ist der Unterschied zwischen den Ergebnissen für B(-) und B(+) nicht so groß. Daher sollten die steigende Komplexität und damit auch die Gefahr, dass bei den Fahrten Fehler geschehen, gegen das Einsparpotenzial sorgfältig abgewogen werden.

Ein Einsatz der eigens erstellten hybriden Zuordnungsmethode kann die Liegezeit des Schiffs verringern. Allerdings besteht die Hauptursache des Erfolgs in der ungleichmäßigen Auslastung der Containerbrücken. Das Ziel der Methode ist, dass alle Brücken am Ende zusammen fertig sind. Brücken mit mehr Aufwand für den weiteren Verlauf werden priorisiert. Hierbei wird nicht berücksichtigt, ob fertige Brücken ggf. für ein anderes Schiff zum Einsatz kommen könnten.

Der TEU-Faktor und der Twin-lift-Anteil haben einen großen Einfluss auf die Leistungsfähigkeit der Zuordnungsmethoden. Dies sollte also bei einer praktischen Anwendung differenziert betrachtet werden.

Für zukünftige Untersuchungen gibt noch diverse Ansatzpunkte, da es sich bei einem Container Terminal um ein sehr komplexes System handelt. Zum Beispiel

könnte die Anzahl der Schiffe erweitert werden und somit das Zusammenspiel verschiedener Schiffsgrößen untersucht werden. Ebenso ist eine Analyse weiterer Zuordnungsmethoden sinnvoll. Weiterhin wird an dieser Stelle die Fahrzeug-Kapazität von eins auf zwei erhöht. Grundsätzlich möglich, jedoch selten genutzt, sind auch Multi-Trailer-Systeme, die ein großes Potenzial bereitstellen könnten.

References

- [1] Linkosaari T. A Functional Mathematical Optimization Algorithm for the Integration of the Tactical Berth, Crane and Vehicle Scheduling. In: Blecker T, Kersten W, Ringle CM, editors. *Innovative methods in logistics and supply chain management: Current Issues and Emerging Practices*, 1st ed. Hamburg-Harburg: Universitätsbibliothek der Technischen Universität Hamburg-Harburg; 2014, 85-97.
- [2] Brinkmann B. *Seehäfen - Planung und Entwurf*. Berlin: Springer; 2005.
- [3] Steenken D, Voß S, Stahlbock R. Container terminal operation and operations research - a classification and literature review. *OR Spectrum* 2004;26:3-49.
- [4] Kemme N. *Design and Operation of Automated Container Storage Systems*. Heidelberg: Physica-Verlag HD; 2013.
- [5] Vis IF, Koster R de. Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research* 2003;147:1-16.
- [6] Stahlbock R, Voß S. Operations research at container terminals: A literature update. *OR Spectrum* 2007;30:1-52.
- [7] Carlo HJ, Vis IF, Roodbergen KJ. Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research* 2014;236:1-13.
- [8] Carlo HJ, Vis IF, Roodbergen KJ. Seaside operations in container terminals: Literature overview, trends, and research directions. *Flex Serv Manuf J* 2015;27:224-62.
- [9] Carlo HJ, Vis IF, Roodbergen KJ. Storage yard operations in container terminals: Literature overview, trends, and research directions. *European Journal of Operational Research* 2014;235:412-30.
- [10] Kim KH, Lee H. Container Terminal Operation: Current Trends and Future Challenges. In: Lee C-Y, Meng Q, editors. *Handbook of Ocean Container Transport Logistics*. Cham: Springer International Publishing; 2015, p. 43-73.
- [11] Gharegozli AH, Roy D, Koster R de. Sea container terminals: New technologies and OR models. *Marit Econ Logist* 2016;18:103-40.
- [12] Angeloudis P, Bell MG. A review of container terminal simulation models. *Maritime Policy & Management* 2011;38:523-40.
- [13] Stahlbock R, Voß S. Vehicle Routing Problems and Container Terminal Operations – An Update of Research. In: Golden B, Raghavan S, Wasil E, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Boston, MA: Springer US; 2008, p. 551-89.
- [14] Schwientek A, Lange A-K, Jahn C. Simulation-Based Analysis of Dispatching Methods on Seaport Container Terminals. In: Freitag M, Kotzab H, Pannek J, editors. *Dynamics in Logistics*: Springer International Publishing; 2018, p. 167-71.
- [15] Grunow M, Günther H-O, Lehmann M. Strategies for dispatching AGVs at automated seaport container terminals. *OR Spectrum* 2006;28:587-610.
- [16] Bae JW, Kim KH. A pooled dispatching strategy for automated guided vehicles in port container terminals. *International Journal of Management Science* 2000;6:47-67.
- [17] Böse JW, Reiners T, Steenken D, Voß S. Vehicle Dispatching at Seaport Container Terminals Using Evolutionary Algorithms. In: Sprague RH, editor. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences: Abstracts and CD-ROM of full papers January 4-7, 2000, Maui, Hawaii*. Los Alamitos, Calif.: IEEE Computer Society Press; 2000.
- [18] Garro A, Monaco MF, Russo W, Sammarra M, Sorrentino G. Agent-based simulation for the evaluation of a new dispatching model for the straddle carrier pooling problem. *SIMULATION* 2015;91:181-202.
- [19] Koster R de, Le-Anh T, Meer JR van der. Testing and classifying vehicle dispatching rules in three real-world settings. *Journal of Operations Management* 2004;22:369-86.
- [20] Tao J, Qiu Y. A simulation optimization method for vehicles dispatching among multiple container terminals. *Expert Systems with Applications* 2015;42:3742-50.
- [21] Briskorn D, Drexel A, Hartmann S. Inventory-based dispatching of automated guided vehicles on container terminals. *OR Spectrum* 2006;28:611-30.
- [22] Choe R, Kim J, Ryu KR. Online preference learning for adaptive dispatching of AGVs in an automated container terminal. *Applied Soft Computing* 2016;38:647-60.
- [23] Kim KH, Bae JW. A Look-Ahead Dispatching Method for Automated Guided Vehicles in Automated Port Container Terminals. *Transportation Science* 2004;38:224-34.
- [24] Song LQ, Huang SY. A hybrid metaheuristic method for dispatching automated guided vehicles in container terminals. In: *2013 IEEE Symposium on Computational Intelligence in Scheduling (SCIS): 16 - 19 April 2013, Singapore*; [part of the] *2013 IEEE Symposium Series on Computational Intelligence (SSCI)*. Piscataway, NJ: IEEE; 2013, p. 52-9.
- [25] Bian Z, Mi W, Yang X, Zhao N, Mi C. Modified Hungarian Algorithm for Real-Time ALV Dispatching Problem in Huge Container Terminals. *JNW* 2014;9.
- [26] Bian Z, Yang Y, Mi W, Mi C. Dispatching Electric AGVs in Automated Container Terminals with Long Travelling Distance. *Journal of Coastal Research* 2015;73:75-81.
- [27] Corman F, Xin J, Negenborn RR, D'Ariano A, Samà M, Toli A, Lodewijks G. Optimal Scheduling and Routing of Free-range AGVs at Large Scale Automated Container Terminals. *Periodica Polytechnica Transportation Engineering* 2016;44:145-54.
- [28] Grunow M, Günther H-O, Lehmann M. Dispatching multi-load AGVs in highly automated seaport container terminals. *OR Spectrum* 2004;26:211-35.
- [29] He J, Huang Y, Yan W, Wang S. Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption. *Expert Systems with Applications* 2015;42:2464-87.

- [30] Kaveshgar N, Huynh N. Integrated quay crane and yard truck scheduling for unloading inbound containers. *International Journal of Production Economics* 2015;159:168–77.
- [31] Ng JWC, Mak KL, Zhang YX. Scheduling trucks in container terminals using a genetic algorithm. *Engineering Optimization* 2007;39:33–47, doi:10.1080/03052150600917128.
- [32] Nguyen VD, Kim KH. A dispatching method for automated lifting vehicles in automated port container terminals. *Computers & Industrial Engineering* 2009;56:1002–20.
- [33] Schwientek A, Lange A-K, Jahn C. Literature classification on dispatching of container terminal vehicles. In: Jahn C, Kersten W, Ringle CM, editors. *Proceedings of the Hamburg International Conference of Logistics (HICL)/Digitalization in Maritime and Sustainable Logistics: City Logistics, Port Logistics and Sustainable Supply Chain Management in the Digital Age*, 2nd ed. Berlin: epubli; 2017, p. 3–36.
- [34] Klerides E, Hadjiconstantinou E. Modelling and solution approaches to the multi-load AGV dispatching problem in container terminals. *Marit Econ Logist* 2011;13:371–86.
- [35] Nishimura E, Imai A, Papadimitriou. Yard trailer routing at a maritime container terminal. *Transportation Research Part E: Logistics and Transportation Review* 2005;41:53–76.
- [36] Petering MEH. Development and simulation analysis of real-time, dual-load yard truck control systems for seaport container transshipment terminals. *OR Spectrum* 2010;32:633–61.

Influence of TAS' characteristics on the related drayage network

Ann-Kathrin Lange^{1*}, Kristof Ole Kühl¹; Anne Kathrina Schwientek¹, Carlos Jahn¹

¹Institute of Maritime Logistics, Hamburg University of Technology, Am Schwarzenberg-Campus 4, 21073 Hamburg

Abstract. Truck drayage transports in the port connect container terminals with other logistics nodes as empty container depots, freight stations or customs stations. Due to the large proportion of drayage transports in the overall truck arrivals at container terminals and their relatively high costs in the maritime transport chain, drayage transports have high importance in port processes. To reduce peaks in truck arrivals, container terminals more and more often implement truck appointment systems (TAS), which require trucking companies to book specific time windows for handling prior to their transports. Besides on their impact on the container terminals, these TAS also effect the other stakeholders in the drayage network, which has been neglected in scientific studies so far. This study aims to analyze the effect of TAS capacity and utilization on the arrival times at other logistics nodes as well as on the number of successfully executed orders per truck.

Introduction

Containerized maritime transport grows continuously since 2010 as well as before 2009 [1]. To enable lower prices and to diminish the emissions per transported container, shipping companies are ordering larger and larger vessel sizes. In 2006, the largest ship by far was the Emma Maersk with a capacity of about 15,000 Twenty-Foot equivalent Units (TEU). In contrast, the largest ship in 2017, the OOCL Hong Kong, has a capacity of over 21,000 TEU. The growing vessel sizes pose many different challenges for the container terminals as well as for the overall port area. When considering the landside operations of a container terminal, the main challenge is the peaks in truck arrivals causing the terminal gate to have either too few personell to prevent long queues reaching the access roads to the terminal or too much personal causing high costs for the terminal. Furthermore, the waiting trucks lead to high

CO₂-emissions due to their running engines. To mitigate these effects, an effective solution is to implement a truck appointment system (TAS) at the container terminal. A TAS is a vehicle booking system used to control the number of trucks arriving at the terminal at different times of the day. With this system, the trucking companies have to book specific time windows to deliver or pick up a container at the terminal. This does not only affect the container terminals and the trucking companies, but also other operative companies in the port, due to shifts in truck arrivals and higher restrictions in the dispatching process of trucking companies.

The aim of this study is to analyze the effects of varying TAS' characteristics on different operative stakeholders in the port network. To do so, many different characteristics, e.g. the length of the time window or the capacity per time window, are studied on their singular and combined impact on the arrival times at logistics nodes as well as on the amount of successfully executed orders per truck. First of all, the state of research is presented. Afterwards the simulation study and the experimental design are described. Finally, the results of the study are outlined and a conclusion is given.

1 Port Drayage Operations

Port drayage is defined as “truck pickup from or delivery to a seaport, with the trip origin and destination in the same urban area” ([2],[3]). Sometimes, drayage transports are also called inter-terminal transports (ITT), with the one difference, that ITT always take place between different container handling areas in one port and does not consider other logistics nodes as freight or customs stations. The main cause for drayage transports is the necessity to transport transshipment containers from one terminal to the next. On container terminals, there are mainly three types of orders: import, export and transshipment. Import container arrive at the terminals via ocean carrier and leave

on barge, train or truck. Export containers arrive vice versa per barge, truck or train and leave per ocean carrier. Transshipment containers are discharged from one vessel, stored, and afterwards loaded on another vessel. As not all vessels stop at all container terminals in one port, many transshipment containers need to be transported to other terminals, mainly by truck and sometimes by train or barge. Furthermore, empty containers, which often have long storage times, are mainly transported to empty container depots to save storage space on seaport container terminals. Other containers are packed or unpacked in freight stations or need to be transported to customs stations. All these transports are called drayage transports if the origin and destination are in the same area. Due to their large proportion of truck arrivals at container terminals and their relatively high costs, they present an important part of the overall maritime supply chain (inter alia [4], [5]).

2 State of Research

The first TAS was introduced in the ports of Los Angeles and Long Beach in 2002 in response to California Assembly Bill (AB) 2650. The evaluation of the program was mixed, due to high entry barriers for the truck drivers, as varying systems for registration and a generally high effort for the overall process, and therefore due to a low participation. [6] Main reason for the introduction of a TAS is the need to reduce CO₂-emissions. Other goals are to reduce truck waiting times at the gates or to improve the terminal processes. Therefore, the TAS as well as other approaches, as webcams at the terminal gates to provide information about the current congestion (e.g. [3]) or to promote transports outside peak times by introducing varying tolls in the port (e.g. [7]), have been studied increasingly. Today, several successful TAS are running in different parts of the world, e.g. Vancouver, Sydney and Southampton, but the development goes on to improve these systems or to find better alternatives (inter alia [8], [9]).

Scientific studies focus mainly on the effects of TAS on container terminal productivity (e.g. [10], [11]). Other publications also consider the possible benefits and challenges for trucking companies (e.g. [12], [13]). To the authors' knowledge, other actors, as empty container depots, freight stations or customs station, are never considered. For a comprehensive overview about literature on port drayage transports and TAS be referred to Lange et al. 2017 [14].

3 Simulation Study

As described above, the focus of this study is on port drayage transports. Therefore, only transports in the port area and between the relevant operative stakeholders are considered. For the simulation study, the program Tecnomatix Plant Simulation 13 is used because it is widely recognized in industry and this research area. The simulation model is generated based on operative data of different stakeholders in the port of Hamburg, especially transport data from trucking companies and process durations from various logistics nodes. A simulation run covers one workday from 0 to 24 o'clock. Every simulation experiment is repeated 20 times.

For the simulation model, a flexible list of orders for one considered trucking company is generated. In this list, the source and the drain of every order is noted. In the next step, time windows are booked for all orders with either source, drain or both at container terminals. For all time windows specific probabilities for a successful booking are considered, based on the assumed utilization of this time window by other trucking companies. The order list is imported in the simulation model. There, all relevant stakeholders are displayed. In every simulation run, one trucking company, four container terminals, six empty container depots, six freight stations and five other logistics nodes are considered. The driving distances between the individual stakeholders are represented by a distance relation matrix, considering the driving durations at different times of the day due to traffic. The durations are determined by using a Google Maps API for the relevant routes between logistics nodes in the port of Hamburg. Similarly, the handling times at the various logistic nodes differ based on the time of the day. At the beginning of every simulation run, the transport orders for the trucking company are checked and each truck is assigned one order. When the first order is completed, a next order is chosen for the truck. When a truck arrives too late at its destination, the order is cancelled and a new order is assigned.

4 Experimental Design

The parameters for the simulation experiments were determined in interviews with different stakeholders in the drayage network in the port of Hamburg. In addition, relevant scientific publications were analyzed. In the simulation experiments, a medium-sized truck-

ing company with 25 trucks and 375 possible orders per day is considered. 50 % of the transports are executed between container terminals and 8.3 % each between container terminals and empty container depots, packing stations and the remaining logistics nodes, vice versa. Transports only between empty container depots, freight stations and other logistics nodes are not considered. Furthermore, the working times of all logistics nodes, except container terminals, which always have three shifts, are limited. Therefore, they restrict the productivity of drayage transports as well as the efficiency of TAS at container terminals. As for TAS' characteristics, the capacity of the time windows in the different shifts (based on current demand, lightly smoothed and heavily smoothed) is varied. In the first shift there are either 40, 60 or 80 slots in total. In the second shift there are 120, 100 or 80 and in the third are always 80 slots. Furthermore, the utilization of the time windows by other trucking companies (80 %, 85 % and 90 %) is varied for the peak times between 9 am and 6 pm. The peak time has been set based on data provided by logistics nodes in the port of Hamburg. The overview of all experiments and their parameters is shown in Table 1.

Capacity	Utilization		
	80 %	85 %	90 %
Realistic	Exp. 1	Exp. 4	Exp. 7
Lightly smoothed	Exp. 2	Exp. 5	Exp. 8
Heavily smoothed	Exp. 3	Exp. 6	Exp. 9

Table 1: Plan of experiments

5 Simulation Results

The results show a high impact of the two selected TAS' characteristics on the operative stakeholders. In Figure 1, the amount of successfully executed orders is shown.

It is surprising to see that the heavily smoothed TAS capacity leads to a better solution as well as the high capacity utilization by other trucking companies. This effect is very likely caused by the structure of transport orders generated. As a high amount of transports is executed between container terminals and only a lesser amount between container terminals and other logistics nodes, vice versa, the trucks are able to pick up and deliver containers 24/7. In the simulation model, the time windows at peak times have a higher priority and

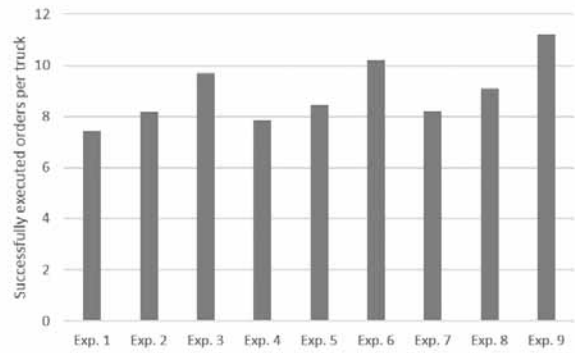


Figure 1: Successfully executed orders per truck and day

are therefore, chosen first. If no time windows in peak hours are left, the time windows in off-peak hours are booked. This leads to shorter waiting times at the terminals due to the lower assumed handling times as well as to less congestion in the port and thereby, to lower transport times. Furthermore, transports in the peak times tend to be more risky due to a higher variance in transport and handling times. Therefore, more transports in peak hours need to be canceled due to predicted late arrivals at their destination. This reduces the productivity of the trucking companies considerably.

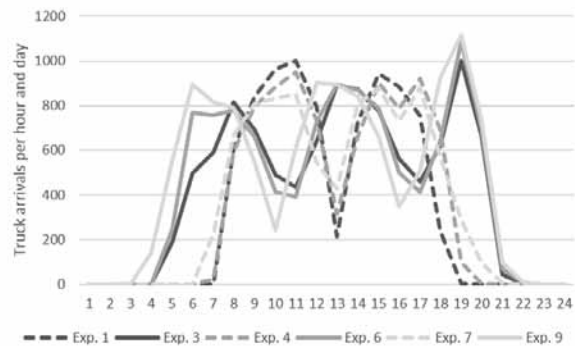


Figure 2: Truck arrival times at all types of logistics nodes for Exp. 1,3,4,6,7 and 9

A similar effect can be seen in Figure 2. There, the arrivals at logistics nodes are spread out in a wider time range for Exp. 3, 6 and 9 compared to Exp. 1, 4 and 7. Considering the lower transport and handling times at the off-peak hours, this leads to a higher number of executed transports and therefore, to a higher productivity for the trucking companies. The transports executed in the off-peak hours are mainly inter-terminal transports,

as the other logistics nodes tend to have limited working hours. This fact can also be seen in Figure 3, where the truck arrival times at container terminals and other logistics nodes are shown exemplarily for Exp. 1 and 9. It is evident that the inter-terminal transports in Exp. 9 are further shifted to off-peak hours. A high percentage of the transports in peak hours is executed between container terminals and other logistics nodes.

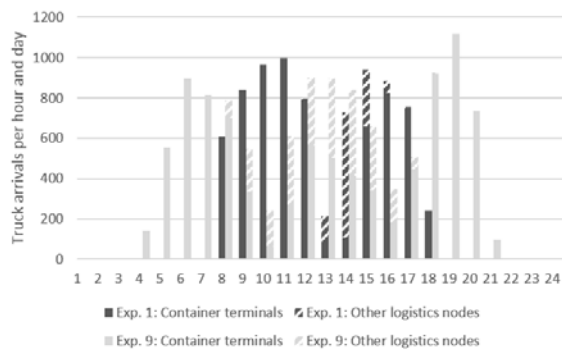


Figure 3: Detailed truck arrival times for Exp. 1 and 9

6 Conclusion and Outlook

In conclusion, a TAS offers chances for trucking companies as well as for logistics nodes. Especially, if a trucking company executes many inter-terminal transports it is flexible enough to adapt to a TAS and seize the opportunities. It is expected that the results would change considerably if a higher amount of non-inter-terminal transports is assumed. In this case, the restrictions imposed by TAS and limited opening hours of other logistics nodes would very likely reduce trucking companies' productivity when the available capacity is limited. In future, more TAS' characteristics should be analyzed on their impact on the different stakeholders in drayage networks. Furthermore, a broader variance in booking strategies for the trucking companies as well as operations at logistics nodes should be considered.

References

- [1] United Nations Conference on Trade and Development. *Review of maritime transport 2017*. New York, Geneva: United Nations; 2017
- [2] Hartmann S. Generating scenarios for simulation and optimization of container terminal logistics. *OR Spectrum*. 2004; 26, 171–192.
- [3] Huynh N, Harder F, Smith D, Sharif O, Pham Q. Truck Delays at Seaports. *Transportation Research Record: Journal of the Transportation Research Board*. 2011; 2222, 54–62.
- [4] Harrison R, Hutson N, West J, Wilke J. Characteristics of Drayage Operations at the Port of Houston, Texas. *Transportation Research Record: Journal of the Transportation Research Board*. 2007; 2033: 31–37.
- [5] Shiri S, Huynh N. Optimization of drayage operations with time-window constraints. *International Journal of Production Economics*. 2016; 176: 7–20.
- [6] Giuliano G, O'Brien T. Reducing port-related truck emissions. The terminal gate appointment system at the Ports of Los Angeles and Long Beach. *Transportation Research Part D: Transport and Environment*. 2007; 12: 460–473.
- [7] Bentolila DJ, Ziedeneber RK, Hayuth Y, Notteboom T. Off-peak truck deliveries at container terminals: the "Good Night" program in Israel. *Maritime Business Review*. 2016; 1/1(1):2–20.
- [8] Davies P, Kieran M.E. Port Congestion and Drayage Efficiency. *METRANS International Urban Freight Conference*. 2015 Oct; Long Beach, USA.
- [9] Huynh N, Smith D, Harder F. Truck Appointment Systems. *Transportation Research Record: Journal of the Transportation Research Board*. 2016; 2548: 1–9.
- [10] Chen G, Govindan K, Yang Z-Z, Choi T-M, Jiang L. Terminal appointment system design by non-stationary M(t)/Ek/c(t) queueing model and genetic algorithm. *International Journal of Production Economics*. 2013; 146: 694–703.
- [11] Zhao W, Goodchild AV. The impact of truck arrival information on container terminal rehandling. *Transportation Research Part E: Logistics and Transportation Review*. 2010; 46: 327–343.
- [12] Heilig L, Lalla-Ruiz E, Voß S. port-IO. An integrative mobile cloud platform for real-time inter-terminal truck routing optimization. *Flex Serv Manuf J*. 2017; 29: 504–534.
- [13] Namboothiri R, Erera AL. Planning local container drayage operations given a port access appointment system. *Transportation Research Part E: Logistics and Transportation Review*. 2008; 44: 185–202.
- [14] Lange A-K, Schwientek A, Jahn C. Reducing truck congestion at ports – classification and trends. In: Jahn C, Kersten W, Ringle C M, editors. Digitalization in maritime and sustainable logistics. City logistics, port logistics and sustainable supply chain management in the digital age. *Hamburg International Conference of Logistics*. 2017 Sep; Hamburg, Germany: epubli. 37–58.

Praxisbeispiel: Simulationsmodell eines Shuttle-Systems zur Ermittlung des Systemverhaltens

Anja Langanki¹, Thomas Kriehn^{1*}, Franziska Schloz², Karl-Heinz Wehking², Stefan Kuhlins¹, Markus Fittinghoff¹

¹Fakultät für Technische Prozesse, Hochschule Heilbronn, Max-Planck-Str. 39, 74081 Heilbronn, Deutschland;
*thomas.kriehn@hs-heilbronn.de

²Institut für Fördertechnik und Logistik, Universität Stuttgart, Holzgartenstraße 15B, 70174 Stuttgart

Abstract. Beschrieben wird ein Simulationsmodell eines ungebundenen Shuttle-Systems mit Fahrzeughlifen, das mit der Simulationssoftware AutoMod erstellt wurde. Das Shuttle-System wird in einem Unternehmen eingesetzt (der Name darf aus Datenschutzgründen nicht genannt werden). Der Aufbau und die Funktionsweise des Simulationsmodells werden erklärt. Weiterhin werden die Nutzungsmöglichkeiten des Simulationsmodells geschildert und Ergebnisse gezeigt.

Einleitung

In diesem Beitrag wird ein Simulationsmodell eines ungebundenen Shuttle-Systems vorgestellt. Ungebundene Shuttle-Systeme ermöglichen den Shuttle-Fahrzeugen jeden Lagerplatz im Lager zu erreichen. Das Simulationswerkzeug ist die Software AutoMod (Version 12.6.1). AutoMod eignet sich dazu, intralogistische Systeme abzubilden und zu untersuchen. Der Simulator ist nach Bausteinen ausgerichtet. Das bedeutet, dass zur Modellierung ein Baukasten zur Verfügung steht, welcher verschiedene Materialflusselemente umfasst. Er enthält sowohl statische als auch dynamische Komponenten, die jeweils über verschiedene Einstellmöglichkeiten verfügen [4]. Die Simulationsmodelle werden dreidimensional visualisiert.

Simulationsuntersuchungen zu ungebundenen Shuttle-Systemen waren bereits Gegenstand einiger Veröffentlichungen, z. B. [1, 2, 3]. In [1] wurden die Anzahl der Lifte und Shuttle-Fahrzeuge sowie Auftragsreihenfolgen variiert und über Versuchspläne optimale Kombinationen ermittelt. In [2] wurde der Einfluss von vertikalen Zonierungen auf den Durchsatz untersucht. Dabei durften Shuttle-Fahrzeuge nur innerhalb bestimmter Ebenen wechseln. In [3] wurde die Anzahl der Gänge, Ebenen und Lagerplätze variiert. Anhand der Ergebnisse der Simulationsläufe wurde eine Regressionsanalyse durchgeführt, um ein analytisches Modell zur Beschreibung der Spiel- und Wartezeiten sowie des Auslastungsgrades der Fördermittel und der Auftragsdurchlaufzeit zu erhalten.

Im Unterschied zu den genannten Veröffentlichungen

werden bei dem hier vorgestellten Simulationsmodell unterschiedlich dimensionierte Ladeeinheiten an dafür vorgesehene Lagerplätze eingelagert. Weiterhin werden vertikale und horizontale Zonierungen untersucht, die Shuttle-Fahrzeuge werden aber nicht an unterschiedliche Ebenen gebunden (im Vergleich zu [2]), sondern können alle in dieselben Ebenen wechseln. Wenn Ebenen oder Gänge aufgrund einer Zonierung nicht angesteuert werden, dann gilt das immer für alle Shuttle-Fahrzeuge im System.

Die Durchführung einer Simulation mittels geeigneter Software dient z. B. zur Leistungsverbesserung eines geplanten oder bereits bestehenden Lagersystems. Sie wird auch dazu genutzt das Systemverständnis, die Lösung oder den Anlagenbetrieb zu verbessern. Ebenso wird die Sicherheit vergrößert, indem Fehlplanungen vermieden werden. Die Voraussetzung für die Durchführung einer Simulationsuntersuchung kann erfüllt sein, wenn beispielsweise Tests am realen System zu kostenintensiv sind, analytische Berechnungen nicht mehr ausreichen, Erfahrungswerte bei einem realen System fehlen oder das Systemverständnis durch komplizierte Wirkungszusammenhänge schwer erfassbar ist [5].

1 Aufbau und Funktionsweise des Simulationsmodells

Der erste Schritt beim Aufbau des Simulationsmodells ist die maßstabgetreue Abbildung des realen Systems [4]. Daraufhin werden die Abläufe innerhalb des Systems implementiert. Diese lassen nach Fertigstellung ein funktionierendes Simulationsmodell entstehen [6].

Da AutoMod nicht über vorkonfigurierbare Shuttle-System-Komponenten verfügt, um die gegebenen Lagereigenschaften nachzubilden, wurde das Modul „Path Mover“ für das Erstellen des Modells verwendet. Ein Path Mover ist ein Systemelement für pfadgebundene Fahrzeuge [7]. Abbildung 1 zeigt das Simulationsmodell des Shuttle-Systems. Zu sehen sind zwei Lifte (rot) auf

dem vertikal verlaufenden Path Mover. Shuttle-Fahrzeuge fördern Ladeeinheiten entlang der horizontalen Pfade zu einem Lagerplatz. Im Hintergrund sind Ladeeinheiten (grün und gelb) zu sehen, die sich nach einem bereits durchgeführten Einlagerungsprozess auf dem Lagerplatz befinden.

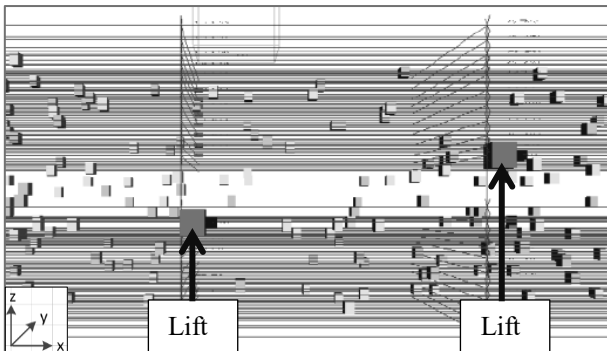


Abbildung 1: Simulationsmodell des Shuttle-Systems

Das im Simulationsmodell dargestellte Shuttle-System erstreckt sich über zwei Etagen, jede Etage besitzt elf Ebenen. Eine Ebene hat jeweils 1.680 Lagerplätze, daraus resultiert eine Gesamtanzahl von 36.960 Lagerplätzen. Das Shuttle-System lagert zwei unterschiedlich dimensionierte Ladeeinheiten in jeweils passende Lagerplätze ein. Die größer dimensionierte Ladeeinheit kann in Ebenen mit hohen Lagerplätzen einfach tief eingelagert werden. Die kleiner dimensionierte Ladeeinheit kann in Ebenen mit niedrigen Lagerplätzen doppelt tief eingelagert werden.

Die Einlagerung wird chaotisch durchgeführt, es erfolgt demnach keine bestimmte Lagerplatzzuteilung. Es wird lediglich ein freier, von der Höhengeometrie passender Lagerplatz ausgewählt. Über eine Zufallsvariable wird vom System entweder eine kleiner oder größer dimensionierte Ladeeinheit generiert. Die Wahrscheinlichkeit des Auftretens der jeweiligen Ladeeinheit ist anhand der realen Unternehmensdaten des Auswertungszeitraums eingestellt. Die Ladeeinheiten wurden im Simulationsmodell jeweils mit unterschiedlichen Farben kenntlich gemacht. Die Auslagerung der Ladeeinheiten erfolgt ebenfalls chaotisch.

In Abbildung 2 ist die vereinfachte Draufsicht des Hochregallagers dargestellt. Das Shuttle-System besitzt drei Gänge (y-Richtung) und sieben Quergänge (x-Richtung), mit Schienen für die Shuttle-Fahrzeuge, welche den Gangwechsel ermöglichen. Das Shuttle-System besitzt insgesamt drei Fahrzeuglifte, die die Vertikalförderung der Shuttle-Fahrzeuge, gegebenenfalls mit Ladeeinheiten, übernehmen. In diesem System werden mit ihnen die Ladeeinheiten sowohl ein- als auch ausgelagert. Die Lifte führen, wenn möglich, Doppelspiele durch. Sie können jede Ebene des Lagers anfahren. Die Anfahrwahrscheinlichkeiten der Ebenen sind hierbei gleich groß. Im Shuttle-System werden

neun Shuttle-Fahrzeuge eingesetzt. Jedes Shuttle-Fahrzeug kann auf jeden Lagerplatz und jeden freien Lift im System zugreifen. Das System ist somit sehr flexibel und ermöglicht einen bedarfsorientierten Einsatz der Shuttle-Fahrzeuge. Blockadeeffekte durch mehrere Shuttle-Fahrzeuge innerhalb einer Ebene werden im Simulationsmodell vernachlässigt.

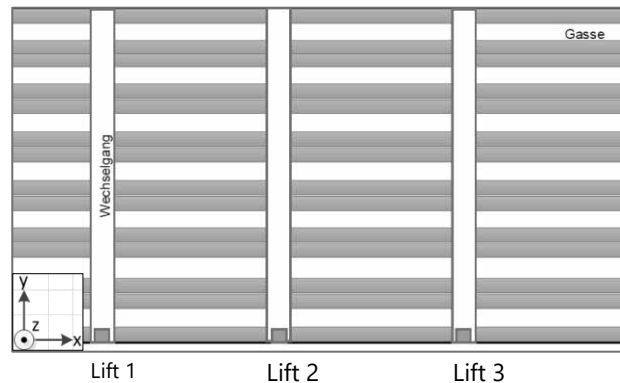


Abbildung 2: Vereinfachte Lager-Draufsicht (Ausschnitt)

In den Abbildungen 3 und 4 sind die Prozessabläufe innerhalb des Simulationsmodells dargestellt. Dabei ist zu beachten, dass Shuttle- und Liftprozesse parallel zueinander ablaufen. Zu Beginn beider Prozesse findet ein Initialisierungsprozess statt, der dazu dient die Shuttle-Fahrzeuge an die richtige Startposition zu setzen. Im Shuttleprozess wird ein freier Lift ausgewählt: Der entsprechende Lift und das Shuttle-Fahrzeug, das eine Förderung benötigt, fahren zum Shuttleaufnahme-punkt. Dort nimmt der Lift das Shuttle-Fahrzeug auf und fährt mit diesem zum Ein-/Auslagerungs-Punkt (E/A-Punkt). Am E/A-Punkt angekommen, nimmt das Shuttle-Fahrzeug eine Ladeeinheit auf. Ist dieser Vorgang abgeschlossen, wird vom Liftprozess eine Einlagerebene ausgewählt. Es wird geprüft, ob sich in der jeweiligen Ebene freie, von der Höhengeometrie passende Lagerplätze befinden. Ist dies der Fall, fördert der Lift das Shuttle-Fahrzeug in die ausgewählte Einlagerebene. Wurde in der Ebene kein geeigneter Lagerplatz gefunden, wird solange weitergesucht, bis in einer Ebene ein Lagerplatz gefunden wird. In der Einlagerebene angekommen, gibt der Lift das Shuttle-Fahrzeug ab. Der Lift ist anschließend bereit für die Aufnahme eines neuen Shuttle-Fahrzeugs. Hierzu kann der Lift je nach Bedarf auf der gleichen Ebene bleiben oder einen Ebenenwechsel durchführen. Unterdessen beginnt im Shuttleprozess die Initialisierung für die Einlagerung. Ein freier Lagerplatz wird ausgewählt und im anschließenden Prozess vom Shuttle-Fahrzeug angefahren. Bei Ankunft am Lagerplatz wird die Ladeeinheit auf den Lagerplatz befördert. Im System wurde ein Zähler eingebaut, der jede Einlagerung erfasst. Bei Erreichung eines bestimmten Wertes wird die Initialisierung für die Auslagerung durchgeführt. Ein belegter Lagerplatz wird ausgewählt

und im anschließenden Prozess angefahren. Das Shuttle-Fahrzeug entnimmt beim Auslagerungsvorgang die entsprechende Lagereinheit. Erneut wird ein freier Lift ausgewählt und der Shuttle-Aufnahmepunkt angefahren.

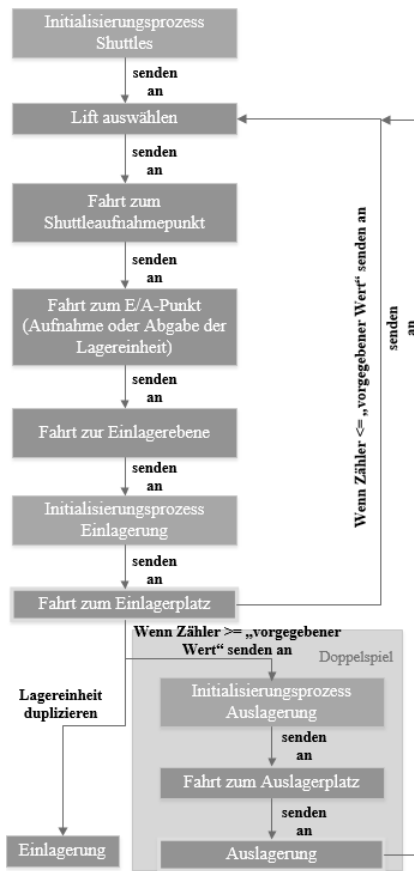


Abbildung 3: Shuttle-Prozesse

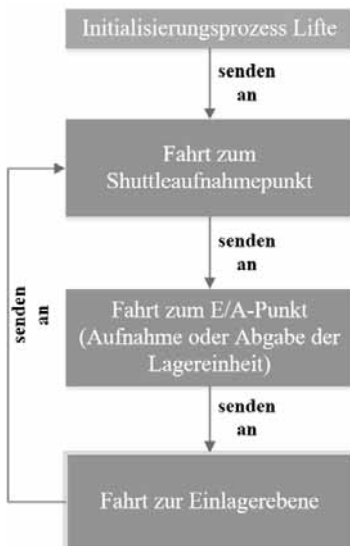


Abbildung 4: Lift-Prozesse

2 Nutzungsmöglichkeiten des Simulationsmodells

Das Simulationsmodell kann verändert werden, indem beispielsweise eine neue Lagerstrategie einprogrammiert oder die Anzahl der Shuttle-Fahrzeuge gesenkt oder erhöht wird. Nach jeder Veränderung am Simulationsmodell können neue Werte gemessen werden, um diese mit den vorherigen Werten zu vergleichen. So kann festgestellt werden, ob eine Verbesserung oder Verschlechterung auf Basis eines gewählten Kriteriums, z. B. der Durchsatzmaximierung, eingetreten ist.

Anhand von verschiedenen Variablen ist es in diesem Simulationsmodell möglich, Messungen durchzuführen. Es können Kennzahlen wie Auslastungsgrad, Spielzeit, Durchsatz und Wartezeiten der Lifte und Shuttle-Fahrzeuge ermittelt werden.

Der Lagerfüllgrad kann bei der Auswahl einer geeigneten Lagerstrategie eine Rolle spielen. Es kann z. B. in Abhängigkeit des Lagerfüllgrads eine Lagerstrategie angewendet werden, bei der freie Lagerplätze ausgeschlossen werden, die sich in den hinteren Bereichen des Shuttle-Systems befinden. Dadurch wird die Fahrtzeit der Shuttle-Fahrzeuge reduziert. Auch die Lift-Fahrtzeiten können reduziert werden, wenn bestimmte Ebenen nicht oder mit geringerer Wahrscheinlichkeit angesteuert werden. Das kann auch durch Bildung geeigneter Zonen in Abhängigkeit der Häufigkeit der Auftragseingänge für die Ladeeinheiten erfolgen.

Wird bei einem veränderten Simulationsmodell nach Durchführung der Simulationsläufe eine Verkürzung der Spielzeit und/oder Wartezeit festgestellt, so können mehr Spiele pro Zeiteinheit durchgeführt werden. Das führt zu einem verbesserten Durchsatz, welcher das Leistungsmaß aller automatischen Lagersysteme ist.

Durch das Simulationsmodell konnten die genannten Kennzahlen für das Unternehmen ermittelt und ein Einblick in die Abläufe des Shuttle-Systems gewährt werden. Es konnte u. a. gezeigt werden, inwieweit die Variation der Anzahl der Shuttle-Fahrzeuge Einfluss auf das Systemverhalten hat: Bei Einsatz von neun Shuttle-Fahrzeugen wird im Vergleich zum Einsatz eines Shuttle-Fahrzeugs eine Durchsatzsteigerung von 515 Prozent erzielt. Abbildung 5 zeigt die Durchsatzsteigerungen bei der Variation der Shuttle-Fahrzeuge von eins bis neun, alle Prozentwerte beziehen sich auf die Durchsatzsteigerung im Vergleich zu einem Shuttle-Fahrzeug. Im gemessenen Szenario führten die Lifte und die Shuttle-Fahrzeuge Doppelspiele aus. Erkennbar ist, dass mit jedem weiteren Shuttle-Fahrzeug die Steigerung des Durchsatzes geringer ausfällt, siehe Trendlinie in Abbildung 5. Mehr als neun Shuttle-Fahrzeuge erbringen keine signifikante Durchsatzsteigerung mehr, da die Lifte vollständig ausgelastet sind. Bereits ab sieben Shuttle-Fahrzeugen wird der Durchsatz nur noch gering-

fällig gesteigert. Das Unternehmen hält derzeit neun Shuttle-Fahrzeuge vor. Um weitere Shuttle-Fahrzeuge im System nutzen zu können, müsste ein weiterer Lift installiert oder die Lagerstrategie optimiert werden.

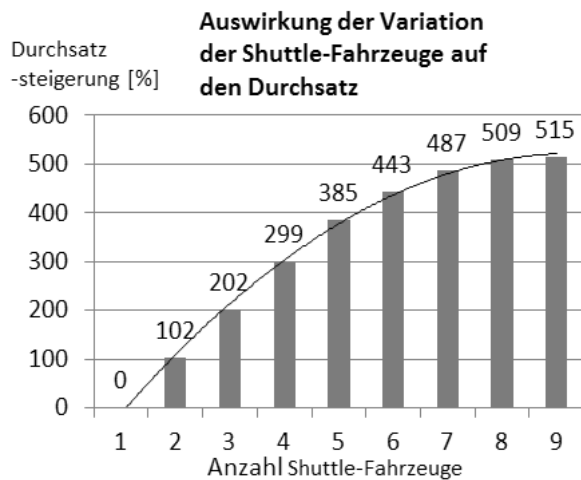


Abbildung 5: Durchsatzsteigerung durch Erhöhung der Shuttle-Fahrzeuge

Das Optimierungspotential bzgl. angepasster Lagerstrategien konnte gezeigt werden, indem die Fahrwege der Lifte und Shuttle-Fahrzeuge reduziert wurden. Alle folgend genannten Ergebnisse wurden mit neun Shuttle-Fahrzeugen ermittelt. Auf Grundlage des Lagerfüllgrads von 75 Prozent wurde das Anfahren der oberen Ebenen (Ebene 18, 19, 20, 21 und 22) unterbunden. Das führte zu einer Erhöhung des Durchsatzes um 3 Prozent.

Die Reduzierung des hinteren Quergangs in jeder Ebene (siehe Abbildung 2) erhöhte den Durchsatz durch die kürzeren Wege der Shuttle-Fahrzeuge um 1 Prozent. Das bedeutet, die alleinige Reduzierung der Fahrtwege auf Basis des Lagerfüllgrades erbringt nur eine geringe Durchsatzsteigerung.

Weiteres Optimierungspotential birgt eine Zonierung nach der Zugriffshäufigkeit der Artikel. Es wurden zwei Zonen gebildet, die häufig zugegriffenen Artikel wurden in den Ebenen 1 – 8, im Bereich der vorderen Hälfte der Lagerplätze eingelagert (80 Prozent Zugriffshäufigkeit). Die selten zugegriffenen Artikel wurden in den Ebenen 1 – 8 im Bereich der hinteren Hälfte der Lagerplätze und in den Ebenen 9 – 17 eingelagert (20 Prozent Zugriffshäufigkeit). Damit konnte der Durchsatz im Vergleich zur chaotischen Einlagerung um 8 Prozent gesteigert werden.

3 Zusammenfassung und Ausblick

Durch das Erstellen des Simulationsmodells und die Durchführung der Simulationsläufe können dem Unternehmen Daten geliefert werden, die bisher nicht bekannt waren und zur Verbesserung des realen Shuttle-Systems

dienlich sind.

Die Vorgänge innerhalb des Shuttle-Systems wurden visualisiert und ermöglichen so ein verbessertes Systemverständnis für den Betrachter. Es konnte ein Einblick in die Leistung und die Abläufe des Shuttle-Systems gewährt werden.

An diesem Simulationsmodell können unterschiedliche Lagerstrategien untersucht werden, hierzu ist nur der Programmiercode anzupassen.

Da Blockadeeffekte der Shuttle-Fahrzeuge innerhalb der Ebenen in diesem Simulationsmodell vernachlässigt wurden, könnten diese in Zukunft noch detailliert untersucht und Lösungen hierzu implementiert werden.

References

- [1] B. Y. Ekren, S. Heragu, A. Krishnamurthy und C. J. Malmberg, Simulation based experimental design to identify factors affecting performance of AVS/RS, Bd. 58, Computers & Industrial Engineering, 2010, pp. 175 - 185.
- [2] A. Kumar, D. Roy und M. K. Tiwan, Optimal partitioning of vertical zones in vehicle-based warehouse systems, Bd. 52, International Journal of Production Research, 2014, pp. 1285 - 1305.
- [3] B. Y. Ekren und S. Heragu, Simulation-based regression analysis for the rack configuration of an autonomous vehicle storage and retrieval system, Bd. 48, International Journal of Production Research, 2010, pp. 6257 - 6274.
- [4] Dieter Arnold und Kai Furmans, Materialfluss in Logistiksystemen VDI-Buch. Springer-Verlag. Berlin, Heidelberg 2006.
- [5] Mathias Schumann, Zur Bestimmung der Umschlagleistung von Hochregallagern unter besonderer Berücksichtigung der Lagerorganisation (Dissertation). Technische Universität Chemnitz, 2007.
- [6] Kudlich, T., Optimierung von Materialflußsystemen mit Hilfe der Ablaufsimulation (Dissertation), Technische Universität München, 2000
- [7] www.automod.de, abgerufen am 03.05.2018

Strategies to Mitigate the Impacts of Climate Change Related Events on Public Transit Networks

Daniel Lückerath, Manfred Bogen, Erich Rome,
Oliver Ullrich, Rainer Worst, Jingquan Xie

Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme, Schloss Birlinghoven, 53754 Sankt Augustin, Germany;
[first.last]@iais.fraunhofer.de

Abstract. Public transit networks are a critical component of urban infrastructure threatened by climate change and related extreme weather events. These events include sudden, heavy precipitation, pluvial and fluvial flooding, and landslides, which can significantly affect network components like stops, switches, or track segments. The ongoing trend towards urbanization will raise the already high demand for public transportation even more, which makes it of paramount importance that public transport providers and local authorities develop proactive crisis management strategies that include the impacts of extreme weather events. This paper describes a number of rule-based, short-term mitigation strategies whose application helps to address those impacts. The strategies are designed in such a fashion that they can easily be understood and applied by practitioners and be implemented in a decision support system. The effectiveness of the proposed strategies is shown exemplarily via simulation experiments on the model of an artificial transit network using a prototypical implementation based on a mesoscopic simulation model for multi-modal public transit networks.

Introduction

Public transit networks are a critical component of urban infrastructure systems, which are increasingly threatened by the impacts of climate change (see [4]). It is not yet clear how much damage climate change-related extreme events will cause to European transportation sectors in the future (see [21]). For the road network, on average 30% to 50% of current road maintenance cost is already seen as consequences of extreme events (see [16]). In Europe, examples for these events include coastal, fluvial, and pluvial flooding, flash floods caused by heavy precipitation, rockslides and landslides, temperature extremes, thunderstorms and tornados, winter storms, and rising sea levels (see [21]). Unfortunately, systemic risks to public transit networks caused by climate change are not well understood across modes, regions, and critical infrastructure sections; a high uncertainty over likelihood, costs, and impact of emerging risks still exists. As

a consequence, there is a lack of resilience schemes integrated in transport infrastructure and operational management; this is reinforced due to the inability to monetize resilience. Given the trend towards increased urbanization in Europe, with over 80% of Europeans expected to live in urban population centers in 2050 (see [27]), it is increasingly urgent to prepare for climate change-related events by finding and systematically evaluating methods to address the related impacts on critical infrastructures.

Simulation technology can be employed to achieve this goal by helping to develop targeted mitigation strategies for transport networks when combined with vulnerability assessment methods like flood risk analysis. The latter helps identifying parts of the network that are endangered by flooding, and the simulation can then be applied for finding and evaluating solutions for minimizing service outages and estimating secondary impacts to the transport system, like delays and traffic jams resulting from rerouting of busses due to a flooded intersection.

As a first step, this paper discusses a number of rule-based, short-term strategies to address the impacts climate change-related extreme weather events can have on multi-modal public transit networks, developed in the context of the ongoing EU project “Climate Resilient Cities and Infrastructures – RESIN” (see [20] and [28]). These events not only have to be addressed with long-term measures as part of urban planning and development strategies, but also with short-term relief strategies to be activated at the occurrence of extreme events. To ensure applicability and viability of these strategies, they have to be tested in simulated scenarios. Therefore, the paper includes an example application of the proposed strategies based on a model of an artificial transit network simulated using a prototypical implementation in a mesoscopic simulation model for multi-modal public transit systems (see [12] and [13]).

The paper continues with a short introduction of existing disturbance mitigation approaches in public transit networks and simulation models able to evaluate them (see Section 1). Afterwards, project RESIN is briefly introduced (see Section 2), followed by a description of a number of the rule-based strategies developed to increase the resilience of public transit networks against climate change-related events (see Section 3). The paper goes on to describe a set of experiments conducted on a small model of an artificial transit network (see Section 4), designed to illustrate the impact of these strategies. It concludes with a short description of the lessons learned and an outlook on the next research steps (see Section 5).

1 Background

A number of approaches to mitigate disturbances in multi-modal public transit networks are known from literature: Borne et al. (see [1]) present a simulation-based approach that combines an agent-based model with a Genetic Algorithm (see [6]) to reduce waiting, travel, and transfer times for passengers by modifying stopping and driving times as well as vehicle routes. Zolfaghari et al. (see [30]) describe a headway-based¹ Simulated Annealing (see [6]) approach, which takes real-time information about vehicle position and capacity into account, to determine optimum stopping times to minimize the combined waiting time of all passengers of a given route. A similar method is presented by Yu and Yang (see [29]), who employ a timetable-based² Genetic Algorithm approach to minimize passenger waiting times, while also considering a vehicle's punctuality at stops further along the route. Ginkel and Schöbel (see [7]) place an emphasis on transfer connections, and formulate a bi-criteria integer linear model based on an event-activity network, reducing both vehicle delays and the number of missed transfer connections. Corman et al. (see [5]) describe the decision support system ROMA (Railway traffic Optimization by Means of Alternative graphs) aimed at evaluating different rerouting options to mitigate a disturbance. These options are not automatically built-up by the system, but result from input gained in interviews with local experts. For every route option the system calculates optimum timetables and allows their evaluation based on

passenger travel or waiting time as well as vehicle delay. Hadas and Ceder (see [8]) use a dynamic programming approach able to consider different rerouting and rescheduling approaches simultaneously, whose applicability they demonstrate with a small artificial network based on four (unidirectional) transit routes. In a follow-up paper, Nesheli and Ceder (see [17]) further develop their approach and validate it based on a partial bus network of Auckland, New Zealand. Most of the methods described until now are optimization-based approaches applying a rolling time horizon. As Kliewer and Suhl (see [11]) demonstrate, this might lead to disadvantages: a measure might lead to an optimum solution for the given – typically narrow – time window, but might yield worse results later on during an operational day. Kliewer and Suhl strengthen their argument via simulation experiments based on a model of the German railway network around Frankfurt and show that online optimization models with rolling time horizon often are not significantly better than simple rule-based models, if the whole operational day is taken into account.

Following the argument from Kliewer and Suhl, the mitigation strategies presented in the remainder of this paper focus on rule-based approaches, which are easily understandable and applicable by practitioners.

To evaluate rerouting and rescheduling approaches, simulation models are applied. These models are usually multi-modal to represent significant parts of the transportation network of an urban area. They often include representations of individual travelers, but usually simplify certain aspects of a transportation system. Nagel (see [15]) describes the microscopic simulation system TRANSIMS, which, comparable to Meignan et al (see [14]), constructs its network model out of sub-networks for light rail, car, and pedestrian traffic. Transfer points are defined to enable the transfer from one sub-network to another. Kendziorra and Weber (see [10]) describe a multi-modal extension for the well-known SUMO open-source framework (Simulation of Urban Mobility, see [23]), while Cats (see [2]) presents an extension to the BusMezzo model (see [26]). Both do not split the network between different modes, but develop one common network model that includes all modes. However, Cats does not represent pedestrian trips. Suzumura and

originally scheduled departure times as much as possible.

¹ Headway-based rescheduling approaches aim to keep to the originally scheduled headways, i.e. the time intervals between successive vehicles, as much as possible.

² Timetable-based rescheduling approaches aim to keep to the

Kanezashi (see [24]) and Suzumura et al. (see [25]) present a very ambitious multi-modal, parallel, agent-based model based on the IBM Mega Traffic Simulator Megaffic (see [18]). Their model represents all traffic participants as agents, and includes pedestrians, bicyclists, cars, buses, as well as light and heavy rail traffic.

All these models are designed to gather general data on the overall performance of a transit network and not to systematically check mitigation strategies and their impact on specific stops, routes, and vehicles. The multi-modal public transit model described by Lückerrath et al. (see [12] and [13]) is explicitly designed for that task. It will therefore be applied in the remainder of this paper.

2 The Project RESIN

The EU H2020 project “Climate Resilient Cities and Infrastructures – RESIN” (see [20] and [28]) investigates climate change adaptation practices in European cities in order to develop methods and decision support tools that decision makers can use to develop local adaptation strategies. The project builds on previous research by combining existing approaches to climate change adaptation, protection of critical infrastructure, and disaster risk management, while taking into account all of the core elements of the urban system and their interrelations. One of the central aims of RESIN is to provide methods and tools for comprehensive risk-oriented vulnerability assessments of an urban population center and its critical infrastructure. These efforts are based on the latest state of discussion and conceptual approaches of the Intergovernmental Panel on Climate Change Assessment Report 5 (IPCC AR5, see [9]).

As part of the vulnerability assessment primary and secondary impacts to critical infrastructures under investigation have to be estimated (e.g. traffic disruptions), taking into account potential mitigation strategies employed by local authorities. Since historical records about the impact of extreme events are limited and usually do not take current state-of-the-art mitigation strategies into account, simulation experiment or expert judgement have to be employed.

All RESIN methods and tools are being developed by means of co-creation with the cities of Bilbao (Spain),

Greater Manchester (United Kingdom), Paris (France), and Bratislava (Slovakia) to ensure their applicability for local practitioners. At the same time, processes, methods, and tools are standardized and modular, so that they can be applied to more European urban population centers and can be adapted to the local context.

RESIN is only one of several interdisciplinary, practice-based research projects investigating climate resilience in European cities. The EU project “Reconciling Adaptation, Mitigation and Sustainable development for cities – RAMSES” (see [19]) developed methods and tools to quantify evidence of the impacts of climate change and the costs and benefits of adaptation measures to cities. The EU project “Smart Mature Resilience – SMR” (see [22]) aims at developing a resilience management guideline to support city decision-makers in developing and implementing resilience measures.

3 Rule-Based Strategies to Mitigate the Impacts of Climate Change Related Events

The developed short-term operational strategies to mitigate disturbances in public transit networks caused by extreme events can be divided into two groups: delay management and disruption management. The delay management strategies cover timetable- and rule-based holding strategies to increase the punctuality of individual vehicles and reduce vehicle bunching³ (see [3]), as well as strategies to dynamically modify the routes of delayed vehicles in a limited way (e.g. via short-turning or dead-heading). The disruption management strategies cover methods for more comprehensive changes of line routes and schedules of multiple vehicles in case of larger disruptions, for example the temporary or permanent blockade of stops, or track segments.

While the second group of strategies is aimed at primary impacts of climate change-related extreme weather events, the first group can be used to limit the secondary impacts, e.g. by keeping vehicles punctual once a new route and/or timetable have been established.

as originally planned. Thus, without intervention the preceding vehicle gets further delayed while the succeeding vehicle catches up to it.

³ Vehicle bunching describes the phenomenon of public transit vehicles to form pairs due to a preceding delayed vehicle taking on more passengers as originally planned and the on-time succeeding vehicle subsequently taking on fewer passengers

3.1 Delay Management Strategies

The *basic bunching mitigation strategy* compares a too early vehicle's deviation from its planned departure time at its current stop to a user-defined threshold to avoid vehicle bunching further down the route. If the deviation is greater than the threshold, the vehicle is held back. In addition to considering the departure deviation, a more complex variant of the strategy also considers a vehicle's estimated punctuality at its next scheduled stop (see [29]). In case the estimate shows that the vehicle would be late at the succeeding stop even if it would depart immediately from the current stop, the vehicle will not be delayed any further. In case a vehicle would arrive at the succeeding stop pre-schedule, it is delayed so ensure a punctual arrival. This might lead to the threshold at the current stop being ignored, if such action would avoid the vehicle to be delayed at the next stop.

The *synchronization assurance strategy* follows the methods to maintain transfer options proposed in [8] and [17]. These transfers are classified either as rendezvous connections or as directed transfers. Rendezvous transfers define a specific time window during which a number of vehicles serving different lines are located at the same station, so that passengers can transfer from any one line to any other. Here, all participating vehicles wait until the last one has arrived at the current station plus a given time duration to enable passenger exchange or until a specified maximum waiting time has elapsed. This might result in vehicles getting delayed, if it ensures that the rendezvous connection is hold. A directed transfer connection is a special case of rendezvous connection, where a vehicle serving a first line waits for a feeder vehicle serving a second line at a given station, and thus enables transfer from the second line to the first.

3.2 Disruption Management Strategies

The designed disruption management strategies cover four different areas: rerouting, short-turning, stop skipping, and route separation. The resulting strategies are hierarchically organized in an overall disruption management strategy.

After the occurrence of a disruption in a section of the transit network, first rerouting of vehicles is considered: sets of potential alternative routes for all lines affected by the disruption are constructed using a pathfinding algorithm, e.g. breadth-first-search. The potential alternative routes do not have to be direct connections between a ve-

hicle's current and succeeding stop and may include additional stops, but have to cover all stops of the regular route. Once all potential alternative routes are determined, the method picks – depending on the operator's preferences – either the route with the least traversal time or the most punctual route. If two or more routes are equally suitable, the route that deviates least from the regular route is selected.

If no alternative route covering all regular stops can be found, short-turning, i.e. ending the trip before reaching the disrupted network section, is considered. Trips are only short-turned, if the executing vehicles are significantly delayed, the vehicles have the option to change direction prior to the disrupted network section, and the vehicles can still service a relevant portion of the regular route. What exactly constitutes a 'relevant portion' has to be defined by the transit network operator beforehand.

In case the method decides to short-turn a trip, that action has an impact on the executing vehicle's next trip: Either the vehicle has to make a deadheading trip to the last stop of its current route, or the next trip of the vehicle has to be short-turned as well to start from the vehicle's current position. The former might be employed under less severe circumstances, where the disrupted network section can still be accessed and the vehicle's delay can be removed or at least substantially reduced by not executing regular boarding processes along the remainder of its regular route. An advantage of deadheading is that the vehicle might be able to start its next scheduled trip from the originally scheduled stop. However, deadheading might be complicated to explain to stranded passengers.

A potential disadvantage of the described short-turning method is that it does not guarantee the reachability of all stops further down the route. A more general method would guarantee that at least one alternative path leads from the station at which a trip is short-turned to each station further down the regular route.

In case short-turning is not possible or not worthwhile, the method tries to skip the disrupted network section by rerouting the affected vehicles in a similar fashion than in the first step, but this time also allowing alternative routes that do not contain (all) stops of the regular route.

If all these measures fail, the method will, where possible, separate all affected routes before/after the disrupted network section. In general, this can be viewed as a two-sided short-turning strategy, where vehicles stuck on either side of the disrupted network section service as

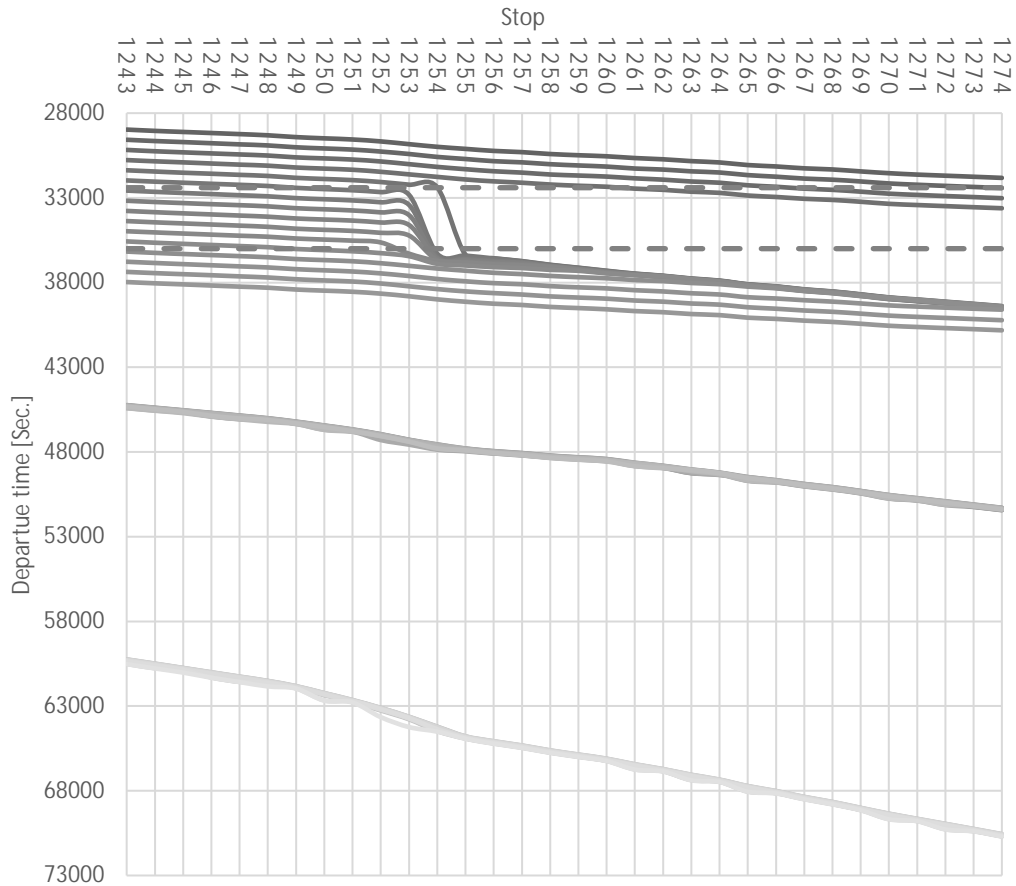


Figure 1: Departure times of all simulated trips of eastbound vehicles, with a disruption from 09:00 to 10:00, and without mitigation strategies; the disruption propagates through the whole operational day

much of their originally scheduled trips as possible. In addition to just short-turning affected trips, route separation also requires adjustments to the timetable and the vehicle schedule: trips originally scheduled to be executed by vehicles stuck on the wrong side of the disruption have to be redistributed to vehicles able to execute them.

4 Experiments

To illustrate the impact of the described disruption mitigation strategies on the performance of public transit networks, they are implemented using the multi-modal public transit simulation described by Lückcrath (see [12]) and Lückcrath et al. (see [13]) and operational days of a simple artificial transit network with a single line are simulated.

Most interesting are measures that go beyond simple delay mitigation and address significant disruptions, e.g. a disrupted network section due to an extreme weather event. In the examined case, a track segment between two

stops (1254 and 1255) is blocked for one hour, from 09:00 to 10:00 am.

In the modeled example, the disruption management method is unable to find a suitable alternative route and also finds short-turning and stop skipping to be ineffective. Subsequently, it decides to split the route before/after the disrupted network section and adjust timetable and vehicle schedule accordingly.

Operational days are simulated both with the described mitigation actions and without any mitigation actions. While the latter case certainly is unrealistic, it is used as an illustrative example for a worst case scenario, where nothing is done to address a disturbance. In this case, the disruption causes extreme delays of between one to two hours throughout the whole operational day (see Figure 1), resulting in massive vehicle bunching right before the disrupted network section and a severe degradation of service quality. Each vehicle is delayed before the disrupted track segment until the outage is re-

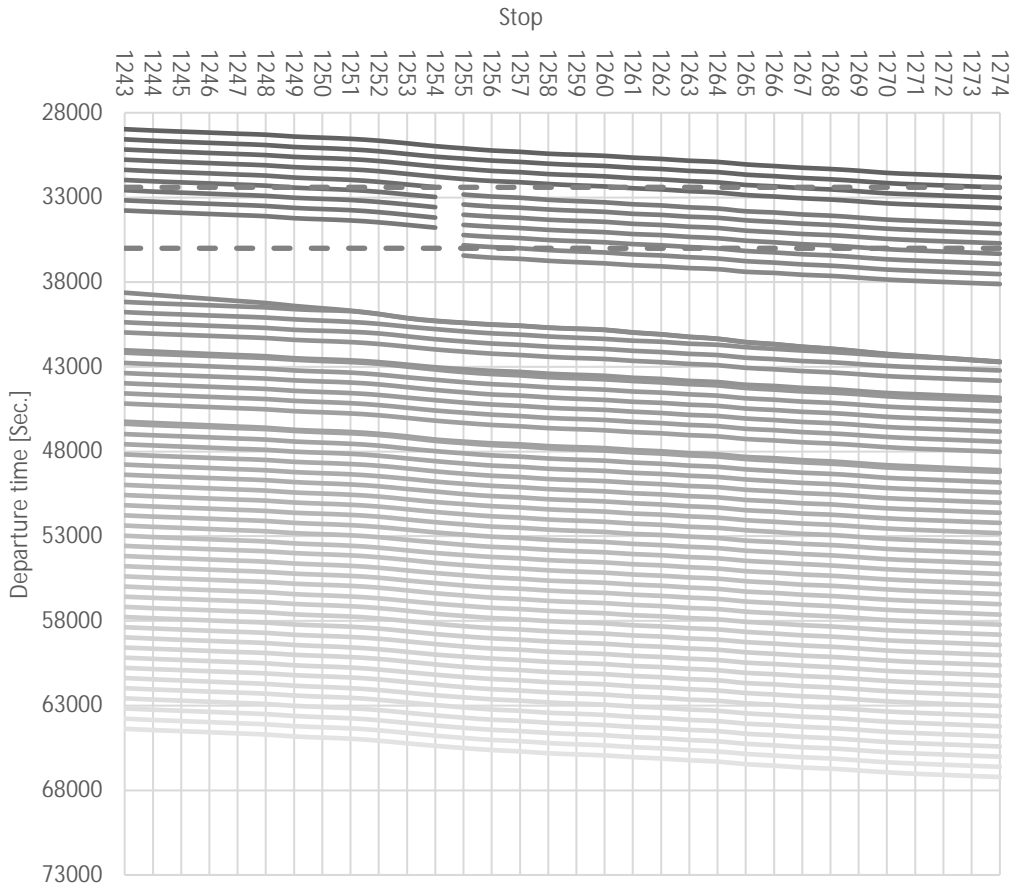


Figure 2: : Departure times of all simulated trips of eastbound vehicles, with a disruption from 09:00 to 10:00, and with active mitigation strategies; after approx. one hour the disruption is successfully mitigated

pealed. Once the track segment is usable again, all vehicles resume service as one single column and for the rest of the operational day these massive delays cannot be made up for.

With active mitigation actions delays are reduced to under one minute one hour after the service resumes (see Figure 2). In addition, during the disruption both parts of the disrupted network section are serviced. The disruption still exists – passengers planning to bypass the disrupted track segments have to find alternative means to cross it, but at least the majority of the operational day sees a comparatively high level of service restored.

However, even with active mitigation a 90 minute time window without active transit service exists. This is due to the fact that the current iteration of the mitigation strategy does not employ any potentially available replacement vehicles and does not allow vehicles to perform deadheading trips to reach their regular starting stop after the disturbance subsided. Therefore, vehicles exe-

cuting short-turned trips on each side of the disrupted network section have to wait and their respective side of the disrupted network section until their regularly scheduled trips from this side resume.

5 Conclusion

This paper presented short-term, rule-based operational strategies to address the impacts of climate change-related extreme weather events on public transit networks and how simulation models can be used to evaluate their effectiveness as well as the resulting primary and secondary impacts of extreme weather events on public transit networks. Following an introduction to motivation and subject matter of the work, the H2020 project RESIN was described, that aims at increasing the resilience of cities and their critical infrastructure components. Finally, the developed operational strategies were described before their mechanics were explored by conducting simulation

experiments on a model of an artificial public transit network.

In further steps, the set strategies will be refined and extended, and simulation experiments based on representations of real public transit networks will be conducted. After rigorous validation, the operational strategies will then be integrated with RESIN's guideline documents addressing decision makers in municipalities and urban districts.

6 Acknowledgments

The authors thank their partners in the RESIN consortium for their valuable contributions during the development and test process. This paper is based in part upon work in the framework of the European project "Climate Resilient Cities and Infrastructures – RESIN". This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 653522.

The sole responsibility for the content of this publication lies with the authors. It does not necessarily represent the opinion of the European Union. Neither the EASME nor the European Commission are responsible for any use that may be made of the information contained therein.

References

- [1] Borne, P., Fayeche, B., Hammadi, S., Maoche, S.: Decision support system for urban transportation networks. In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, Vol. 33, No. 1, S. 67–77, 2013.
- [2] Cats, O.: Multi-agent Transit Operations and Assignment Model. In: *Procedia Computer Science*, Vol. 19, S. 809–814, 2013.
- [3] Chapman, R.A., Michael, J. F.: Modelling the Tendency of Buses to Form Pairs. In: *Transportation Science*, Vol. 12, No. 2, S. 165–175, 1978.
- [4] Coletti, A., De Nicola, A., Villani, M.L. Building climate change into risk assessment. In: *Natural Hazards*, Vol. 84, No. 2, pp. 1307–1325, 2016.
- [5] Corman, F., D'Ariano, Hansen, I.A.: Disruption handling in large railway networks. In: *WIT Transactions on The Built Environment*, Vol. 114, S. 629–640, 2010.
- [6] Dreot, J.: Petrowski, A., Siarry, A.: Taillard, E.: Metaheuristics for Hard Optimization. Springer 2006.
- [7] Ginkel, A., Schöbel, A.: To wait or not to wait? The bicriteria delay management problem in public transportation. In: *Transportation Science*, Vol. 41, No. 4, S. 527–538, 2007.
- [8] Hadas, Y., Ceder, A.: Optimal coordination of public-transit vehicles using operational tactics examined by simulation. In: *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 6, S. 879–895, 2010.
- [9] Intergovernmental Panel on Climate Change (IPCC). Summary for Policymakers, in: *Climate Change 2014: Impacts, Adaptation, and Vulnerability. Part A: Global and Sectoral Aspects*. Cambridge University Press, Cambridge, United Kingdom, and New York, NY, USA, pp. 1–32, 2014.
- [10] Kendziorra, A. ; Weber, M: Extensions for logistic and public transport in SUMO. In: *SUMO 2015 - Intermodal Simulation for Internal Transport, Berichte aus dem DLR-Institut für Verkehrstechnik*. Vol. 28. Berlin, S. 83–90, 2015.
- [11] Klierer, N., Suhl, L.: A note on the online nature of the railway delay management problem. In: *Networks*, Vol. 57, No. 1, S. 28–37, 2011.
- [12] Lückcrath, D.: Ein Simulationsmodell für Öffentlichen Personennahverkehr mit regelbasiertem Verkehrsmanagement. Dissertation, Universität zu Köln, 2017.
- [13] Lückcrath, D., Speckenmeyer, E., Ullrich, O., Rische, N.: A Mesoscopic Bus Transit Simulation Model Based on Scarce Data. In: *Simulation Notes Europe (SNE)*, Vol. 28, No. 1, 2018, to appear.
- [14] Meignan, D., Simonin, O., Koukam, A.: Simulation and evaluation of urban bus-networks using a multiagent approach. In: *Simulation Modelling Practice and Theory*, Vol. 15, No. 6, S. 659–671, 2007.
- [15] Nagel, K.: Multi-modal traffic in TRANSIMS. In: *Pedestrian and evacuation dynamics*, S. 161–172, 2001.
- [16] Nemry, F., Demirel, H.: *Impacts of Climate Change on Transport: A focus on road and trail transport infrastructures*, 2012 Joint Research Centre Institute for Prospective Technological Studies, JRC72217, EUR 25553 EN, 93 pg., 2012.
- [17] Nesheli, M., Ceder, A.: A robust, tactic-based, real-time framework for public-transport transfer synchronization. In: *Transportation Research Part C: Emerging Technologies*, Vol. 60, S. 105–123, 2015.
- [18] Osogami, T., Imamichi, T., Mizuta, H., Morimura, T., Raymond, R., Suzumura, T., Takahashi, R., Idé, T.: *Research Report IBM Mega Traffic Simulator*: Technical report, 2012.
- [19] RAMSES – Reconciling adaptation, mitigation and sustainable development for cities. URL, visited on Dec 05, 2017, <http://www.ramses-cities.eu/>
- [20] RESIN – Climate Resilient Cities and Infrastructures. URL, visited on Nov 11, 2017: <http://www.resin-cities.eu/>
- [21] Rossetti, M A: Potential impacts of climate change on railroads, In: *Workshop on the Potential Impacts of Cli-*

- mate Change on Transportation*, Washington, D.C., October 1-2, 2002.
- [22] SMR – Smart Mature Resilience. URL, visited on Dec 05, 2017. <http://smr-project.eu>
- [23] SUMO – Simulation of Urban Mobility. URL, visited on May 08, 2018: <http://sumo.dlr.de/index.html>
- [24] Suzumura, T., Kanezashi, H.: Multi-modal traffic simulation platform on parallel and distributed systems. In: *Proceedings of the 2014 Winter Simulation Conference*: IEEE Press, S. 769–780, 2014.
- [25] Suzumura, T., McArdle, G., Kanezashi, H.: A high performance multi-modal traffic simulation platform and its case study with the Dublin city. In: *Proceedings of the 2015 Winter Simulation Conference*: IEEE Press, S. 767–778, 2015.
- [26] Toledo, T., Cats, O., Burghout, W., Koutsopoulos, H. N.: Mesoscopic simulation for transit operations. In: *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 6, S. 896–908, 2010.
- [27] United Nations, Department of Economic and Social Affairs, Population Division. *World Urbanization Prospects: The 2014 Revision, Highlights*. United Nations, 2014.
- [28] Xie, J., Bogen, M., Lückerrath, D., Rome, E., Sojeva, B.; Ullrich, O., Worst, R.: Assessment of Climate Change-Related Risks and Vulnerabilities in Cities and Urban Environments. In: *Interoperability for Crisis Management: Increasing Resilience of Smart Cities (ICRIM 2018)*, 2018, to appear.
- [29] Yu, B. Yang, Z.: A dynamic holding strategy in public transit systems with real-time information. In: *Applied Intelligence*, Vol. 31, No. 1, S. 69-80, 2009.
- [30] Zolfaghari, S., Azizi, N., Jaber, M. Y.: A model for holding strategy in public transit systems with real-time information. In: *International Journal of Transport Management*, Vol. 2, No. 2, S. 99–110, 2004.

Agent-based Simulation with Process-Interaction Worldview

Shufang Xie, Tao Zhang^{*}, Oliver Rose

Fakultät für Informatik, Universität der Bundeswehr München, Werner-Heisenberg Weg 39, D-85577 Neubiberg, Germany; ^{*} tao.zhang@unibw.de

Abstract. After analyzing the agent-based simulation (ABS), we realize that most researchers focus on the agent-based modeling, but only a few researchers pay attention to the simulation. The reason is that the agent-based model (ABM) can run directly in a real-time manner by communication among agents and the running of the ABM is already one kind of simulation. However, the ABS is less efficient when being used into a non-real-time system even though it can be speeded up by giving a timescale. So, to speed up the ABS, we introduce a process-interaction worldview (PIW) originated in the discrete event simulation to the ABS. A method for combining the agent-based simulation and the PIW is proposed. The method is validated by applying in a simple queue system and compared with the normal ABS with a timescale.

Introduction

There still exists controversy over the definition of agents. One concept of the agent appears in the distributed intelligence in Artificial Intelligence[1], in which besides distributed, autonomous, and social features, agents have to be intelligent, such as being able to perceive, learn, and adapt to the environment. Most of the researchers following this concept direct towards the multi-agent system[2], and some researchers focus on the hardware agent[3] which has been widely used in the robotics[4]. Another concept, which is derived from the emergence theory[5] in which a key notion is that simple rules generate complex behaviors, in other words, system properties emerge from its constituent agent interactions[6], is very similar to the cellular automata[7] in which agents are asked to keep simple and short, which is contrary to the first concept because the intelligence certainly makes agents more complex. However, they

also have a lot in common, such as autonomy, society, distribution and so on. In most research, the contrary parts of the agents seem to be discarded. Both the intelligence and the simplicity are not given weight, but much attention is paid to the common parts, autonomy, society, and distribution [8-10]. These research lead to the formation of a new bottom-top modeling method[11], i.e., agent-based modeling, in which a system is modeled as a collection of autonomous agents. Based on a set of rules each agent individually assesses its situation, makes decisions and may execute various appropriate behaviors for the system it represents[6]. The agent-based modeling is kind of similar to the object-oriented modeling[12], but more flexible and more natural to describe the system. There are no strict requirements of intelligence or simplicity for the agents in the agent-based model. Now we define the ABS. The ABS is the process of designing an ABM of a real system and conducting experiments with this model to understand the behavior of the system and evaluating various strategies for the operation of the system[13].

At present, the researches on the ABS mainly focus on two directions, agent-based modeling, and its application. The first one is trying to build a standard, and universal framework for the modeling [14-16]. Autonomy, society, and distribution involve in. Another one focuses on creating domain agents through studying their attributes and behaviors [17-19]. Both have provided lots of approaches and mechanisms for agent-based modeling and application in practice. However, only a few researches focus on the simulation and its efficiency. The study will discuss the efficiency of the non-real-time ABS and combine the ABS with the discrete event simulation to boost the efficiency of the simulation. The paper is structured as follows: In Section 1, we present the efficiency problem of the ABS and analyze some

worldviews which may speed up the simulation. In Section 2, a procedure for introducing process-interaction worldview to the ABS is proposed. In Section 3, some experiments are carried out to validate the procedure. The paper is concluded in Section 4.

1 Agent-based Simulation

1.1 Simulation in ABS

As we mentioned before, the current studies on the ABS focus on the ABM, and few researchers pay attention to the simulation in the ABS. Probably because the ABM can run directly in a real-time way [11], in which the running of the ABM is a simulation run, and there is no necessity to study the simulation separately. However, this type of simulation misses some important contents in the computer simulation such as the worldview in the simulation, also referred to as a simulation strategy. Therefore, many researchers studying on the computer simulation, especially the discrete event simulation, cannot help asking where the simulation is in the ABS. The so-called simulation in the ABS is a real-time simulation in which the simulation time equals the real time. However, the ABS is less efficient when being used in a non-real-time system. In some researches, the ABS is speeded up by giving a timescale under the condition of synchrony. There are two ways to achieve the synchronization: conservative algorithm and optimistic algorithm. The Conservative algorithm keeps the model running in sync exactly, but the optimistic algorithm allows asynchronous phenomenon to occur and then makes it synchronous, such as SimJade does. The SimJade [20] is a synchronization service for the JADE using an optimistic synchronization technique to manage the time in a distributed way. Because the optimistic synchronization techniques allow the asynchronous phenomenon to appear, the agents influenced by the asynchronous phenomenon have to roll back. So, lots of time is consumed by the roll-back.

In addition, the real-time ABS with a timescale has two features: (1) there is no central time manager, and the agents move on according to their local time (computer clock); (2) The time spent on executing code is counted in the simulation time. In the real-time ABS without a timescale, the time for code execution is very short and can be ignored. On the other hand, in the real world, it also takes time while people make a decision. So, the

time for executing the codes exists reasonably in the real-time ABS. However, in the non-real-time ABS with the time scale, the execution time is enlarged, and errors occur. And the worse thing is that the precision of simulation results will decrease as the timescale increases. We face a tradeoff between the efficiency and the precision. So how to speed up the ABS without losing any precision is a key issue.

1.2 Efficiency of the ABS

If back to the computer simulation again, we can find that the simulation has different efficiencies when different worldviews are used. There are two main types of the worldview, time-driven worldview, and event-driven worldview. In the time-driven worldview, the world progresses as the time is passing with a fixed increment (time step). Correspondingly, the world progresses as some events occur in the event-driven worldview which includes three sub worldviews: event scheduling, activity scanning, and process interaction. Introducing a suitable worldview into the ABS will be a good way to speed up the simulation. But some researchers [12, 20] argue that the simulation worldview violates the autonomy principle of the agents due to the centralization of time handling and sharing. So, most researchers did not study the ABS and the worldview together. Siebers even declares that the discrete event simulation is dead, long live the ABM [21]. However, the time handling is only in charge of simulation time which is independent and never affected by the agent. The local clocks in each agent merged into one sharing simulation clock does not intervene in the behavior of agents at all, and the agents still take action autonomously. Therefore, it is possible to introduce the worldview to the ABM.

A few researches have already focused on this field and obtained great progress, such as the entity-relationship and agent-oriented-relationship (ER\AOR) [22]. Agents, objects, events, and messages are entities in the ER\AOR; the agents and the objects are distinguished; the messages and the events are managed together to control the simulation time. In the ER\AOR, it is natural to partition the simulation system into the environment simulator and some agent simulators. The environment simulator is responsible for advancing simulation time and managing the state of all external (or physical) objects and the external/physical state of each agent; a number of agent simulators are in charge of managing the internal (or mental) state of agents. By means of ER modeling and

combination with the discrete event simulation, the ER\AOR has attracted extensive attention. However, we realize that the ER\AOR simulation is not a pure ABS because of the objects in the model. In the pure ABS, the objects which cannot be modeled as agents must belong to certain agents (attributes). In addition, because the conditional events or messages are involved, and the lifecycle of the agent is divided into many activities, the AOR simulation is the ABS with the activity-scanning worldview. As we all know, the activities-scanning worldview is not the most efficient one. It is better to choose a more efficient worldview for the ABS. Now we come to the next question: which worldview is more suitable and efficient for the ABS.

1.3 Worldview for the ABS

The event-scheduling worldview focuses on the events that instantaneously transform a system’s state and schedule future events [23]. The advantage of this worldview is that periods of inactivity can be skipped over by jumping the clock from one event time to the next event time. The event-based approach is the most computationally efficient one of the three classical worldviews. The activity-scanning worldview focuses on activities and their preconditions (triggers)[23]. An activity’s preconditions must be satisfied for an activity’s operations. This worldview is less efficient than the event-scheduling worldview because it requires a frequent evaluation of conditions. The process-interaction worldview can be considered a combination (hybrid) of the activity-scanning worldview and the event-scheduling worldview [24]. It focuses on processes and the entities that flow through the processes and interact with resources [25]. The process-interaction worldview is more efficient than the activity-scanning worldview, but it is less efficient than the event scheduling worldview.

From the analysis above, we can see that the process-interaction worldview is the second most efficient worldview. But besides efficiency, the choice of worldview should be made by considering other characteristics such as maintainability, modifiability, reusability, and ease of development. The process-interaction worldview is considered to be a natural way to describe models [26] and is closer to most people’s mental model. In addition, the notion of “process” corresponds closely to the lifecycle of the agent and the implementation of the process-interaction worldview is very similar to the

agent-based model. Moreover, if we use the event-scheduling worldview or the activity-scanning worldview, the flexibility, maintainability, and modifiability of the ABM will shrink. Therefore, we introduce the process-interaction worldview into the ABS to speed up the simulation.

2 Agent-based Simulation with the PIW

In this section, we will introduce an approach (ABS&PIW) which brings the process-interaction worldview into the agent-based simulation. First, we discuss agents in detail including its attributes, behaviors, and messages. And then some concepts in the ABS&PIW are defined. To make the approach more formal and rigorous, we formulate the approach. At last, we discuss the parallelism in the ABS&PIW.

2.1 Agents in the Agent-based Simulation

In our study, an agent has some attributes, behaviors, messages, and activation points (see Figure 1). Agents are endowed with behaviors to make independent decisions, and the messages are medium for their communications. Activation points are designed for the process-interaction worldview.

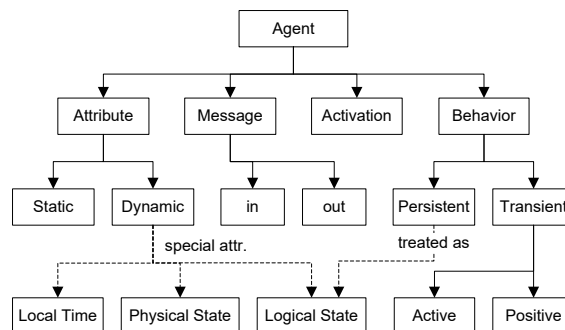


Figure 1: Attributes, behaviors, messages, and activations in an agent

Attributes.

Attributes are characteristics of the agent. An agent’s attributes can be static, i.e., not changeable during the simulation, or dynamic, i.e., changeable by behaviors as the simulation progresses. For example, a static attribute is an agent’s name; a dynamic attribute is an agent’s memory of past interactions. The agent adapts to the environment by changing its attributes. There can be a large number of attributes in an agent, but only attributes related to the goal of the system need to be considered. The

agent has three special attributes: local time, physical state, and logical state. The local time is from the inner clock in the agent, and it may be not synchronous with the simulation time. The two states will be defined in section 2.2.

Behaviors

There are two types of behaviors: persistent behaviors and transient behaviors. The persistent behaviors are equal to activities, and they will change the state of the agent. One persistent behavior is related to one of the logical states, so the persistent behaviors are treated as logical states. Transient behaviors, which are behaviors considered in the agents, can be divided into passive behaviors and active behaviors. The passive behaviors are responsible for receiving messages and updating dynamic attributes, and the active behaviors are in charge of generating and sending messages.

Messages and Activation

Agents can receive messages from the other agents and send messages to them. The message has a given format and typically contains sender, receivers, sending time, keywords, and content. The sending time is the local time of the sender agent when the message is sent. The local time is updated with the sending time of the received messages. The concept of activation will be introduced in section 2.2

2.2 Concepts in the ABS&PIW

Concepts in the ABS&PIW

Firstly, six concepts are given. Delays and activation points come from the process interaction method but offer some improvements.

Physical state of the agent, which has two states, active and blocked, is related to the implementation. If an agent is blocked, it gives up control of the CPU. Otherwise, it occupies the CPU.

Logical state of the agent is closely connected to the application domain and is as same as the state of the entity. The logical state is a very important dynamic attribute.

State of the agent-based model has two types, ready or unready. If the physical states of all agents are blocked, the state is ready. Otherwise, it is unready.

Straggler message is a message that its sending time is earlier than the local time of the agent who receives the

message. It means a later message is received before the earlier message. The straggler message will make the simulation wrong and must be avoided. There are two ways to avoid it: a conservative algorithm which does not allow the straggler message and an optimistic algorithm which allows it and then corrects it.

Delay is a period in which the logical state of the agent is not changeable. When a delay occurs, the agent will create the next activation point and become blocked.

Activation point is the time position where the agent will be after a delay ends. The agent is activated at this point and performs actions until a new delay occurs. An activation point has such a given format including activation time, activation agent, and keywords. There are two types of activation point, conditional and non-conditional. Non-conditional activation point is explicit. In contrast, the conditional activation point is uncertain in which delays of the agents does not end until the agents meet the given condition.

Relation among some Concepts

The relation among these concepts is shown in Figure 2. The agent is similar to the active entity; the life cycle of the agent is the process of the entity and is made up of a series of activation-delay-activation. The activation point is located at the time an event occurs, and the physical state is active at this point. The agent responds to the event by a transient behavior. During a delay, an activity is carrying out, and the physical state of the agent is blocked. The activity is a persistent behavior corresponding to the certain logical state.

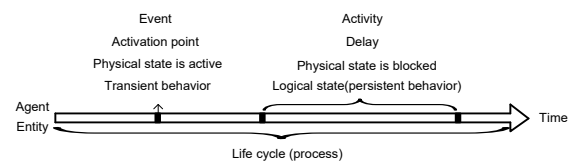


Figure 2: Relation between activation point, event, and so on

2.3 Formulation of the ABS&PIW

Symbol Definition

t, t_{plan} current and planned simulation time

t_{sa} sampling interval time

s_{mo} state of the ABM(0-ready,1-unready)

u flag of model update (0-no update,1-update)

R_{FAL}, R_{FAL}^0 activations in the future activation list and initial value

R_{CoAL}, R_{CoAL}^0 activations in CoAL and initial value

R_{CuAL} current activations in CuAL

a an agent, $a \in A, A$ is a set of all agents

r_a current activation point of the agent a

c_a type of activation point(1-conditional,0-uncondt.)

g_{r_a} flag of the condition(0-unmeet,1-meet)

R_a a set of activation types from the agent a

m_a a message received by the agent a

M_a^{out} a set of message types sent by the agent a

M_a^{in} a set of message types received by a

τ_{m_a} timestamp of the message m

τ_r activation time (timestamp of the activation point)

τ_a local time of agent a updated by the time stamp

τ_{m_a} or τ_r

s_a physical state of the agent a

f_a agent attribute, $f_a \in F_a$

F_a, F_a^0 a set of all attributes and initial value

b agent behavior

B_a^{po}, B_a^{ac} a set of passive and active behaviors

$H_a^1 = RS(r_a, b_a^{po}) = \{h_{r,b}^1 \mid r \in R_a, b \in B_a^{po}\}$ relationship

between activations and passive behaviors . If b is

related to $r, h_{r,b}^1 = 1$, otherwise $h_{r,b}^1 = 0$. The

following relationships have the same rule.

$H_a^2 = RS(m_a^{in}, b_a^{po}) = \{h_{m,b}^2 \mid m \in M_a^{in}, b \in B_a^{po}\}$

$H_a^3 = RS(r_a, b_a^{ac}) = \{h_{r,b}^3 \mid r \in R_a, b \in B_a^{ac}\}$

$H_a^4 = RS(m_a^{in}, b_a^{ac}) = \{h_{m,b}^4 \mid m \in M_a^{in}, b \in B_a^{ac}\}$

$H_a^5 = RS(b_a^{po}, f_a) = \{h_{b,f}^5 \mid b \in B_a^{po}, f \in F_a\}$

$H_a^6 = RS(b_a^{ac}, a', m_a^{out}) = \{h_{b,a',m}^6 \mid b \in B_a^{ac},$

$a' \in A, m \in M_a^{out}\}$

$H_a^7 = RS(b_a^{ac}, r_a) = \{h_{b,r}^7 \mid b \in B_a^{ac}, r \in R_a\}$

Four-tuple of the ABS&PIW

We provide a mathematical framework for the ABS following the process interaction worldview. The simulation is specified as a four-tuple $SIM = (I, TM, ABM, O)$.

I is a set of inputs $I = (R_{FAL}^0, R_{CoAL}^0, \{F_a^0 \mid a \in A\})$ including initial activation points and initial attributes of all agents.

O is a set of outputs $O = (\{(t, F_a^t) \mid a \in A, 0 < t < t_{aim},$

$\text{mod}(t, t_{sa}) = 0\})$, which is made up by attributes of all

agents at each sample point. TM is a time manager

$TM = (t, t_{plan}, R_{FAL}, R_{CoAL}, R_{CuAL})$ who is in charge of the

simulation time and manages all of the activation points

created by agents. The activation points are grouped into

three lists: conditional activation list (CoAL), future ac-

tivation list (FAL) and current activation list (CuAL).

FAL and CoAL are direct lists in which the activation

points come from agents. CuAL is an indirect list in

which the activation points are moved in from FAL. The

planned time t_{plan} is the maximal simulation time. The

simulation will end when the simulation time t reaches

the planned time. Note that there is no TM in the general

real-time ABS. ABM is an agent-based model described

as $ABM = (s_{mo}, u, \{(\tau_a, s_a, F_a, B_a^{po}, B_a^{ac}, M_a^{in}, M_a^{out}, R_a,$

$RS_a) \mid a \in A\})$. The model state s_{mo} and the updated flag

u are used by the TM to advance the simulation. The sim-

ulation clock advances whenever the model state is ready

and the model does not update anymore. In this way, we

can avoid the straggler messages and ensure that all con-

ditional activation points which meet the corresponding

conditions are activated as soon as possible. This is a con-

servative synchronization algorithm. The ABM is differ-

ent from the general one which contains three elements,

agents, relationship, and messages[11]. In a complex sys-

tem, the relationship changes dynamically and there are

massive situations. It is difficult to express the relation-

ship among all agents by a two-axis matrix. But for the

individual agent, the situations of relationship with other

agents are countable. So, in our ABM , the relationship

and messages are specified in the individual agents and

the target agents with corresponding messages can be got

by some simple IF-THEN rules. RS_a is a set of the rela-

tionship sets, $RS_a = (H_a^1, H_a^2, H_a^3, H_a^4, H_a^5, H_a^6, H_a^7)$,

which contains seven relationships in the agent such as

the relationship between received messages (in) and pas-

sive behaviors, as well as the relationship between active

behaviors, target agents and corresponding messages

(out). Local time and physical state are two special attrib-

utes and play a great role in the simulation. We extract

them from the attributes and consider separately. The

physical state is used to determine the model state.

Procedure of the ABS&PIW

The following is a procedure for the simulation. Simulation initialization (1), advancing time (2), and activating agents (3) are executed by the *TM*. The simulation is initialized with the inputs. Simulation clock t advances according to the time of the earliest activation points. All concurrent activation points, including both current activation points and conditional activation points, are activated at a time.

(1) Initialize

$$t = 0, R_{FAL} = R_{FAL}^0, R_{CoAL} = R_{CoAL}^0$$

$$F_a = F_a^0, s_a = 0, \text{ where } a \in A$$

$$s_{mo} = 1$$

(2) Advance time

if $R_{FAL} = \phi$ or $t > t_{plan}$ then simulation ends

$$\tau_{min} = \min(\{\tau_r \mid r \in R_{FAL}\})$$

$$R_{CuAL} = \{r \mid \tau_r = \tau_{min}, r \in R_{FAL}\}$$

$$R_{FAL} = R_{FAL} - R_{CuAL}$$

$$t = \tau_{min}$$

(3) Activate

$$R' = \{R_{CoAL}, R_{CuAL}\}, R_{CuAL} = \phi$$

if $R' = \phi$ then go to (2)

$$A' = \{a' \mid r_{a'} \in R'\}$$

$$S_{A'} = 1, \text{ where } S_{A'} = \{s_{a'} \mid a' \in A'\}$$

TM activates A' at R'

(4) When an agent $a' \in A'$ is activated at $r_{a'} \in R'$

$$\tau_{a'} = \tau_{r_{a'}}$$

Get passive behavior b_1 by $H_{a'}^1$,

which satisfies $h_{r_{a'}, b_1}^1 = 1, h_{r_{a'}, b_1}^1 \in H_{a'}^1$

if $c_{r_{a'}} = 1$ then

$g_{r_{a'}} = g_{r_{a'}}^{new}$, where $g_{r_{a'}}^{new}$ is new one from b_1

If $g_{r_{a'}} = 0$ then $s_{a'} = 0$ and go to (6)

end if

if $F_{a'} \neq \phi$ then $F_{a'}' = F_{a'}^{n'}$,

where $F_{a'}^{n'}$ are new values calculated by b_1

and $F_{a'}' = \{f \mid h_{b_1, f}^5 = 1, f \in F_{a'}', h_{b_1, f}^5 \in H_{a'}^5\}$

$u=1$

Get active behavior b_2 by $H_{a'}^3$,

which satisfies $h_{r_{a'}, b_2}^3 = 1, h_{r_{a'}, b_2}^3 \in H_{a'}^3$

if $\exists r_{a'}' \in R_{a'}' : h_{b_2, r_{a'}'}^7 = 1, h_{b_2, r_{a'}'}^7 \in H_{a'}^7$, then

if $c_{r_{a'}'} = 0$ then $R_{FAL} = R_{FAL} + \{r_{a'}'\}$

if $n \ c_{r_{a'}'} = 1$ then $R_{CoAL} = R_{CoAL} + \{r_{a'}'\}$

end if

$(A'', M_{a'}^{out'}) = \{(a'', m) \mid h_{b_2, a'', m}^6 = 1, h_{b_2, a'', m}^6 \in H_{a'}^6\}$ if

$A'' \neq \phi$ then

$S_{A''} = 1$, where $S_{A''} = \{s_{a''} \mid a'' \in A''\}$

send messages $M_{a'}^{out'}$ to A''

end if

if $c_{r_{a'}} = 1$ then $R_{CoAL} = R_{CoAL} - \{r_{a'}\}$

$$s_{a'} = 0$$

go to (6)

(5) When an agent $a'' \in A''$ receives the message $m_{a''}$

from the agent a'

It is similar to step (4) and just needs the following replacements and to ignore condition activation:

$$a' \rightarrow a'', r_{a'} \rightarrow m_{a''}, H_{a'}^1 \rightarrow H_{a''}^2, H_{a'}^3 \rightarrow H_{a''}^4$$

(6) When an agent is blocked

if $\forall a \in A : s_a = 0$ then $s_{mo} = 0$

if $s_{mo} = 0$ then

if $u=1$ then $u=0$ and go to (3)

if $u=0$ then go to (2)

end if

In steps (4, 5), when an agent becomes active or receives messages, passive behaviors handle the received messages or activations and update its attributes. Active behaviors create new activation points and communicate with others. Decisions on the timing of advancing the time and quitting repeat of the conditional activations (6) are made by ABM according to the model state and the updated flag whenever the physical state of one agent becomes blocked.

Parallelism in the ABS&PIW

In the agent-based simulation, agents run in parallel. While concurrent activation points are activated simultaneously, associated agents will respond in parallel. The parallelism in ABS is shown in Figure 3. To avoid the straggler messages mentioned above, we adopt the conservative synchronization algorithms to ensure the correct local time (see step 6). Even though it cannot fully take advantage of parallelism, it can prevent the straggler messages from appearing at all and save the rollback time in the optimistic algorithm.

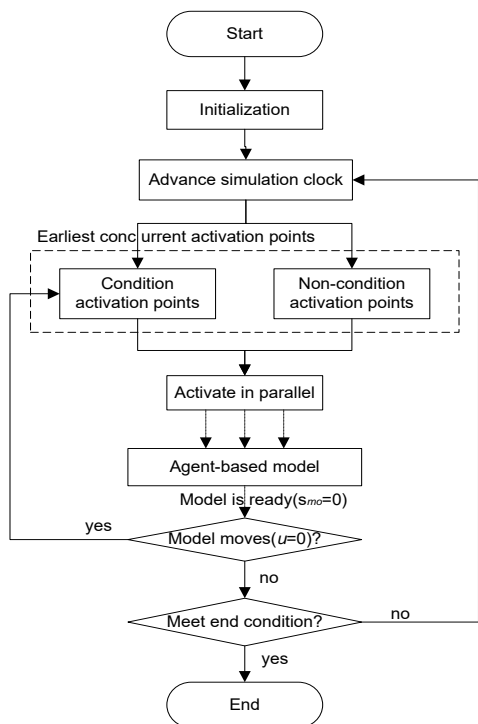


Figure 3: Parallelism in the agent-based model

3 Experiments

A queuing system $M / M^r / 1$ with a batch service is one of the classical discrete event systems. We use it to validate the proposed approach. An ABM of the system is built for the queue system, and the simulation result is compared with the theoretical value. We also compare the efficiency of the approach with the real-time ABS by using the built model.

3.1 Queuing System $M / M^r / 1$

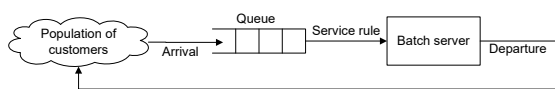


Figure 4: A Queuing system

The queuing system $M / M^r / 1$, shown in Figure, consists of an infinite population of customers, an infinite queue with FCFS (First Come First Serve) dispatching rule, and one batch server. The batch server provides service in batches (of size r) for arrived customers based on the rule. Customers who arrive and find the server busy join in the queue. Customers in a batch start service at the same time and depart together after served. The interval

arrival and service times follow exponential distributions.

3.2 Agent-based Model of the $M / M^r / 1$ Queuing System

Three types of agent are abstracted from the queuing system: customer source, customer, and server. Because a server is also an active entity, activation points of the whole system are simplified to two types: customer arrival and service completion. We build a single group ABM for the queuing system. The customer source generates customers according to a certain time distribution. The behaviors of the customers are requesting service, joining the queue to wait, accepting service, and leaving the system. The behaviors of the server include handling customer messages and providing service. The queue is part of the server agent. After a customer is served, the queue will use the given rule to choose new customers to begin service.

3.3 Correctness Verification

Assuming that customers arrive one by one; arrival rate λ is 9.76 per hour; the service batch r is 3; service rate μ is 5 per hour. Measures of performance in the steady state can be calculated with the following formulas[27].

The probability that n customers are in the system,

$$P_n = \begin{cases} (1 - s_0^{-n-1}) / r & 0 \leq n < r \\ \rho (s_0 - 1) s_0^{r-n-1} & r \leq n \end{cases}$$

where $\rho = \lambda / (r\mu)$, s_0 satisfies $|s_0| > 1$ and

$$r\rho s_0^{r+1} - (1 + r\rho)s_0^r + 1 = 0$$

The average number of customers in the system,

$$\bar{N} = (r - 1) / 2 + 1 / (s_0 - 1)$$

The average number of customers in the queue,

$$\bar{N}' = (r - 1) / 2 + 1 / (s_0 - 1) - r\rho$$

The average waiting time of customers,

$$\bar{W} = (r - 1) / (2\lambda) + 1 / (r\mu(s_0 - 1))$$

The average time of customers in the system,

$$\bar{T} = (r - 1) / (2\lambda) + 1 / (r\mu(s_0 - 1)) + 1 / \mu$$

Results	p_0	$p(n>0)$	\bar{N}'	\bar{N}	\bar{W} (min)	\bar{T} (min)
Theoretical	0.067	0.933	3.05	5.00	0.37	0.57
ABS&PIW	0.067	0.933	3.01	4.96	0.31	0.51

Table 1: Comparison between theoretical result and simulation result

3.4 Comparison with the Real-time ABS (Time Scale)

In order to prove the less efficiency and precision, a brief simulator for the real-time ABS with timescale is developed by using JADE (Java Agent Development Environment). In JADE, the messages are not in sync sent and received (asynchronous communication). To achieve the simulation with the timescale, we improved it to the synchronous communication. After improvements, an agent (1) who just sends a message (a) will move on only after the message (a) is received and handled by its receiver

agent (2). If the receiver (2) needs to send another message (b) in the message (a)'s handling process, the agent (1) has to wait for the receiver (2) until its message (b) is received and handled by another receiver. In addition, when a delay occurs in the agent, the agent will be blocked until the delay ends. The timescale is used in such delays to decrease the delay time so as to speed up the simulation.

We still use the queuing system but with the constant arrival rate (3 per hour) and service rate (3 per hour) to avoid the stochastic influences. The theoretical values can be got easily, shown in Table 2. The simulation runs ten days, and the simulation results from our approach and the real-time ABS are shown in Table 2 too.

Results	p_0	$p(n>0)$	\bar{N}	\bar{N}	\bar{W} (min)	\bar{T} (min)	Time spent (s)
Theoretical	0.000	1.000	1.00	4.00	20.0	80.0	-
ABS&PIW	0.000	1.000	1.00	4.00	20.0	80.0	0.374
ABS (scale 70000)	0.002	0.998	1.07	3.20	24.6	86.4	12.300
ABS (scale 500000)	0.041	0.959	1.05	1.85	147.5	300.4	1.743

Table 2: Comparisons with real-time ABS (time scale)

We can see that the results of our approach are same as the theoretical values. It takes only 0.374 seconds (the configuration of hardware is Intel i3-330M 2.13GHz CPU and 2GB memory). However, for the ABS with the timescale, the error is very big, and it also took the longer time.

4 Conclusions

Because the PIW is more natural and closer to the mental model, it is combined with the ABS to speed up the simulation. The ABS&PIW approach is proposed on the basis of the agent-based model. We provide a four-tuple $SIM = (I, TM, ABM, O)$ with elements, inputs (I), time manager(TM), ABM, and outputs (O) to describe the approach strictly. The procedure of the approach is presented mathematically in which the simulation clock advances in a sequence of activation points and all concurrent activations are activated at a time, and associated agents respond in parallel. A conservative algorithm is adopted to avoid straggler messages while the simulation is running. The result from an application to the queuing

system $M / M' / 1$ shows the validity of the proposed approach. Comparing the efficiency with the real-time ABS, it performs more efficiently. Besides the advantages mentioned above, the ABM can be naturally combined with the process interaction worldview. The flexibility, maintainability, and modifiability of the ABM are also enhanced in this way.

References

- [1] Sugumaran, V., *Distributed Artificial Intelligence, Agent Technology, And Collaborative Applications*. 2008: Information Science Reference.
- [2] Lian, J., S.M. Shatz, and X. He, *Flexible coordinator design for modeling resource sharing in multi-agent systems*. Journal of Systems and Software, 2009. **82**(10): p. 1709-1729.
- [3] Tapia, D.I., et al., *Integrating hardware agents into an enhanced multi-agent architecture for Ambient Intelligence systems*. Information Sciences, (0).
- [4] García, C., et al., *Design and modeling of the multi-agent robotic system: SMART*. Robotics and Autonomous Systems, 2012. **60**(2): p. 143-153.

- [5] Namatame, A., *Collective Intelligence of Networked Agents*, in *Emergent Intelligence of Networked Agents*, A. Namatame, S. Kurihara, and H. Nakashima, Editors. 2007, Springer Berlin Heidelberg. p. 159-176.
- [6] Bonabeau, E., *Agent-based modeling: Methods and techniques for simulating human systems*. Proceedings of the National Academy of Sciences of the United States of America, 2002. **99**(Suppl 3): p. 7280-7287.
- [7] Wolfram, S., *Cellular automata and complexity*. 1994: Addison-Wesley Pub. Co.
- [8] Váncza, J. and A. Márkus, *An agent model for incentive-based production scheduling*. Computers in Industry, 2000. **43**(2): p. 173-187.
- [9] Oliva, G., S. Panzieri, and R. Setola, *Agent-based input-output interdependency model*. International Journal of Critical Infrastructure Protection, 2010. **3**(2): p. 76-82.
- [10] McLane, A.J., et al., *The role of agent-based models in wildlife ecology and management*. Ecological Modelling, 2011. **222**(8): p. 1544-1556.
- [11] Macal, C.M. and M.J. North, *Tutorial on agent-based modelling and simulation*. J of Sim, 2010. **4**(3): p. 151-162.
- [12] Davidsson, P., *Multi Agent Based Simulation: Beyond Social Simulation*, in *Proceedings of the Second International Workshop on Multi-Agent-Based Simulation-Revised and Additional Papers*. 2001, Springer-Verlag. p. 97-107.
- [13] Shannon, R.E., *Systems simulation: the art and science*. 1975: Prentice-Hall.
- [14] Botta, N., et al., *A functional framework for agent-based models of exchange*. Applied Mathematics and Computation, 2011. **218**(8): p. 4025-4040.
- [15] Fortino, G. and W. Russo, *ELDAMeth: An agent-oriented methodology for simulation-based prototyping of distributed agent systems*. Information and Software Technology, 2012. **54**(6): p. 608-624.
- [16] Grimm, V., U. Berger, and F. Bastiansen, *A standard protocol for describing individual-based and agent-based models*. Ecological Modelling, 2006. **198**(1-2): p. 115-126.
- [17] Chen, L., *Agent-based modeling in urban and architectural research: A brief literature review*. Frontiers of Architectural Research, 2012. **1**(2): p. 166-177.
- [18] Rebaudo, F. and O. Dangles, *An agent-based modeling framework for integrated pest management dissemination programs*. Environmental Modelling & Software, (0).
- [19] Cao, Y. and X.-h. Chen, *An agent-based simulation model of enterprises financial distress for the enterprise of different life cycle stage*. Simulation Modelling Practice and Theory, 2012. **20**(1): p. 70-88.
- [20] Pawlaszczyk, D. and I.J. Timm, *A hybrid time management approach to agent-based simulation*, in *Proceedings of the 29th annual German conference on Artificial intelligence*. 2007, Springer-Verlag: Bremen, Germany. p. 374-388.
- [21] Siebers, P.O., et al., *Discrete-event simulation is dead, long live agent-based simulation!* J of Sim, 2010. **4**(3): p. 204-210.
- [22] Wagner, G., *AOR Modelling and Simulation: Towards a General Architecture for Agent-Based Discrete Event Simulation*, in *Agent-Oriented Information Systems*, P. Giorgini, B. Henderson-Sellers, and M. Winikoff, Editors. 2004, Springer Berlin / Heidelberg. p. 174-188.
- [23] Miller, J.A., et al., *Investigating Ontologies for Simulation Modeling*, in *Proceedings of the 37th annual symposium on Simulation*. 2004, IEEE Computer Society. p. 55.
- [24] Zeigler, B., H. Praehofer, and T. Kim, *Theory of Modeling and Simulation, Second Edition*. 2000: Academic Press.
- [25] Banks, J. and J.S. Carson, *Process-interaction simulation languages*. SIMULATION, 1985. **44**(5): p. 225-234.
- [26] Franta, W.R. and K. Maly, *An efficient data structure for the simulation event set*. Commun. ACM, 1977. **20**(8): p. 596-602.
- [27] Gelenbe, E. and G. Pujolle, *Introduction to queueing networks*. b, ed. a. Vol. c. 1987, New York: John Wiley & Sons. 900.

Agent-based Simulation of Job Shop Production

Tao Zhang^{*}, Shufang Xie, Oliver Rose

Fakultät für Informatik, Universität der Bundeswehr München, Werner-Heisenberg Weg 39, D-85577 Neubiberg, Germany; ^{*} tao.zhang@unibw.de

Abstract. The most important design principles of Industry 4.0 are decentralization and intelligence. The entities, like resources and materials, can make decisions on their own by means of cyber-physical systems. For research purposes, we unify cyber entities and physical entities and build an agent-based simulation model. The agents learn knowledges offline during simulation runs and become smarter and smarter. The model will finally connect to the physical systems and carry out online decision-making. The study is the first stage of the whole project. A framework for the agent-based simulation is developed and an agent-based model of the job shop production including release agents, machine group agents, and job agents are built.

Introduction

Industry 4.0[1] is currently one of the most frequently discussed topics among practitioners and academics. Taking advantages of cyber-physical systems[2], the internet of things[3] and the internet of services, it enables faster, more flexible, and more efficient processes to produce higher-quality goods at reduced costs in an environmental-friendly and resource-conserving way. This in turn will increase manufacturing productivity, shift economics, foster industrial growth, and modify the profile of the workforce—ultimately change the competitiveness of companies and regions. Most of studies and industry cases about Industry 4.0 still stay at a very basic level, such as studies and cases about sensors, communications, and automations. These are certainly foundations of Industry 4.0. However, once these foundation works are done, there will be a very urgent requirement for more effective decision-making methods on the platform of Industry 4.0. As we all know, the most important design principles of Industry 4.0 are decentralization and intelli-

gence[4]. The cyber-physical systems can make decisions on their own. To make the cyber-physical systems intelligent, the systems must have abilities to learn knowledge. Online learning and offline learning are two common learning techniques. For the research purposes, the online learning is unpractical. In the study, we will create an agent-based simulation model (cyber system) and the agents (cyber entities) will learn to make decisions from the simulation. At last, the model will connect to the physical systems and form the cyber-physical systems.

The study is an extension of our previous works[5, 6] in which we realized that the agent-based simulation(ABS) is less efficient when being used into a non-real-time system even though it can be speeded up by giving a timescale. So, to speed up the ABS, we introduced a process-interaction worldview (PIW) originated in the discrete event simulation to the ABS. In this study, we are going to design a framework for the ABS with the PIW and build a model of the job shop production based on the framework. In the model, each agent is related to one physical entity. The agents can make decisions by either some decision rules or their knowledge learnt from some other independent simulation-based approaches that we have proposed[7-9].

1 Agent-based Simulation with the PIW

The ABS&PIW approach [5, 6] was proposed on the basis of the agent-based model(ABM). We provided a four-tuple $SIM = (I, TM, ABM, O)$ with elements, inputs (I), time manager(TM), ABM , and outputs (O) to describe the approach strictly. The procedure of the approach is presented mathematically in which the simulation clock advances in a sequence of activation points and

all concurrent activations are activated at a time, and associated agents respond in parallel. To reuse the code and make the development of the ABS with the PIW easy, in this section we will give a general idea that how to develop a framework for the ABS&PIW. We will design the individual agent and the time manager, and draw a static structure of the framework.

1.1 Individual Agent

The structure of the agent is shown in Figure 1. The agent is composed of attributes, an initialization method, a behavior controller, and a message handler.

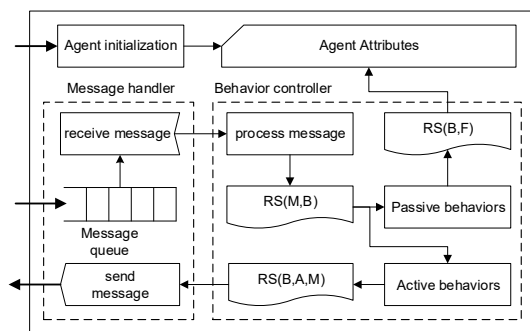


Figure 1: Structure of individual agent

Agent Initialization

Agent initialization, which is applied to set the values of attributes when an agent is created, is the only possibility to change the state of an agent directly by the external environment and enables the simulation to start in any state of the agent. There are also many other common methods (e.g., reset, start, and stop) for controlling agents, but they are not visible to the environment and called only by agents themselves.

Behavior Controller

The behavior controller decides which behavior to be executed when receiving a message. The decision is made depending on the relationships RS. Here we just summarize them in three types of relationships: RS (M, B) is the relationship between messages (in) and behaviors; RS (B, F) maps attributes to behaviors; RS (B, A, M) denotes the relationship between behaviors and messages (out). The behavior controller can also control behaviors, such as adding behavior, removing behavior, and changing behavior's state according to the environment.

Message Handler

Through the message handler, the agent communicates with other agents. A message handler includes one message queue and two methods: "send" and "receive." Messages from other agents will be stored in the message queue and received by "receive" method. Similarly, the agent can send the messages to other agents by using the method "send." These two methods are behaviors of the agent.

1.2 Time Manager

In order to keep consistent with the agent-based model, the time manager is also developed as an agent to be in charge of advancing time, activating agents, and managing activation points. The time manager extends the class of agents, and Figure 2 shows its structure.

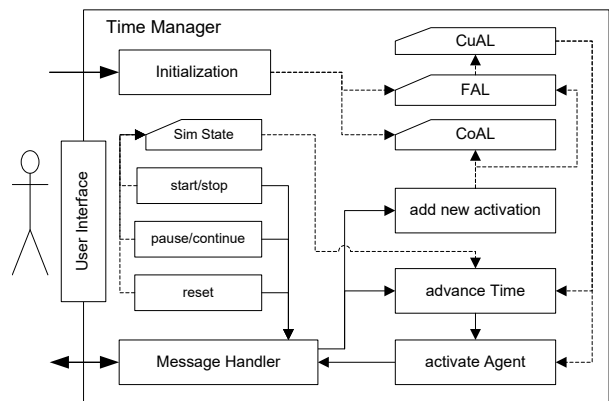


Figure 2: Structure of the time manager

Behavior of the Time Manager

There are three main behavior types: advance simulation clock, receive activation point and activate agent. A user interface is provided to control the simulation. Before the simulation runs, at least one activation point needs to be given in advance. Activation points are conveyed in the form of messages between the time manager and other agents. After the activation, the time manager is blocked until a new activation point is received or the ABM notifies it to advance the simulation clock when the model state is ready.

Activation Point Lists in the Time Manager

Three activation point lists are created in the time manager: conditional activation list (CoAL), future activation list (FAL) and current activation list (CuAL). The time

manager puts new received activation points into the appropriate list. The earliest activation points are moved from the future list to the current list every time the simulation clock advances. After activation, the current list is cleared. Conditional activation points are tried again and again and removed from the list when the conditions are met.

1.3 Agent-based Model

The agent-based model includes the agent environment, the agent manager, and a set of agents (shown in Figure 3). The agent environment is a medium for communication among the agents. The time manager and the agents send or receive activation points in the environment too. The agent manager is in charge of all agents and provides the model state to the time manager when the simulation is running. A new agent needs to register with the manager. The agents report their physical states and updated flags when they become blocked. The time manager also needs to register. In the model, each agent has a unique name which is used to specify the target agent in the communication.

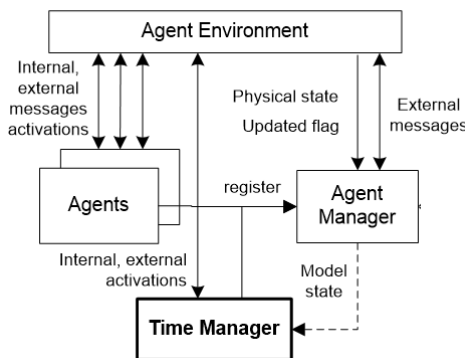


Figure 3: Structure of the Agent-based Model

2 Agents in the Job Shop Production

Agent-based modeling enables us to model the job shops more realistically and systematically comparing to other modeling methods, like discrete event simulation modeling, Petri nets, and so on. We know that an agent-based model is composed of agents and their relationships. Thus, we should determine who the agents are in the job shop production first. Obviously, material plays a leading role in the job shops. However, materials are usually transformed to other types of materials in shops, and their

lifecycle is very short. Paying attention to them makes no sense. We usually focus on jobs which organize all materials that one product needs in a serial of sequential operations. Besides, machines and transporters also make up a large proportion of the job shops. Because we assume that the transportation capacity is finite, the transporters are out of scope. The transportation in our study is treated as a delay. We only consider the machines here. As a type of flow, the material flow must have one or more sources and let jobs derive from it. The sources are usually job pools. Each type of job, i.e., product, is connected to a job pool. There is also a valve controlling the flow from each pool. In the job shops, this valve is a job release procedure. The release procedure is what we will concentrate rather than the job pools. To date on the basis of analysis above, we list three entities concerned in the job shops: job, machine group, and release procedure. We will model these three entities as agents in the agent-based model: release agent, job agent, and machine agent. In this section, we will describe how to create them on the basis of the framework, including their attribute and behavior definitions, communication and cooperation design, and simulation-related delay and activation design.

2.1 Release Agent

In a release agent, there are one piece of product data, multiple release policies, and one buffer. The release agent creates the job agents who are given in the product data based on a release policy. The buffer is located behind the release agent (see Figure 4). If the corresponding buffer is full in the first operations, the released job cannot start the operations. It will be blocked and stays in the release buffer until the buffer of the first operations has free space. If the buffer of the release agent is full, the release agent stops releasing until the buffer is not fully occupied.

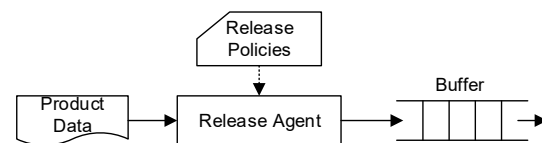


Figure 4: the release agent with one buffer

Product Data in the Release Agent

At the beginning of the simulation, the simulator reads product and process data from files and generates each product a release agent. Each product has a unique name and five data elements: a production probability, process,

priority, the release interarrival time, and the target work-in-process (WIP) level. The process element is the name of the process flow which is defined in the process file. Once the job agents are created, they will obtain their process flows from the product. The priority specifies the urgency of products. The value of the priority is from 0 to 1 and is used by some dispatch rules. The elements of interarrival time and WIP level are used by the release policies.

Release Policies of the Release Agent

The release policy is the criterion which decides when to release jobs. The release policies are a very common way to solve the job release problem. For now, the release agent has three very common types of release policies: constant interarrival time (CONINT), constant WIP level (CONWIP), and avoiding starvation (AS). CONINT releases the job in the same interval. To the policy of CONWIP, one interval time is still needed to be designated before reaching the target WIP. After reaching the target WIP, the policy of CONWIP takes effect. If the WIP level is less than the target WIP level, a new job is released. In the AS policy, a target buffer size is set for a bottleneck machine group. When the buffer size of the bottleneck is more than the target value, the releasing stops. When the buffer size is less than the target, new jobs are released. The quantity of the released jobs is the difference between the buffer size and the target value.

Communication with the Time Manager and Other Agents

If the release policy is CONINT, “releasing job” is an activation point which will be sent to the time manager and put in the activation list. When reaching the activation time, the time manager will send an activation message to the release agent. Receiving the activation message from the time manager, the release agent will be activated and start to release jobs. In the case of CONWIP, the finished job will send a message to the related release agent which will release a job immediately after receiving the message. For the AS policy, the machine agent of the bottleneck will request the release agent to stop releasing or ask the release agent to release jobs. If a released job is refused by the machine agent in the first operation due to the fullness of its buffer, the job sends blocked message to the release agent, and the release agent puts it into the buffer. When accepted, the job sends unblocked message

to the release agent. The job will be removed from the buffer and moved to the first operation.

2.2 Job Agent

The job agent is a temporary entity. After released, it will be processed on many machines in the order of its process flow which depends on the target product, and after finished it will be destroyed. The job agent has a unique name in the model and four logical states: transporting, waiting, processing, and blocking.

Behaviors and Lifecycle of the Job Agent

A job agent has five behaviors: to request resources, to be transported, to enter the buffer, to wait, to be processed, and to be blocked. The lifecycle of a job agent is shown in Figure 5. After release, the job requests the next operations. If accepted, it begins transporting and then enters the buffer to wait. If refused, it is blocked on the current machine. After receiving the start message from the machine group, the job starts. The job finishes when receiving the end message from the machine group and then requests the next operation. If it is the last operation, the job completes.

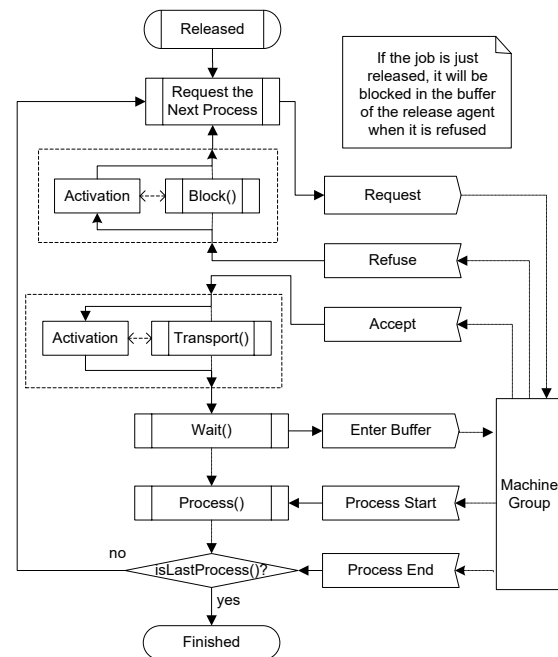


Figure 5: Lifecycle of a job agent

2.3 Delays and Activations in the Job Agent

There are four types of delays related to the logical state: transporting, blocking, waiting and processing. Blocking

delay means that a finished job cannot move to the next operation due to the fullness of the related buffer and will continue to stay on the current machine. This is a conditional delay, and it will be activated by the time manager every time the simulation time advances. When a job begins to be transported or is blocked, associated activation points (transporting end, blocking end) will be sent to the time manager. After that, the job agent will wait for activations from the time manager and the delay will end when it receives the activation messages. The waiting delay and the processing delay end when the job receives the related message from a machine.

2.4 Machine Group Agent

The machine group agent is a permanent and active entity. It includes one buffer and several machines (see Figure 6). The machines have the same function and share the buffer.

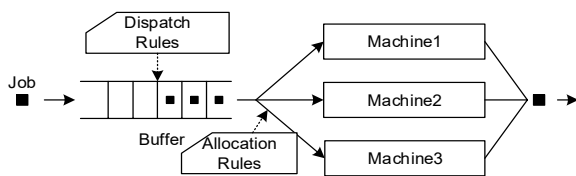


Figure 6: Structure of the machine group

There are five logical states of the machines: idle, busy, setup, breakdown, and maintenance. The machines may be single processing machines or batch processing machines. When a job arrives, and cannot be processed at once, the job joins the buffer. The buffer has a finite capacity, and it dispatches the waiting job to the idle machine according to a dispatch rule. There is no buffer behind the machine group. When a job finishes, the job will either be transported to the next machine group or be blocked on the current machine.

Behaviors of Machine Group Agent

The machine group agent has two behaviors: to respond to the request from the job agent and to select the best machine to process the job. The buffer has two behaviors: to store the job and to dispatch the stored jobs to the machines in a certain batch and the order of the given priority. The machines have five behaviors: to request the jobs from the buffer, to setup, to process, to interrupt, and to recover.

When a job enters a buffer or a machine just finishes one job, the machine group will decide on the start of a

new process. If the buffer is not empty (in case of batch processing, a batch must be ready) while the machine group has an idle machine, the idle machine will start processing and inform related job agents. If there is more than one idle machine, one machine will be selected according to an allocation rule. If the setup is needed, the machines will start the process after the setup. Figure 7 shows the behavior flow of the machine group agents.

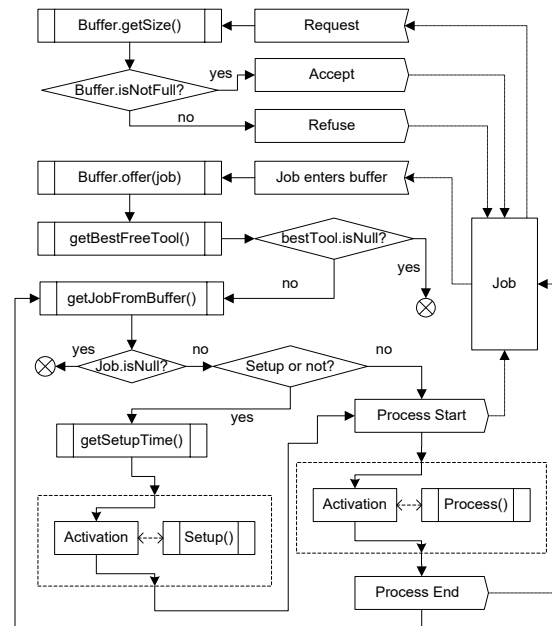


Figure 7: Behavior flow of a machine group agent

Dispatch and Allocation Rules in Machine Group Agent

A dispatch rule is a criterion for determining the jobs' process priority in the buffer. An allocation rule is for selecting one machine from the idle machines. The dispatch and allocation rules are very common ways to solve the sequencing and routing problems. A buffer may include a set of dispatch rules. When a job joins in the buffer, all messages associated with dispatch rules are sent to the buffer. Based on these messages and under a given dispatch rule, the buffer determines the priority of each job and queues them in that sequence. Once a job joins the buffer, the priority is updated. Currently, 17 types of common dispatch rules, such as FIFO (First In First Out), EDD (Earliest Due Date), CR (Critical Ratio), etc., have been preset in the buffer. The allocation rules are used by the group agent. Once a job comes out from the buffer, the group agent will collect information of all machines

and select one to process the job.

Delays and Activations in Machine Group Agent

There are four types of delays related to the logical state in the machine group agents: setup, processing, breakdown and preventive maintenance. When the setup delay occurs, the machine group agent sends an activation message (setup end) to the time manager and informs related jobs. The jobs will wait and cannot be processed by other machines. When the setup delay ends, the machine group

agent sends an activation (process end) message to the time manager. Meanwhile, it informs the jobs and starts processing. Activated by the time manager, the machine finishes processing and informs the jobs.

2.5 Communication among the Agents

We summarize the communication in Figure 8. It includes the delays and activations conveying between the time manager and the agents as well as the communication among the agents.

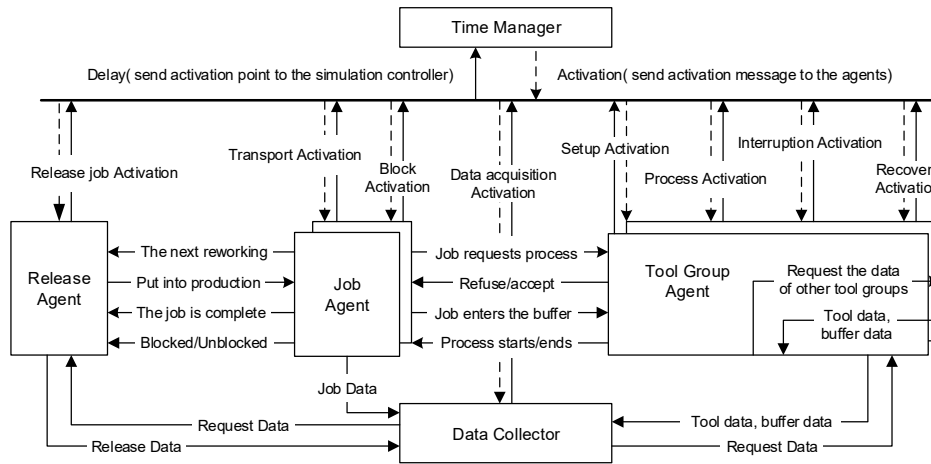


Figure 8: Communications among the agents and the time manager

3 Agent-based Simulation of the Job Shop Production

3.1 Overview of the Simulation

The agent-based simulation of the job shop production consists of a simulator, production data, and performance measures (see Figure 9). The production data are inputs, and the performance measures are outputs.

Production Data

We classify the production data into three parts: machine group data, product data, and process data, and then adopt XML to store the data. Thus, three XML files are created to store the data. The machine group file includes data about all machine groups, e.g., the number of machines, buffer size, dispatching rule, etc. The process file consists of the process flows of all products. Each processing flows has many operations. The product file consists of buffer size, waiting time (by machine group), blocking

data about all products which will be produced. Each product has a name of the process flow which is defined in the process file. The process file links the product file and the machine group file together.

Simulator for the Job Shop Production

The agent-based model (including release agents, machine group agents, and job agents), data collector and time manager make up the simulator for the job shop production. The time manager creates the release agents and the machine group agents according to the machine group data at the beginning of the simulation, and it is responsible for advancing the simulation time and handling the simulation control (e.g., start, stop, pause, etc.). The data collector is responsible for collecting simulation data and computing the performance measures. It can collect all simulation data in detail, as well as part of sample data.

Performance Measures

The performance measures include WIP level, cycle time,

time (by machine group), and machine utilization information (e.g., idle time, processing time, breakdown time and setup time). All data can be shown in graphs and tables. These measures can be used to improve the job

shop production and can be provided for the optimization or control algorithms to achieve the goal of the optimization and control.

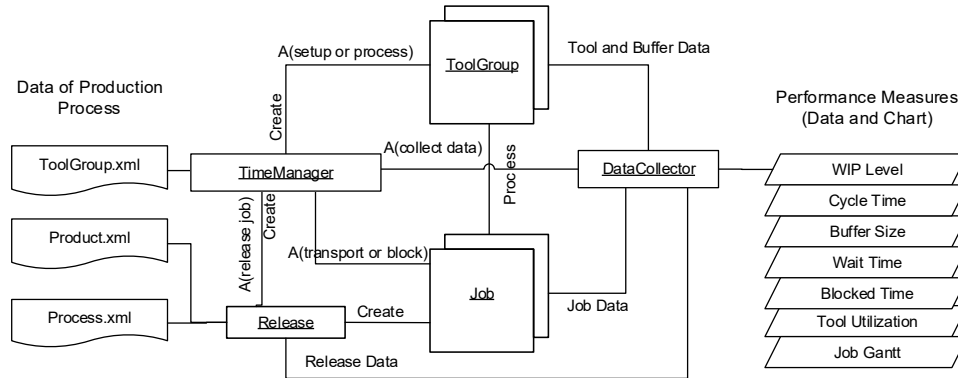


Figure 9: Agent-based simulation of the job shop production

3.2 Static Structure of the Simulator

The static structure of the simulator (see Figure 10) has three layers: framework layer, agent layer, and production characteristics layer.

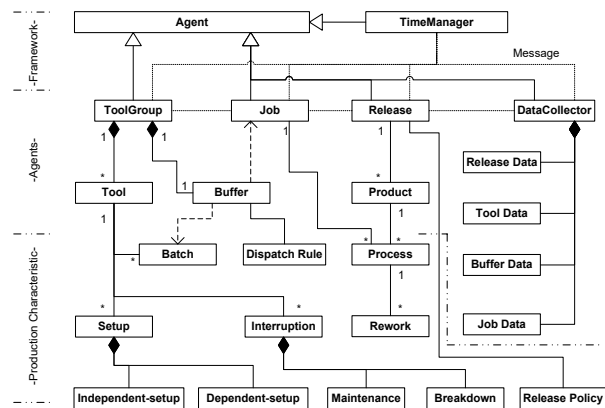


Figure 10: Static structure of simulator

The framework layer provides the agent base class and the time manager. The agent layer contains the agents which are abstracted from the job shops, and these agents extend the base class of the agent in the framework. A machine group agent has multiple machines and one buffer. Each release agent can release one kind of product. The data collector, which is also an agent and in charge of data collection from machines, buffers, jobs, and release agents, is contained in the agent layer. In the production characteristics layer, many characteristics are considered, such as reentrant flows, rework, setups, batch processing, breakdowns, and preventive maintenance. Dispatch rules and release policies are part of this layer.

4 Applications

We create a simulation model of an example job shop.

The job shop contains five machines and produces two products (Pa and Pb) with two process flows and two different throughputs. The throughputs of product Pa and Pb are 110 and 197 units per week. There are no batch processing machines. The machines need sequence-dependent setups. The interval between two breakdowns on the machines is subject to the exponential distribution and the repairing time follows an exponential distribution too.

The simulator is used to solve real-time sequencing problems in the job shop in which we should decide which job should be processed first once a machine becomes available. The objective is to minimize the cycle time. The sequencing problems are sequential decision-making problems. The decision-making is the selection of the best job from possible jobs. In other words, it is a priority calculation problem. For each job, a priority value is computed. The job with the highest priority is selected. We proposed three simulation-based approaches to calculate the priority values of jobs, including the simulation try-then-decide method (STTD)[8], the intelligent method based on the simulation try-then-decide method (INT1)[7], and the intelligent method based on Markov decision process (INT2)[9].

The STTD method is a pure simulation approach. It uses the alternative simulations to predict the future after a job is taken and select jobs according to the future in-

formation from the simulation. The most important innovation is the usage of the base-rule in the alternative simulations. The base-rule avoids the exponential explosion of the number of the alternative simulations. To evaluate the STTD method, we replace the system with an environment simulation. The decisions we made are executed in the environment simulation. The results from the environment simulation can be used to evaluate the STTD method.

The INT1 method combines the experiences & data approach with the simulation approach. A data-driven model is introduced to calculate the priority values for jobs. It is built on the data from the simulation with the STTD method. So, it manages to learn the knowledge of the STTD method. Two types of factor influence the priority value of the job: global factors and local factors. The state of the job shops is divided into several patterns by clustering the global data. In each pattern, the priority value is only up to the local factors. The relationship between the priority and the local factors is mapped in the neural networks. For each pattern, one neural network is created. In the decision maker, the centroids of the patterns make up a pattern pool, and the neural networks make up a network pool. While making the decision, the decision maker determines the pattern of the current state according to the pattern pool and selects one corresponding neural network from the network pool. The neural

network will calculate the priority for each job according to the local factors.

The INT2 method combines the experiences & data approach, mathematical approach, and simulation approach. It models the sequencing as Markov decision process with multiple decision makers. We defined the five-tuple for the problems, including decision points, state space, action sets, transition procedure, and a reward function. The data-driven model is still used to map the value of the action to the state-action pair. The simulation-based batch-mode Q-learning algorithm explores the state space by the simulation. Each simulation run is one iteration. The data-driven model is updated and improved gradually after each iteration.

The three methods compare with each other as well as other two decision rules, First In First Out (FIFO) and Shortest Processing Time(SPT). The experiment results are shown in Table 1. The results show that the STTD and INT1 methods always outperform the rules. The STTD method performs best but consumes much time. Contrarily, the INT1 method takes less time while the performance is just a little bit worse than the STTD. Thus, the STTD is more suitable for offline applications, and the INT1 can be used in the real-time control. Unfortunately, the INT2 method performs unsteadily. It will be further studied in the future.

Approach \ Items	FIFO	SPT	STTD	INT1	INT2	
Cycle Time /Hour	Pa	4.09	3.64	3.53	3.58	3.73
	Pb	4.78	4.91	4.49	4.56	4.88
	Sum	4.52	4.46	4.12	4.21	4.46
Run Time (s)	1.0	1.3	86.8	1.4	1.5	

Table 1: Comparison between the proposed approach and some decision rules.

5 Conclusions

On the basis of the previous works, a framework for the ABS is designed and an agent-based model of the job shop production including release agents, machine group agents, and job agents are built. The behaviors and interactions among the agents are explicit defined. A large variety of decision rules are preset in the model as well. The agents can make decisions simply according to the rules. The agent-based model, the data collector, and the time

manager make up the simulator for the job shop production. Applying the simulator to an example job shop, we solved the real-time sequencing problem by three simulation-based approaches we have proposed. The application and experiment results indicate that the overall idea of simulation-based learning and decision-making is feasible in the job shops.

References

- [1] Lasi, H., et al., *Industry 4.0*. Business & Information Systems Engineering, 2014. 6(4): p. 239-242.
- [2] Jazdi, N. *Cyber physical systems in the context of Industry 4.0*. in *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*. 2014.
- [3] Wortmann, F. and K. Flüchter, *Internet of Things*. Business & Information Systems Engineering, 2015. 57(3): p. 221-224.
- [4] Hermann, M., T. Pentek, and B. Otto. *Design Principles for Industrie 4.0 Scenarios*. in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. 2016.
- [5] Xie, S., T. Zhang, and O. Rose, *Agent-based Simulation with Process-interaction Worldview*, in *ASIM 2018 24. Symposium Simulationstechnik*. 2018: Hamburg.
- [6] Zhang, T. and O. Rose, *A framework for agent-oriented parallel simulation of discrete event systems*, in *Proceedings of the Winter Simulation Conference*. 2012, Winter Simulation Conference: Berlin, Germany. p. 1-2.
- [7] Zhang, T. and O. Rose. *Intelligent dispatching in dynamic stochastic job shops*. in *2013 Winter Simulations Conference (WSC)*. 2013.
- [8] Zhang, T. and O. Rose, *Simulation-based Dispatching in Job Shops*, in *ASIM 2014 22. Symposium Simulationstechnik*. 2014: Berlin.
- [9] Zhang, T., S. Xie, and O. Rose. *Real-time job shop scheduling based on simulation and Markov decision processes*. in *2017 Winter Simulation Conference (WSC)*. 2017.

Agentenbasierte Modellierung der Schlaganfallversorgung in Österreich

Dominik Rothschedl^{1*}, Jakob Rosenberger^{1**}, Martin Bicher¹, Niki Popper²

¹Institute for Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *dominik.rothschedl@tuwien.ac.at **jakob.rosenberger@tuwien.ac.at

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

Abstract. Vor allem durch die stetig alternde Bevölkerung ist Schlaganfall in den letzten Jahrzehnten auf Platz 2 der Liste der häufigsten Todesursachen geklettert. Umso wichtiger ist also die Erforschung dieser Krankheit um nachhaltige Verbesserungen der damit verbundenen Versorgungslandschaft zu ermöglichen. Im Rahmen des österreichischen Forschungsprojekts DEXHELPP (Decision Support for Health Policy and Planning) haben wir uns dieser Fragestellung unter Anderem mithilfe eines Simulationsprojektes angenommen, welches mittels agentenbasierter Modellierung das Auftreten von Schlaganfällen und deren Auswirkungen auf eine Bevölkerung so realitätsnahe wie möglich modelliert. Die Ergebnisse werden anschließend in ein spezielles Programm eingelesen und graphisch aufgearbeitet, um das Verhalten der Population zu analysieren. Auf Basis dieses einzigartigen Modells kann man interessante Einblicke gewinnen, in dem man gewisse "Was wäre wenn..."-Fragen analysiert und eine passende Simulation durchführt.

Einleitung

Vor allem durch die stetig alternde Bevölkerung ist Schlaganfall in den letzten Jahrzehnten auf Platz 2 der Liste der häufigsten Todesursachen geklettert [1]. In Österreich sind 80% aller Schlaganfallpatienten über 60 Jahre alt [2] und ein Drittel aller Todesfälle gehen auf diesen Ausfall der Nervenzellen im Hirn zurück [3]. Umso wichtiger ist also die Erforschung dieser Krankheit um nachhaltige Verbesserungen der damit verbundenen Versorgungslandschaft zu ermöglichen. Im Rahmen des österreichischen Forschungsprojekts DEXHELPP (Decision Support for Health Policy and Planning) haben wir uns

dieser Fragestellung unter Anderem mithilfe eines Simulationsprojektes angenommen. Ziel dieses Projektes war es, anhand einer modellierten Bevölkerung das Auftreten von Schlaganfällen und deren Eigenschaften so realitätsnahe wie möglich zu simulieren. Das Grundgerüst dieser Arbeit ist ein bereits validiertes agentenbasiertes Modell der österreichischen Bevölkerung, das sich mittels Daten der Statistikagentur Statistik Austria wie eine echte Bevölkerung verhält. Das heißt, die Personen dieser Bevölkerung können gewissen Ereignissen unterliegen, zum Beispiel Todesfällen, Geburten oder Immigration. Wir werden dieses Modell in Kapitel 2 noch genauer beschreiben. Die erste Aufgabe bestand darin, dieses Modell zu modifizieren, um weitere Modellprozesse zu inkludieren, nämlich verschiedene Arten und Schweregrade von Schlaganfällen. Als Nächstes mussten erhebliche Erweiterungen des Programmes durchgeführt und eine Vielzahl an neuen Attributen, sowie Features hinzugefügt werden. Dabei sollte es sich um Eigenschaften wie in etwa Regionalität, Speicherung von medizinischen Daten und genetisch bedingten Vererbungen handeln. Anhand dieser Veränderungen sollte das Modell möglichst realitätstreu validiert werden, indem man die Wahrscheinlichkeiten dieser Ereignisse von medizinischer Vorgeschichte, Regionalität und anderen Merkmalen in einer bestimmten Weise abhängen lässt.

Um das Modell schließlich verwenden zu können, sollten die in einer Simulation entstandenen Resultate in einem speziellen Format ausgegeben werden, damit man die gewonnenen Simulationsergebnisse möglichst flexibel grafisch aufarbeiten kann. Dabei handelt es sich um eine csv Datei, in welcher jede Zeile einem Schlaganfall entspricht und jede Spalte weitere Informationen über diese Ereignisse beinhaltet. Diese Datei wurde anschließend in dem von der Firma VRVis zur Verfügung gestellten Programm Visplore [4] eingelesen und

graphisch aufgearbeitet, um das Verhalten der Population zu analysieren.

1 Agentenbasierte Modellierung

Die agentenbasierte Modellierung (Agent-based modelling) ist eine mächtige Modellbildungsstrategie, die in den letzten Jahren in vielen Anwendungsgebieten verwendet wurde. Die Methodik dieses Modellparadigmas ist eine sehr einfache: Das Modell wird aufgebaut aus vielen Agenten, also kleinen Einheiten, die mit der Umwelt interagieren. Jeder Agent hat bestimmte Eigenschaften und eigene Entscheidungsmöglichkeiten, die gewisse Handlungen der Individuen nach sich ziehen. Dadurch bekommt das Modell eine ganz spezielle Dynamik, die durch richtige Analyse sehr informativ sein kann. Dabei untersucht man unter anderem Emergenz, also Effekte des Systems die durch Aktionen der Agenten entstehen, die aber nicht direkt durch die Algorithmen der einzelnen Einheiten vorhersehbar sind [5]. Man erkennt ein essentielles Konzept, auf dem das Modell basiert: Das Mikro-Verhalten der Individuen bestimmt das Makro-Verhalten des Systems.

Für gewisse Fragestellungen ist agentenbasierte Modellierung die optimale Lösung. Man kann für gewisse Prozesse Szenarien-Analysen erstellen oder Prognosen abgeben und verschiedenste Verhaltensmuster beobachten. Zu den wichtigsten Anwendungsfeldern gehören heute zum Beispiel (siehe [5]):

- Gesundheitswesen
- Makroökonomie
- Finanz- und Energiemärkte
- Markt- und Strategiesimulation
- Ökologische Systeme und Umwelt
- Sozialwissenschaften
- Stadt- und Populationsdynamik

2 Generic Population Concept

Das Generic Population Concept (GEPOC) ist ein im Rahmen des Forschungsprojekt DEXHELPP entwickeltes agentenbasiertes Populationsmodell. Dieses kann

man in vielerlei Hinsicht nutzen, da man darauf etliche Projekte, die man für die Forschung in der Medizin und in vielen anderen Bereichen verwendet, aufbauen kann[6]. GEPOC ist ein sehr umfangreiches Programm, das in der Lage ist, eine virtuelle österreichische Bevölkerung zu erschaffen und diese über nahezu beliebige Zeitspannen hinweg zu simulieren. Die für unsere Arbeit als Grundlage verwendete Implementierung des Modells ist in Python3 umgesetzt. Da die objektorientierte Implementierung des Modells die Modelldefinition sehr gut widerspiegelt, wollen wir das Modell anhand der wichtigsten Programmelemente von GEPOC ein wenig genauer erklären.

2.1 Population



Abbildung 1: Klassendiagramm eines Agenten.

Funktionsparameter wurden mit params angedeutet, da diese Grafik nur zur Visualisierung dient.

Wenn man mit einem agentenbasierten Modell zu tun hat, fragt man sich natürlich zuerst, was dabei die Agenten sind. Im Projekt GEPOC sind sehr offensichtlich einzelne Personen beziehungsweise, je nach Maßstab, entsprechende Personencluster der Bevölkerung die Agenten. Diese sind Objekte einer Klasse, die spezielle Eigenschaften und Methoden besitzen, welche Geburtstag, Index, Mutter, Geschlecht, Kinder, ein Boolean ob die Person zugewandert ist, und zu guter Letzt, die Routine Protocol, die ein Agent ausführt wenn er über seinen Status befragt wird. Letztere spezifiziert welche Ereignisse und Variablen im Modell überhaupt dokumentiert werden und schlussendlich im Simulationsoutput zu sehen sind.

2.2 Simulation

Das Herz von GEPOC ist klarerweise jene Routine, die sich um die Populationsdynamik kümmert. Das Modell arbeitet in diskreten Zeitschritten. Zum einfacheren Verständnis betrachten wir das Flussdiagramm für die Berechnung der Ereignisse, welches in Abbildung 2 und Abbildung 3 zu sehen ist.

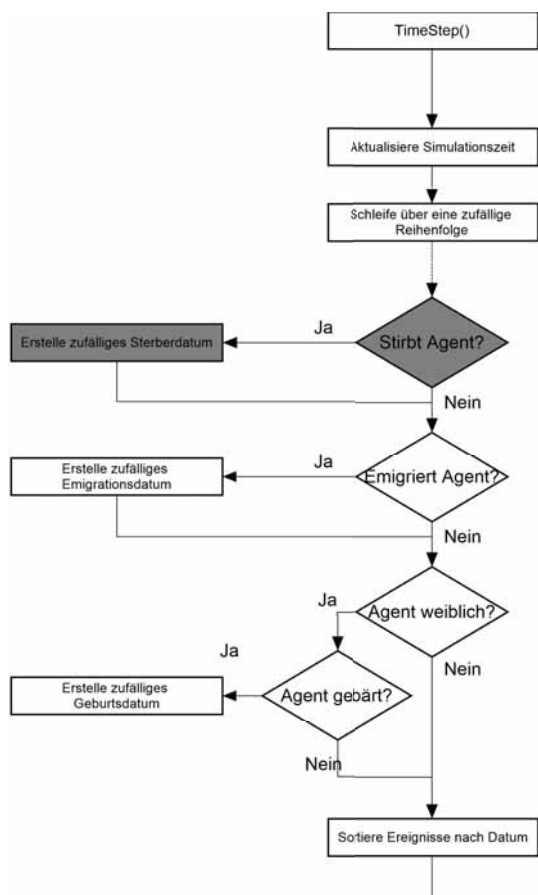


Abbildung 2: Zeitupdate im GEPOC Modell als Blockdiagramm. Erste Phase: Erstellen der Ereignisliste.

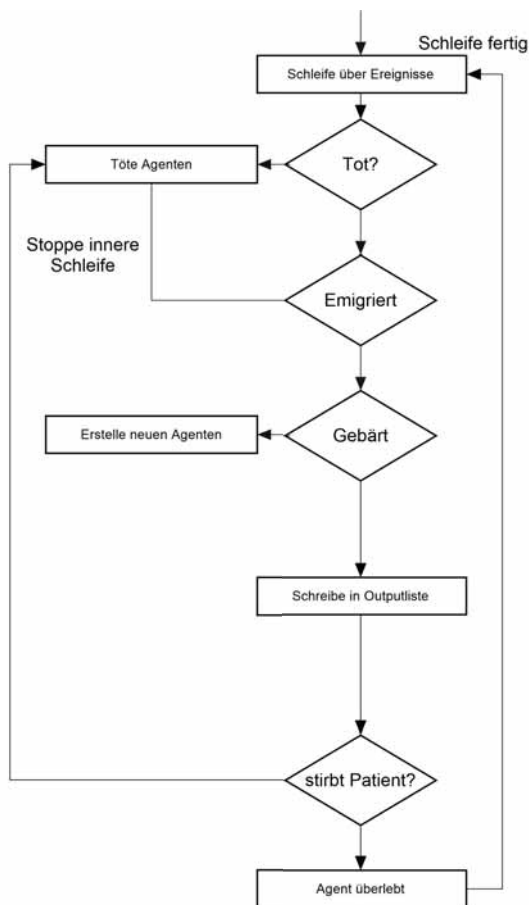


Abbildung 3: Zeitupdate im GEPOC Modell als Blockdiagramm (Fortsetzung von Abbildung 2). Zweite Phase: Auswirkungen auf das System

Das Konzept des Updates der Agenten in GEPOC kann man in zwei Phasen gliedern, wobei die erste Phase in Abbildung 2 zu sehen ist (auf den farblich markierten Bereich werden wir später in Kapitel 2.3 noch näher eingehen). Zuerst wird das Datum am Ende des Timesteps berechnet, in unserem Fall war das Startdatum der 01.01.2006 und ein Schritt dauert ein Jahr. Somit ist das Datum mit dem wir im ersten Durchlauf der Funktion rechnen, der 01.01.2007.

Da wir unsere Agenten in einer Liste speichern und es Laufzeitverschwendung wäre, alle Personen in jedem Timestep zu betrachten, wird diese Liste nochmals initialisiert. Doch dieses Mal werden alle Agenten, die verstarben oder emigrierten, nicht mehr berücksichtigt. Nun werden die Ereignisse dieser Zeitspanne für die Agenten in zufälliger Reihenfolge berechnet. Hierfür

gibt es die sogenannten *randomly_decide_to_...* Funktionen, welche mittels geeigneter Parameter entscheiden ob das jeweilige Ereignis eintritt. Falls eine Aktion eintritt, wird ein zufälliges Datum vor dem aktuellen Simulationsdatum berechnet, welches innerhalb des Zeitschrittes liegt. Diese Daten werden in eine Liste gespeichert. In Phase 2 (Abbildung 3) werden dann die geplanten Ereignisse die nun in der Liste gespeichert sind in der kausal-korrekten (sortiert nach dem Ereignisdatum) Reihenfolge abgearbeitet. Dadurch werden bei den Agenten, je nach Ereignis, Änderungen vorgenommen, welche für den nächsten Zeitschritt Veränderungen für diesen nach sich zieht. Außerdem werden alle Verstorbenen und Emigrierten bei den nächsten Durchläufen nicht mehr beachtet.

2.3 Exitus

Wir haben jetzt also schon zumindest den groben Ablauf des GEPOC Modells beleuchtet. In diesem Unterkapitel wollen wir aber noch genauer auf den Prozess eingehen, in dem entschieden wird, ob einer der Agenten stirbt oder überlebt, also den blau markierten Abschnitt in Abbildung 2. Es wird die Klasse *Deaths* definiert, welche genau diesen Prozess bewältigt. Dieses Skript kommuniziert mit dem Parameter *Files*, somit kann man darin auf die benötigten Daten zugreifen, nämlich die Wahrscheinlichkeiten für Tode. Diese werden mittels einer Funktion richtig für einen Zeitschritt skaliert. Jetzt kommt die Funktion *randomly_decide_to_die* ins Spiel, welche mithilfe eines Zufallszahlengenerators entscheidet, ob der Agent im beobachteten Zeitschritt stirbt oder nicht.

Die anderen *randomly_decide_to_...* Funktionen für Emigration und Geburten sind auf die selbe Art und Weise implementiert, deshalb werden wir sie nicht im Detail erklären.

3 GEPOC-Stroke

In den vorherigen Kapiteln wurde das ursprüngliche GEPOC-Modell erklärt, welches nun zu einem realitätsnahen Schlaganfallmodell umgebaut wird, welches wir in Folge als GEPOC-Stroke bezeichnen wollen. Genauer gesagt wollen wir die Schlaganfälle der gesamten österreichischen Bevölkerung und deren Auswirkungen auf diese über einen beliebigen Zeitraum simulieren, graphisch auswerten und analysieren. In den nächsten Abschnitten werden diese

Erweiterungen und die dazu nötigen Implementierungen genauer erklärt.

3.1 Agenten

Die Klasse für unsere Agenten samt Eigenschaften und Methoden wurde im Kapitel GEPOC (2.1) schon diskutiert. Für unsere Erweiterung des Programmes fordern wir allerdings noch mehr Eigenschaften bei unseren Agenten. Da wir in GEPOC-Stroke Schlaganfälle betrachten, ist es naheliegend, Informationen darüber für jedes Individuum der virtuellen Bevölkerung abzuspeichern. Manche dieser neuen Variablen sind, wie wir im nächsten Kapitel sehen werden, nur für den Output notwendig, einige sind jedoch für die Berechnungen der zukünftigen Ereignisse sehr wichtig:

<i>firststroke</i>	Datum des ersten Schlaganfall
<i>firstdiag</i>	Diagnose des ersten Schlaganfall
<i>cntr</i>	counter für erlittene Schlaganfälle
<i>dos</i>	Schweregrad des Schlaganfalles
<i>factor_of_dos</i>	Summe aller bisherigen Schweregrade

Table 1: Beschreibung der neuen Variablen (Erweiterung zur Abbildung 1)

Auch die Bezirke von Österreich werden alle in unserem Modell verwendet und der Wohnsitz der virtuellen Bewohner wird in *district* unterschieden. Jeden Agenten wird mit einer Funktion stochastisch ein Bezirk zugeordnet, der in *district* vermerkt wird. Dafür wurden die absoluten Bevölkerungszahlen von den Bezirken vom 1.1.2017 aus öffentlichen Daten von Statistik Austria verwendet, um die Wahrscheinlichkeiten zu berechnen. Nach dem ein Schlaganfall passiert, wird die betroffene Person zu einem Patienten, bekommt also einen Patientenindex und ein Entlassungsdatum zugewiesen. Um die Anzahl der Aufenthaltstage zu bestimmen verwenden wir folgende Funktion:

$$days = \lfloor age \cdot dos \cdot 0.1 \rfloor + 2 \cdot factor_of_dos \quad (1)$$

3.2 Parametrisierung

Um die GEPOC-Stroke zu parametrisieren wurden zwar vorwiegend geschätzte Parameterwerte verwendet, Teilaspekte des Modells konnten jedoch mit Real-

daten abgestimmt werden. Die Realdaten der Schlaganfälle in Form einer Liste mit einzelnen Schlaganfällen und Informationen darüber wie Alter, Geschlecht, etc. wurden uns von der Gesundheit Österreich GmbH zur Verfügung gestellt. Die Herausforderung hierbei war, anhand dieser Daten eine Tabelle zu generieren, welche uns die Wahrscheinlichkeit eines Schlaganfalls jeder Altersgruppe pro Jahr angibt. Wir haben uns für dieses Format entschieden, da sowohl die Tabellen für die Population (welche wir für die Berechnung verwenden), als auch die der Sterbewahrscheinlichkeit dieses besitzen und somit ein einheitliches Format der Tabellen beibehalten wird, was zu einem größeren Verständnis für Dritte führt. Für die Berechnung werden die Daten eingelesen und jeder Schlaganfall dem richtigen Jahr, Geschlecht und dem entsprechenden Alter zugeordnet. Die anspruchsvollste Aufgabe war der Block für die Wahrscheinlichkeitsberechnung. Zu diesem Zeitpunkt haben wir zwei Matrizen (männlich und weiblich) deren Spalten die Jahre und deren Zeilen die Altersgruppen widerspiegeln. Jeder Eintrag gibt somit die absolute Anzahl der Schlaganfälle pro Jahr und Alter an. Die erste Vermutung legt Nahe, dass man die Einträge durch die Anzahl der Bevölkerung des entsprechenden Jahres und Alters dividiert. Doch da die Geburtsjahre der Schlaganfallpatienten in Österreich in den realen Daten durch 5 dividiert wurden, um die Anonymisierung der hoch sensitiven Daten sicherzustellen, sind die Matrizen schwach besetzt. Somit wäre die Wahrscheinlichkeit für die meisten Altersgruppen 0.

Deshalb werden die Ergebnisse jeder Altersgruppe addiert und dann durch die Anzahl der Einträge, die das Ergebnis nicht verfälschen dividiert. Auf diese Zahl wird wiederum durch die absolute Bevölkerungsanzahl der Altersgruppe dividiert.

3.3 Wann tritt ein Schlaganfall im Modell auf?

Für die Entscheidung ob es zu einem Schlaganfall kommt werden die Daten herangezogen, die im Kapitel 3.2 beschrieben wurden, und des Weiteren mittels der Vorgeschichte des Patienten die aktuelle Wahrscheinlichkeit für einen Hirnschlag berechnet. In Abbildung 4 erkennt man farbige markiert, welche Modellabläufe im Zuge der Schlaganfall Erweiterung des Modells zu jenen aus Abbildung 2 ergänzt wurden.

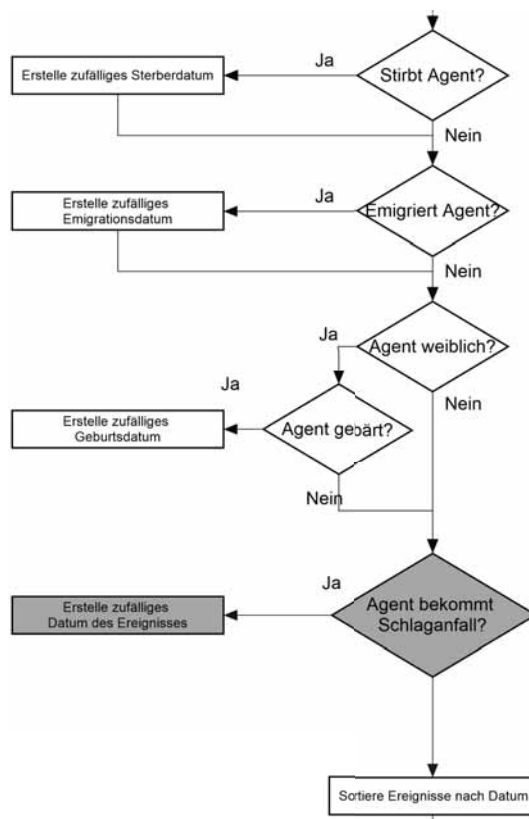


Abbildung 4: Erweiterung der Abbildung 2 auf GEPOC-Stroke Modell Abläufe

Nun beschäftigen wir uns mit der Implementierung für diese Abläufe, mit welcher wir das GEPOC Modell zu einem Schlaganfall-Modell erweitert haben. Die Funktionsweise ist nichts anderes wie bei der Funktion *randomly_decide_to_die*, die in 2.3 erklärt wurde. Nur mussten wir die Wahrscheinlichkeit anpassen, die von deutlich mehr Parametern abhängt als die Wahrscheinlichkeit zu sterben. Ein entscheidender Faktor dabei ist der medizinische Vorverlauf, der in der Erweiterung des Grundmodelles neu ist und bei jedem Agenten (siehe 3.1) vermerkt wird. Wir nehmen in unserem Modell an, dass die Wahrscheinlichkeit einen weiteren Schlaganfall zu erleiden um 10 – 20% pro vorangegangenen Hirnschlag höher ist [7, 8]. Um in unserer Implementierung zu erreichen, dass diese Erhöhungen auch zwischen 10 – 20% variieren, lassen wir den Koeffizienten, mit dem wir die Wahrscheinlichkeit multiplizieren, vom Schweregrad (3.4) abhängen. Dafür bekommt jedes der fünf Stadien einen Wert zugewiesen, das erste und leichteste Stadium bekommt 1, das letzte den Wert 5.

Mit

$$1.075 + dos \cdot 0.025 \quad (2)$$

$$dos \in \{1, \dots, 5\}$$

erhalten wir einen Faktor, der die Wahrscheinlichkeit um die gewünschten Prozent erhöht. Deshalb wird vor dem Abprüfen, ob eine zufällig erstellte Zahl zwischen 0 und 1 kleiner ist, die Wahrscheinlichkeit mit folgendem Verfahren vergrößert: Falls für ein Schlaganfallopfer genau ein vergangener Schlaganfall vorliegt, dann wird die Wahrscheinlichkeit mit der Formel

$$p = (1.075 + dos \cdot 0.025)p \quad (3)$$

erhöht. Falls schon mehrere Schlaganfälle im medizinischen Vorverlauf vermerkt wurden, wird der gerundete Durchschnitt der Schweregrade genommen, was zu folgender Änderung der obigen Formel führt:

$$p = \left(1.075 + \left[\frac{factor_of_dos - 1}{cntnr} \right] \cdot 0.025 \right)^{cntnr} \cdot p \quad (4)$$

Wobei *factor_of_dos* und *cntnr* Variablen von jedem einzelnen Agent sind (siehe 3.1). Die Formel liefert zwar die gewünschte Veränderung der Wahrscheinlichkeit, doch damit könnten wir Wahrscheinlichkeiten höher als 1 erhalten, was nicht der Realität entspricht. Deshalb werden wir dieses Verfahren nur bis zu einem gewissen Grenzwert anwenden, und danach mit einer anderen Funktion die abgeschwächte Steigerung der Wahrscheinlichkeiten erzielen.

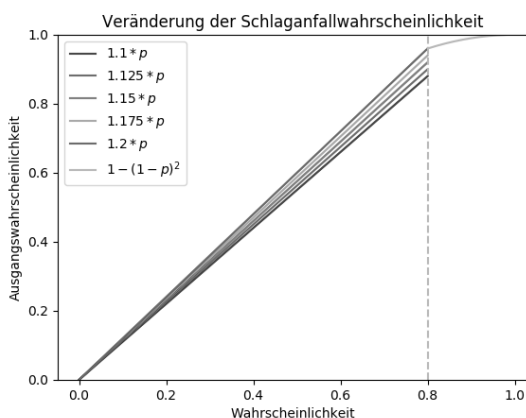


Abbildung 5: Verhalten der Funktion für die Erhöhung der Schlaganfallwahrscheinlichkeit

In den Abbildungen 5 ,6 wenden wir uns der Funktion zu, die wir für den erwähnten Prozess dafür verwenden. Da wir nun definiert haben, wann ein Schlaganfall im

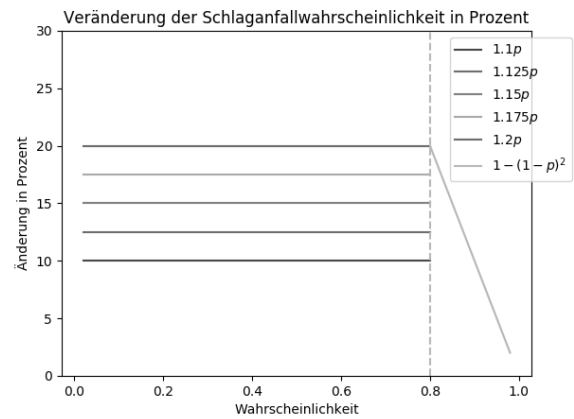


Abbildung 6: Änderung der Wahrscheinlichkeiten in Prozent

Modell eintritt, beschäftigen wir uns nun mit der Art des Hirnschlages und dessen Auswirkungen. Betrachten wir zunächst wieder die Abläufe der Abbildung 3. In Abbildung 7 sind die Erweiterungen abermals rot gefärbt.

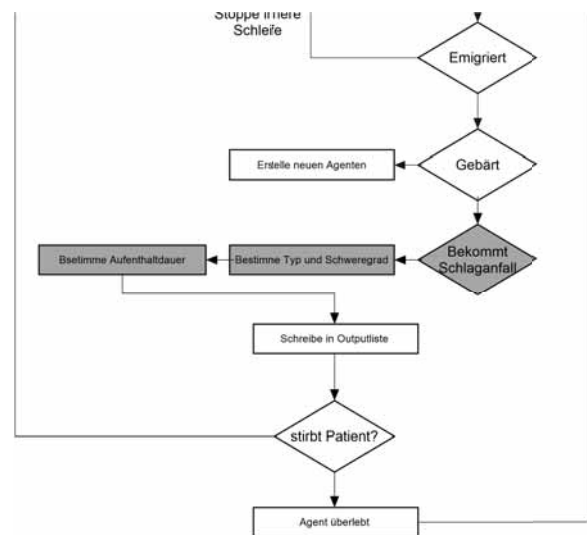


Abbildung 7: Bestimmung des Typs und Auswirkungen auf das System

3.4 Bestimmung des Typs und Schweregrad

Die Bestimmung des Typs erfolgt mittels einer Zufallszahl und den Wahrscheinlichkeiten, die die Realdaten liefern. Die Verteilung der Typen eines Schlaganfalles sehen nach der ICD-10 Klassifikation[9] wie folgt aus:

- *G45, Zerebrale transitorische Ischämie und verwandte Syndrome*
- *I60, Subarachnoidalblutung*: Wird auch Spinnenhaut genannt und durch Eintreten von Blut in den mit Hirnflüssigkeit gefüllten Subarachnoidalraum hervorgerufen.
- *I61, Intrazerebrale Blutung*: Ist ähnlich zu I60, wird aber unter anderem durch Bluthochdruck, Veränderung von kleinen Arterien und Gefäßmalformationen hervorgerufen
- *I63, Ischämischer Schlaganfall*: Hiermit ist ein plötzlicher Sauerstoffmangel der Nervenzellen im Gehirn, welcher durch Verengung oder Verschluss der Arterien entsteht, gemeint.

Um die Verteilung der einzelnen Typen zu bestimmen zählen wir die Häufigkeit und dividieren jede dieser durch die Gesamtzahl der Schlaganfälle, was eine Verteilung mit sich bringt, die wir in Table 2 sehen.

G45	0.0462
I60	0.1075
I61	0.2442
I63	0.6019

Table 2: Wahrscheinlichkeiten der einzelnen Schlaganfalltypen

Für die bessere Lesbarkeit, wurden die Werte bei der vierten Nachkommastelle gerundet.

Die Bestimmung des Schweregrads gestaltet sich etwas schwieriger, da uns hierfür keine Echtdaten zur Verfügung gestanden sind und in der Literatur wenig über die Verteilung zu finden war. Wir entschieden uns die Klassifikation von [10] zu verwenden wonach folgende fünf Schweregrade unterschieden werden:

- *Stadium 1*: Hierbei müssen keine Symptome auftreten.
- *transistorische ischämische Attacke (TIA)*: Ist eine rasch vorübergehende Durchblutungsstörung des

Gehirns, welche sich durch 10 – 15 Minuten lang andauernde neurologische Ausfallerscheinungen (Blindheit, Sprach- und Schluckstörung, etc.) bemerkbar macht. Langanhaltende Schäden können mittels chirurgischen Eingriffen verhindert werden.

- *prolongiertes reversibles ischämisches neurologisches Defizit (PRIND)*: Ist Vergleichbar mit TIA, doch die vollständige Genesung dauert länger. In der Literatur wird manchmal zwischen TIA und PRIND nicht unterschieden.
- *progressiver Schlaganfall (PS)*: Ist der typische Schlaganfall, welcher durch Lähmungserscheinungen, die später in spastische Verkrampfungen übergehen, charakterisiert wird. Die neurologischen Schäden können komplett, teilweise beziehungsweise nicht behoben werden.
- *kompletter Schlaganfall, Apoplex (CS)*: Hiermit ist ein Schlaganfall gemeint, dessen neurologischen Symptome, wie Lähmungen, anhalten sind.

Wir entscheiden uns für die *Betaverteilung*, welche mittels geeigneter Parameterwahl die Häufigkeit des Schweregrades nahezu beliebig ändern kann (siehe Abbildung 8).

Aufgrund der fehlenden Daten, wählen wir für unser Modell die Parameter der Verteilung so, dass die "typischen" Schlaganfälle, wie TIA, PRIND und PS am häufigsten vorkommen. Es ist anzunehmen, dass Schlaganfälle aus Stadium 1 häufiger vorkommen, doch da in unserem Projekt die Krankenhausdiagnosen betrachten werden und bei dieser Art keine Symptome auftreten und eher zufällig entdeckt werden [10] ist die Dunkelziffer zwar hoch, doch die der Diagnosen vergleichsweise gering. Deshalb wählten wir für $\alpha = 1.5$ und $\beta = 1.5$.

3.5 Auswirkung auf Todesfälle

Auch die im Kapitel 3 erklärte Funktion *randomly_decide_to_die* wurde im neuen Modell etwas abgeändert. Da es zu einem Drittel wahrscheinlich ist, nach erleiden eines Schlaganfalles zu sterben [3], wird bei der Funktion überprüft, ob in diesem Time-step ein Schlaganfall passiert ist, und wenn ja, wird der Patient diesen nur zu 66 % überleben. Auch dieser Parameterwert musste mangels Datenverfügbarkeit mit Expertenunterstützung geschätzt werden.

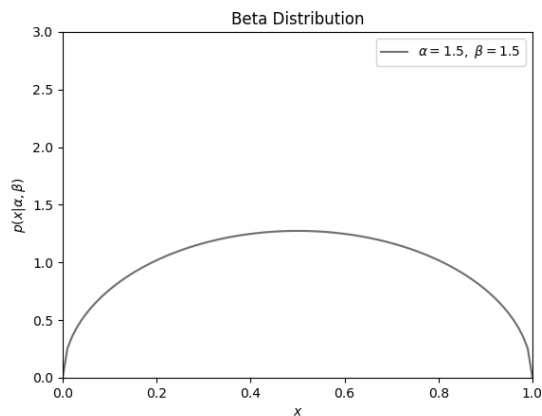


Abbildung 8: Betadichte für $\alpha = 1.5, \beta = 1.5$

4 Outputvisualisierung mit Visplore

Zum Abschluss unseres Projekts verwendeten wir das Programm Visplore um unsere Simulation zu analysieren. Visplore ist eine Softwaretechnologie des Komet K-Zentrums VRVis, mit der man flexible und kombinierbare Visualisierungen, Interaktionen und Berechnungen durchführen kann [4]. Diese Software bietet die Möglichkeit interaktive Analyse von sehr großen, mehrdimensionalen Datensätzen zu machen. Diese Eigenschaft ist wichtig, da, je nach Einstellungen, sehr große Ausgabedateien von unserem Programm entstehen können.

In Abbildung (9) erkennt man die zahlreichen Funktionen der Arbeitsfläche. Man kann nach verschiedensten Merkmalen filtern, zwischen beliebigen Zeiträumen wählen und diese Daten dann graphisch auswerten lassen.

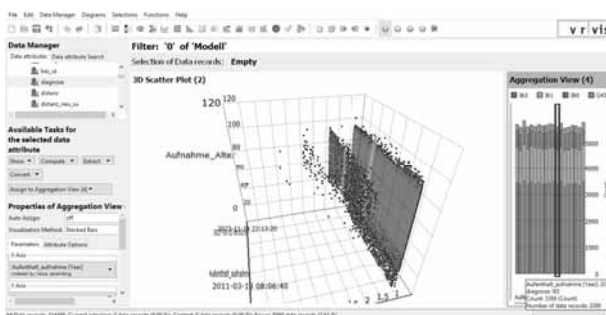


Abbildung 9: Benutzeroberfläche von Visplore mit eingelesenem Datensatz einer Simulation über 20 Jahre

In dieser Abbildung ist ein Datensatz eingelesen worden, der durch eine Simulation der österreichischen Bevölkerung über 20 Jahre von unserem Programm erzeugt wurde. Die Achsen dieses 3D Plots sind gewisse Eigenschaften der Schlaganfallerträge. Drei Informationen eines Eintrags bilden die Koordinaten für einen Punkt im Plot. Wir werden später noch näher darauf eingehen, was hier zu sehen ist.

5 Analyse

Zunächst wurde die Simulation auf vielerlei Arten einem Verifikationsprozess unterzogen. Zum Beispiel stimmt im Modelloutput die Verteilung der Schweregrade (Abbildung 10) gut mit der vorgegebenen Dichte (Abbildung 8) überein.

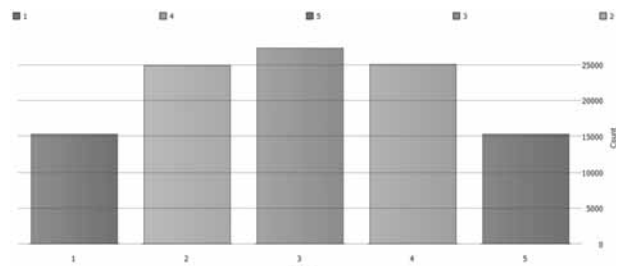


Abbildung 10: Verteilung der Schweregrade in den Modellergebnissen

In Abbildung 11 ist der 3d-Plot aus Abbildung 9 größer dargestellt. Diesen werden wir im Folgenden näher diskutieren.

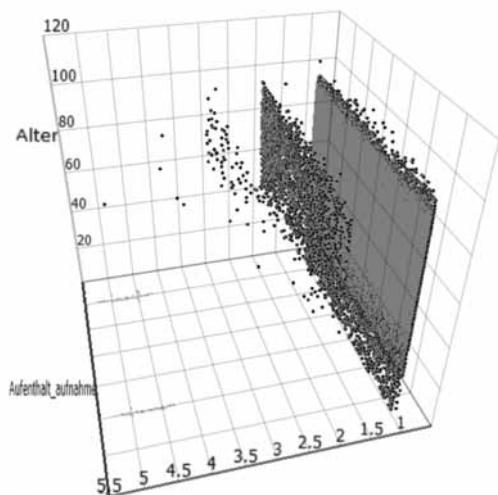


Abbildung 11: 3D Scatterplot der simulierten Schlaganfälle. Ein Punkt entspricht einem Schlaganfall von einer Simulation über 20 Jahre

Die Koordinate für die grüne Achse ist das Alter des Patienten, der diesen Schlaganfall erlitten hat, die für die blaue Achse entspricht dem Datum des Ereignisses und auf der roten Achse erkennt man, der wievielte Schlaganfall für den Patienten im Eintrag vermerkt wurde. Diese Bild visualisiert die Häufigkeit der Schlaganfälle pro Altersgruppe. Daran kann man ablesen, dass die älteren Personen, wie erwünscht, mehrere Hirnschläge als Jüngere bekommen.

Eine weitere interessante Grafik, die wir mit Visplore erstellt haben, ist in Abbildung 12 zu sehen.

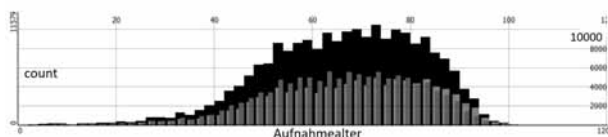


Abbildung 12: Verteilung der Schlaganfälle über Altersgruppen (x-Achse) und Geschlecht (rot-weiblich, blau-männlich, schwarz-gesamt)

Dabei entsprechen die schwarzen Balken der Gesam-

tanzahl aller Schlaganfälle in der bestimmten Altersgruppe, welche wiederum mit Männer (rot) und Frauen (blau) kategorisiert sind. In unserem Modell kann auch unterschieden werden, ob ein simulierter Todesfall Folge eines Schlaganfalls war oder nicht. Bei einer Simulation der gesamten österreichischen Bevölkerung über 20 Jahre entstehen mit unserem Modell ca. 216000 Schlaganfälle, wobei in etwa 72000 direkt zum Tod führen.

Zum Abschluss betrachten wir noch in Abbildung 13 die Auswirkungen des Alters auf die Todesursache, welche durch die erhöhte Todeswahrscheinlichkeit, nicht so groß sein sollte.

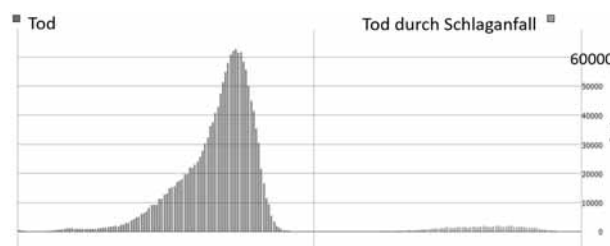


Abbildung 13: Auswirkung des Alters auf die Todesursache (blau-gesamt, orange-durch Schlaganfall) über Altersgruppen (x-Achse)

Wie man sieht vereinfachen die Grafiken, welche mittels Visplore erstellt wurden, die Bestätigung der Erwartungshaltung von den erzeugten Ergebnissen der Simulation.

6 Ausblick

Die Aufgabe unseres Projektes war es, aufbauend auf das GEPOC-Modell ein vollständiges Schlaganfallmodell zu kreieren, welches realitätsnahe Simulationen durchführen kann. Auf Basis unseres einzigartigen Modells kann man die verschiedensten Szenarien durchtesten und breitgefächerte Analyse betreiben. So könnte man unterschiedliche Risikoparameter auf deren Auswirkung testen und Interventionstrategien simulieren. Beispielhaft könne man hierbei regelmäßiges Saunieren anführen, welchen nach einer aktuellen Studie das Schlaganfallrisiko um bis zu 61% senken kann [11]. Auf der anderen Seite ist z.B. bekannt [12], dass missbräuchlicher Drogenkonsum (z.B. Crystal Meth) die Schlaganfallwahrscheinlichkeit stark erhöhen kann. Interessant zu testen ist auch, wie sich die alternde Bevölkerung auf die Schlaganfallsituation in der Zukunft auswirken wird. Dafür

kann unser Modell gewisse Einsichten bringen, die bei wichtigen strukturellen Entscheidungen zu Rate gezogen werden, zum Beispiel für Ausbau in Infrastruktur für Kliniken, Rehabilitationszentren oder Bedarf für neue Medikamente. Hierbei kommt der Vorteil des agentenbasierten Modells zum Vorschein, da durch die Interaktion der einzelnen Individuen Bottlenecks bei medizinischen Einrichtungen erkannt werden können. Damit könnte unser Modell den medizinischen Markt revolutionieren. Leider hat das Modell primär Mangels Parametrisierungsdaten und Validierungsdaten immer noch großen Validierungsbedarf. Dementsprechend wäre der nächste logische Schritt die Intensivierung der Validierungsbemühungen, was jedoch in Anbetracht der sehr sensiblen und dementsprechend komplizierten Datenlage nicht ganz einfach ist. Abgesehen davon haben wir noch einige Ideen um unser Modell auszubauen und zu verbessern, deshalb wollen wir uns noch über mögliche Erweiterungen dieses Konzeptes unterhalten.

Unter Anderem wäre die Regionalität der Agenten ein Punkt, den man für Erweiterungen der Berechnung von Todes- und Schlaganfallwahrscheinlichkeiten verwenden könnte, wenn man genau wüsste, unter welchen Faktoren und wie die Bezirke von Österreich Einfluss darauf nehmen. Damit könnte man die österreichische Versorgungslandschaft (z.B: die Verfügbarkeit von Stroke Units) noch deutlich besser unter die Lupe nehmen.

Ein anderer Aspekt wäre die genetische Vererbung gewisser Krankheiten, die man für die Kinder von diagnostizierten Agenten einfließen lassen könnte.

Weiters könnte man berücksichtigen, dass ein gewisser Anteil der Bevölkerung ein hohes Einkommen besitzt und damit die finanziellen Möglichkeiten, um die besten medizinischen Versorgung, wie zum Beispiel Aufenthalte in speziellen Privatkliniken, Rehabilitationszentren und Ähnlichem, zu bekommen. Sogar der Bildungsgrad könnte zum Optimieren des Modelles helfen, wenn man bedenkt, je höher dieser ist, desto höher ist die Wahrscheinlichkeit, dass die Person finanziell gefestigt ist, einen gesunden Lebensstil besitzt, ernährungsbewusst isst und das sie vernünftiger in Not-situationen agiert. Selbst damit ergeben sich wieder kleine Faktoren, die Wirkung auf das Modell haben können.

Wie wir sehen, könnte man auch unser Modell wieder in gewissen Bereichen erweitern und ausbauen, wodurch das sich unser Projekt als sehr nützlich erweist.

Danksagung

Das K-Projekt DEXHELPP wird von BMVIT, BMFWF und dem Bundesland Wien via COMET - Competence Centers for Excellent Technologies unterstützt. Das Programm COMET wird von der FFG abgewickelt. (Projektnummer 843550)

References

- [1] Liste der 10 häufigsten Todesursachen weltweit, <https://www.lifeline.de/medizinwissen/todesursachen-id123761.html>.
- [2] Schlaganfall, <https://www.gesundheit.gv.at/krankheiten/gehirn-nerven/schlaganfall/inhalt>.
- [3] des Bundes G. Gesundheit in Deutschland. *Berlin: Robert Koch-Institut*. 2006;pp. 117–8.
- [4] VRVis, <http://goo.gl/wqJ4AS>.
- [5] Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*. 2002;99(suppl 3):7280–7287.
- [6] Bicher M, Glock B, Miksch F, Schneckenreither G, Popper N. Definition, Validation and Comparison of Two Population Models for Austria. In: *Proceedings of 4th UBT Annual International Conference on Business, Technology and Innovation*. Durres, Albania: UBT - Higher Education Institution. 2015; .
- [7] Stroke Unit, <https://www.gesundheit.de/krankheiten/gehirn-und-nerven/schlaganfall/experteninterview-zum-schlaganfall>.
- [8] Ein Schlaganfall kommt selten allein Neue Perspektiven der Boehringer Ingelheim-Forschung zur Verhinderung von Schlaganfällen.
- [9] <http://www.icd-code.de/>.
- [10] Schlaganfall, <https://www.lecturio.de/lexikon/schlaganfall>.
- [11] Kunutsor SK, Khan H, Zaccardi F, Laukkanen T, Willeit P, Laukkanen JA. Sauna bathing reduces the risk of stroke in Finnish men and women. *Neurology*. 2018;.
- [12] Anglin MD, Burke C, Perrochet B, Stamper E, Dawud-Noursi S. History of the methamphetamine problem. *Journal of psychoactive drugs*. 2000; 32(2):137–141.
- [13] Nikolai C, Madey G. Tools of the trade: A survey of various agent based modeling platforms. *Journal of Artificial Societies and Social Simulation*. 2009;12(2):2.

Modellierung von Bewegung mit Hilfe Zellulärer Automaten

Carina Hörandtner^{1*}, Andreas Körner¹

¹Institute for Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *carina.hoerandtner@tuwien.ac.at

Abstract. Ein zweidimensionaler Zellulärer Automat (CA) wird dazu verwendet, Bewegung in einem Raum zu modellieren. Dabei wird versucht, Personen durch eine oder mehrere Türen zu evakuieren und somit eine gewisse Dynamik zu erzeugen. Es werden sowohl Räume mit als auch ohne Hindernisse betrachtet. Wie die Größe des Ausgangs mit der Zeit der Evakuierung in Verbindung steht, wird in einem Raum ohne Hindernisse untersucht, wobei ab einer gewissen Türbreite keine nennenswerten Verbesserungen mehr erzielt werden. Periodische Randbedingungen ermöglichen es, das Verlassen und neuerliche Betreten eines Raumes zu simulieren. Am Beispiel einer Lagerhalle zeigt sich, dass Hindernisse den Bewegungsfluss von Personen stören und die Lage des Ausgangs für die Evakuierung eine wichtige Rolle spielt. Um das Modell valider zu machen und einen Gegenstrom zu simulieren, werden zwei Gruppen von Personen mit zwei verschiedenen Ausgängen einander gegenübergestellt. Die verwendete CA Regel ermöglicht es den Personen, sich ohne Kollisionen durch den Raum zu bewegen und in geschlossener Formation diesen auch zu verlassen.

Einleitung

Eine wichtige Eigenschaft, die Lebewesen von lebloser Materie unterscheidet, ist die Fähigkeit sich zu bewegen. Dabei beeinflussen sich Menschen in ihrer Fortbewegung gegenseitig, da sie dazu neigen, Kollisionen mit anderen Personen aus dem Weg zu gehen und Hindernissen auszuweichen. Besonders interessant zu beobachten, ist das Verhalten in geschlossenen Räumen mit nur einem oder wenigen Ausgängen. Wollen alle Personen den Raum gleichzeitig verlassen, gerät die Bewegung meist rasch ins Stocken und es kommt zu einer Staubildung vor dem Ausgang. Es ist deshalb von großer Bedeutung, die Türen an optimalen Posi-

tionen anzubringen, um somit im Notfall eine schnelle Evakuierung gewährleisten zu können.

Auf Grund ihrer Simplizität erweisen sich Zelluläre Automaten als gängige Methode, Evakuierungsprozesse und Bewegungsabläufe zu modellieren, aber auch agentenbasierte Modelle oder partielle Differentialgleichungen (PDE) finden häufig Anwendung.

In dieser Arbeit wird ein CA Ansatz vorgestellt, der Bewegung in einem Raum simuliert, wobei sich der Aufbau des CA an dem verwendeten Modell in [3] orientiert und in Abschnitt 2 erklärt wird. In Abschnitt 3 werden einige Szenarien dargestellt und analysiert. Dabei wird sowohl das Verhalten bei fixen Randbedingungen, als auch bei periodischen Randbedingungen, sowie das Verhalten in leeren Räumen und in Räumen mit Hindernissen untersucht. Des Weiteren werden zwei Gruppen von Personen mit unterschiedlichen Ausgängen einander gegenübergestellt und ihre Bewegungen analysiert. Abschnitt 4 fasst die Ergebnisse zusammen und gibt einen Ausblick über mögliche Modellerweiterungen.

1 Aufbau eines Zellulären Automaten

Im Allgemeinen besteht ein Zellulärer Automat aus folgenden Komponenten:

- einem d -dimensionalen Gitter aus Zellen
- einer abzählbaren, diskreten Zustandsmenge \mathbb{S}
- einer endlichen Nachbarschaft \mathcal{N}
- einer lokalen Übergangsregel $\mathcal{F} : \mathbb{S}^{\mathcal{N}} \rightarrow \mathbb{S}$

Im Folgenden werden die oben angeführten Elemente näher erläutert.

1.1 Definition des Zellraums

Als Zellraum bezeichnet man jenen Raum, in dem die Abläufe stattfinden und der durch ein d -dimensionales Gitter repräsentiert wird. Für gewöhnlich betrachtet man ein-, zwei- oder dreidimensionale Automaten, wobei die Wahl der Dimension von dem zu modellierenden System abhängt. Im zweidimensionalen Fall wird am häufigsten ein rechteckiges oder quadratisches Gitter verwendet, da es sich am einfachsten realisieren lässt.

Jede Zelle des Gitters bekommt einen Zustand $s \in \mathbb{S}$ zugewiesen. Die Zustandsmenge \mathbb{S} ist abzählbar und sollte nicht zu groß gewählt werden, wobei ihre Elemente nicht notwendigerweise Zahlen sein müssen.

1.2 Definition der Nachbarschaft

Jeder Gitterplatz besitzt eine Nachbarschaft, die im Vorfeld festgelegt wird und für jede Zelle gleich ist. Grundsätzlich gibt es keine Größenbeschränkung der Nachbarschaft, man wählt jedoch meist angrenzende Zellen. Die am häufigsten verwendeten Nachbarschaften bei zweidimensionalen Zellulären Automaten mit quadratischem Gitter sind die von Neumann und die Moore Nachbarschaft. Die von Neumann Nachbarschaft besteht aus der betrachteten Zelle selbst sowie ihren vier direkt an den Zellkanten angrenzenden Nachbarzellen. Die Moore Nachbarschaft berücksichtigt noch zusätzlich die vier angrenzenden Eckzellen, siehe [1].

1.3 Definition der Übergangsregel

Die wichtigste Komponente eines CA ist die Übergangsregel, welche die zeitliche Entwicklung der Zellen bestimmt. Durch diese als Funktion angegebene Regel werden die Zustände der Zellen in Abhängigkeit der Nachbarn und der Zelle selbst in jedem diskreten Zeitschritt aktualisiert. Die Übergangsregel ist für alle Zellen gleich und wird räumlich parallel und zeitlich synchron auf alle Zellen angewendet, siehe [2]. Man kann jedoch CA erweitern und eine asynchrone zeitliche Entwicklung zulassen.

Formal lässt sich die Änderung des Zustandes $s_{\vec{r}}^t$ einer Zelle mit Position \vec{r} zum Zeitpunkt t in den Zustand $s_{\vec{r}}^{t+1}$ durch eine Funktion $\mathcal{F} : \mathbb{S}^{\mathcal{N}} \rightarrow \mathbb{S}$ wie folgt darstellen:

$$s_{\vec{r}}^{t+1} = \mathcal{F}(\{s_{\vec{r}'}^t \mid \vec{r}' \in \mathcal{N}(\vec{r})\}), \quad (1)$$

wobei \mathcal{F} die Übergangsregel und \vec{r}' die Nachbarzellen von \vec{r} beschreiben, siehe [2].

Es werden zwei Arten von Zellulären Automaten unterschieden: Automaten mit deterministischen oder mit stochastischen Regeln. Bei deterministischen CA wird die Entwicklung eines Automaten für einen gegebenen Anfangszustand eindeutig festgelegt, das heißt, eine Wiederholung mit den gleichen Anfangszuständen führt stets zu demselben Ergebnis. Für manche Anwendungen ist es jedoch von Vorteil, einen gewissen Grad an Zufälligkeit in der Übergangsregel zu haben. Im Falle von stochastischen Automaten liefern die Übergangsregeln kein eindeutiges Ergebnis für jede Konfiguration, sondern geben mehrere mögliche Zustandsänderungen unter einer Wahrscheinlichkeitsverteilung an. Die Summe der Wahrscheinlichkeiten für das Erreichen aller definierten Zustände muss 1 ergeben, siehe [4].

2 Modellbeschreibung

Der Raum in dem sich die Personen bewegen, wird durch ein zweidimensionales quadratisches oder rechteckiges Gitter realisiert. Jede Zelle besitzt eine Moore Nachbarschaft und kann entweder frei sein oder durch eine Person, ein Hindernis, einen Ausgang oder durch Wandabschnitte, die den Raum begrenzen, besetzt sein. Es sei bemerkt, dass sich auf einer Zelle höchstens eine Person befinden darf. Diesem Gitterplatz wird dann der Zustand 1 zugewiesen, wohingegen freie Zellen den Zustand 0 bekommen. Wände haben den Wert 2 und werden nur durch Ausgänge, bestehend aus einem oder mehreren Gitterplätzen mit Zustand 3, unterbrochen. Hindernisse bekommen den Zustand 4 zugewiesen. Wie sich vermuten lässt, wechseln nur Zellen mit den Werten 0 oder 1 ihren Zustand, Wandzellen und Hindernisse ändern ihren Wert jedoch im gesamten Zeitraum nicht.

Im Unterschied zu einem klassischen CA wird das betrachtete Modell um eine Komponente erweitert. Das sogenannte Bodenfeld dient als Hilfsmittel der hier verwendeten Übergangsregel und wird im folgenden Abschnitt näher erklärt.

2.1 Beschreibung des Bodenfelds

Nachdem die Raumgeometrie und die Position bzw. die Größe des Ausgangs festgelegt wurde, wird jeder Zelle ein Wert, unabhängig ihres Zustandes, zugewiesen, der

den Abstand zum Ausgang repräsentiert. Zellen, deren Kanten oder Ecken an den Ausgang angrenzen, bekommen den kleinsten Wert 1, da die Person dann nur noch einen Schritt benötigt, um den Raum zu verlassen. Je weiter man sich von der Tür entfernt, desto größer werden die Werte. Wandzellen oder Hindernisse, falls vorhanden, können natürlich nicht betreten werden und bekommen daher einen sehr hohen Wert zugewiesen. Das so entstehende Bodenfeld zeigt den Personen im Raum nun den kürzesten Weg zum Ausgang an, vergleiche [3]. In Abbildung 1 wird ein Bodenfeld eines 22×22 Zellen großen Raumes ohne Hindernisse dargestellt. Je dunkler die Zellen werden, desto größer ist der Abstand zur Tür, die sich auf der rechten Seite befindet.

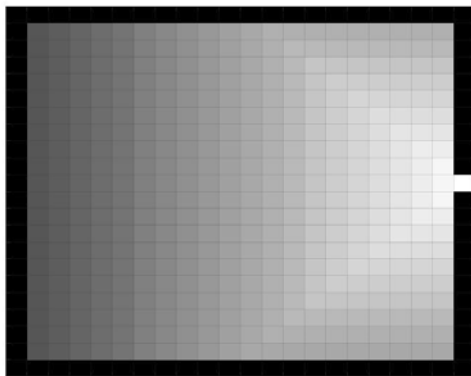


Abbildung 1: Bodenfeld eines 22×22 Zellen großen Raumes mit Ausgang auf der rechten Seite

2.2 Beschreibung der Übergangsregel

Am Anfang $t = 0$ werden zufällig N Personen im Raum verteilt, wobei darauf geachtet wird, keine Person auf ein Hindernis oder ein Wandstück zu setzen. In jeder der darauffolgenden Iterationen wird der Zellraum auf den Zustand 1 untersucht und die jeweilige Zelle mit ihrer Moore Nachbarschaft einer Übergangsfunktion übergeben. Die betrachtete Person sucht nun in dem 3×3 Block nach dem geringsten Wert des Bodenfeldes, also den kürzesten Weg zum Ausgang. Dabei sei bemerkt, dass mehrere Zellen in der betrachteten Moore Nachbarschaft den gleichen kleinsten Wert besitzen können. Ist die ausgewählte Zelle frei, wird sich die Person im nächsten Zeitschritt dorthin bewegen. Ist die Zelle von einer anderen Person besetzt, wird der

nächste Gitterplatz mit dem kleinsten Bodenfeldwert in Betracht gezogen. Sind alle angrenzenden Zellen, die den Abstand zum Ausgang verkürzen, belegt, verharrt die Person auf ihrer Position. Da die Zustände aller Zellen synchron upgedatet werden, kann es passieren, dass sich mehrere Zellen im nächsten Zeitschritt auf ein und denselben Gitterplatz bewegen möchten. Weil in diesem Modell ein deterministischer Zellulärer Automat betrachtet wird, wandert die Person, die im Zellraum vor den anderen liegt, auf den bevorzugten Platz und die anderen suchen sich eine neue freie Zelle. Durch diese Regel entsteht Bewegung im Zellraum, die solange beibehalten wird, bis alle Personen den Raum verlassen haben oder eine gewisse Schranke an Iterationen erreicht ist.

3 Simulation und Analyse der Szenarien

3.1 Raum ohne Hindernisse

Im ersten Szenario wird untersucht, welche Auswirkung die Größe des Ausgangs auf die Zeit, die vergeht bis alle Personen den Raum verlassen haben, hat. Dabei werden N Personen in einem 30×30 Zellen großen Raum zufällig verteilt und danach für unterschiedliche Türgrößen die durchschnittliche Zeit der Räumung berechnet, vergleiche [3].

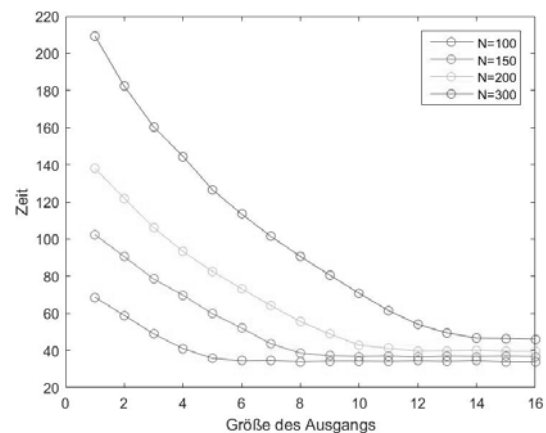


Abbildung 2: Türgröße verglichen mit der Zeit der Evakuierung, wobei am Anfang N Personen zufällig im Raum verteilt werden

In Abbildung 2 wurde die Simulation einmal mit 100, 150, 200 und 300 zufällig verteilten Personen

durchgeführt. Wie man erkennen kann, sinkt die Zeit nichtlinear bei Erweiterung des Ausgangs. Ab einer gewissen Breite hat die Türgröße jedoch nur mehr bedingt Einfluss auf die Zeit der Räumung. Für eine vergleichsweise geringe Anzahl von nur 100 Personen in dem betrachteten Raum wird schon ab einer Breite von 5 Zellen keine nennenswerte Verbesserung mehr erzielt. Das kann man damit begründen, dass die Personen eine gewisse Zeit benötigen, um den Ausgang zu erreichen. Ist nun die Tür groß genug, verlassen alle ankommenden Individuen den Raum sofort und es kommt zu keiner Staubildung.

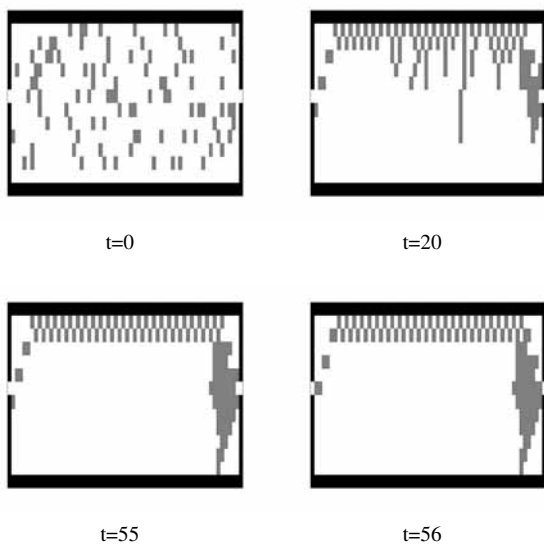


Abbildung 3: Bewegung in einem Raum der Größe von 12×60 Zellen mit periodischen Randbedingungen

In einem anderen Szenarium werden periodische Randbedingungen für den Eingang und Ausgang verwendet, um das Verlassen und neuerliche Betreten eines Raumes zu simulieren. Immer wenn eine Person den Raum durch eine Tür verlässt, kommt auf der anderen Seite eine neue Person herein. In Abbildung 3 wird ein Raum der Größe von 12×60 Zellen dargestellt, wobei sich am Anfang 100 Personen auf zufällig ausgewählten Positionen im Raum befinden. Wie man zum Zeitpunkt $t = 20$ schon deutlich erkennen kann, bildet sich früh eine Schlange am Rand des Raumes, die ihre Formation annähernd beibehält und sich Richtung Ausgang bewegt. Die Tatsache, dass sich die Personen am oberen Rand des Bildes fortbewegen, basiert auf der

festgelegten Regel. Wie in Abschnitt 2.2 erwähnt, wird in der 3×3 Nachbarschaft nach dem kleinsten Bodenfeldwert gesucht, wobei immer bei den oberhalb angeordneten Zellen begonnen wird. Dies führt dazu, dass sich die Individuen am oberen Rand orientieren.

Ab einer gewissen Iteration, hier ab $t = 55$, wechselt das Bild nur mehr zwischen denselben zwei Konfigurationen hin und her, wobei die Personen im unteren Teil den Raum nie mehr verlassen können, da Individuen, die sich in der oberen Schlange aufhalten, auf Grund der CA Regel als erstes verschwinden können. Das Entstehen einer Schlange wird auch in der Realität sehr häufig beobachtet und spricht daher für den hier verwendeten deterministischen Zellulären Automaten.

3.2 Raum mit Hindernissen

Das Platzieren von Hindernissen in einem Raum verändert dessen Bodenfeld und hat somit Auswirkungen auf die Bewegungsrichtung einzelner Personen. Hindernisse bekommen einen sehr hohen Bodenfeldwert zugeschrieben, um das Betreten jener Zellen zu verhindern. Der Weg zum Ausgang kann sich dadurch wesentlich verlängern und den Bewegungsfluss der Personen stören.

Um dies zu verdeutlichen, wird ein 30×30 Zellen großer Lagerraum betrachtet, in dem 10 Regale der Länge 10 wie in Abbildung 4 angeordnet werden. Es

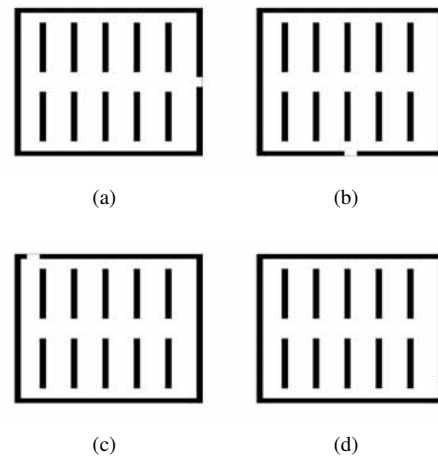


Abbildung 4: Lagerraum mit verschiedenen Lagen der Ausgänge

wird untersucht, welche Auswirkungen die Lage und Größe des Ausgangs auf die Zeit der Evakuierung aller Personen haben. Dafür werden wieder 100 Personen

zufällig im Lagerraum verteilt und für verschiedene Türpositionen Simulationen durchgeführt und jeweils die Durchschnittszeit berechnet, vergleiche [3]. Die gleichen Berechnungen werden auch für einen gleichgroßen Raum ohne Hindernisse angestellt. In Abbildung 5 werden die zwei Ergebnisse grafisch dargestellt. Die besten Resultate, sowohl mit als auch ohne Hindernisse, werden erzielt, wenn sich der Ausgang in der Mitte einer Wand befindet. Im Falle des Lagerraumes ist es ratsam, den Ausgang am Ende des Zwischengangs anzubringen, siehe Abbildung 4(a), da die Regale hier den Bewegungsfluss kaum stören und die Personen schon bei einer geringen Türgröße schnell den Raum verlassen können. Befindet sich der Ausgang in der Nähe einer Ecke, siehe Abbildung 4(c), 4(d), kommt es relativ rasch zu einer Staubildung, wobei die Hindernisse die Evakuierung zusätzlich erschweren. Allgemein spielt die Größe der Tür in der Lagerhalle mit Regalen kaum eine Rolle, da sich die Zeiten schon ab einer Breite von 3 Zellen kaum verbessern.

3.3 Zwei Gruppen an Individuen mit verschiedenen Ausgängen

Bisher wurden nur Personen betrachtet, die sich in eine gemeinsame Richtung bewegen, was klarerweise weniger Kollisionen zur Folge hat. Im nächsten Szenarium werden zwei Gruppen von Personen beobachtet, die jeweils einen anderen Ausgang anzielen. Diese Simulation erweist sich als wirklichkeitsgetreuer, da in der Realität oftmals mit einem Gegenstrom in großen Räumen oder in langen Gängen zu rechnen ist. Die Personen in einer Gruppe besitzen immer denselben Zustand. In diesem Fall bekommen Personen der Gruppe 1, die sich in Abbildung 6 zunächst auf der linken Seite befinden, den Zustand 1 und Personen der Gruppe 2, die sich zunächst rechts aufhalten, den Zustand 2 zugewiesen. Werte für Wandzellen, Hindernisse und Ausgänge werden dementsprechend nach hinten verschoben. Um nun verschiedene Bewegungsrichtungen zu erzeugen, müssen zwei verschiedene Bodenfelder generiert werden. Der Zellraum wird nun auf die Zustände 1 und 2 untersucht und die entsprechenden Zellen plus ihrer Moore Nachbarschaften der Übergangsfunktion übergeben. Die CA Regel für zwei Gruppen von Personen ist annähernd die gleiche, wie wenn nur eine Gruppe mit ein und demselben Ziel betrachtet wird. Einziger Unterschied ist, dass mögliche Kollisionen, die auftreten können, wenn zwei Personen unterschiedlicher Gruppe im nächsten Schritt die

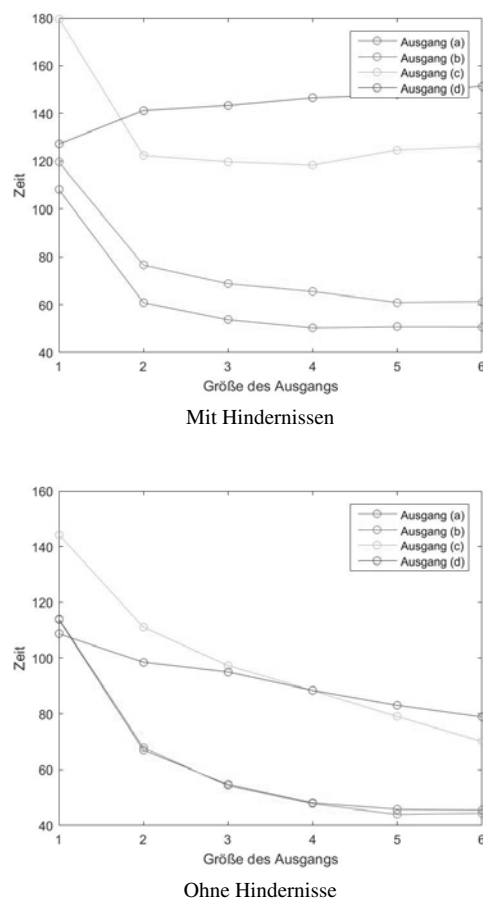


Abbildung 5: Türgröße verglichen mit der Zeit der Evakuierung in Abhängigkeit von der Türposition (a) - (d)

gleiche Zelle anstreben, verhindert werden, indem immer die Person mit dem kürzeren Weg zum Ausgang fortschreiten darf und die andere in ihrer Zelle verharren muss.

In Abbildung 6 werden zwei Gruppen mit je 75 Personen in einem 30×30 Zellen großen Raum gegenübergestellt, wobei darauf geachtet wird, dass sie sich am Anfang in ihrer Hälfte des Raumes befinden. Gruppe 1 hat nun den rechten Ausgang, Gruppe 2 den linken zum Ziel. Die Ausgänge sind jeweils 2 Zellen breit. Schon nach einigen Zeitschritten stoßen die beiden Gruppen aufeinander und die Bewegungen der Personen geraten ins Stocken. Die gewählte CA Regel ermöglicht es den Individuen jedoch, sich ohne Kollision durch den Raum zu bewegen und schlussendlich den jeweiligen Ausgang zu erreichen. Auffällig ist, dass sich die meisten Personen im oberen Teil des Bildes

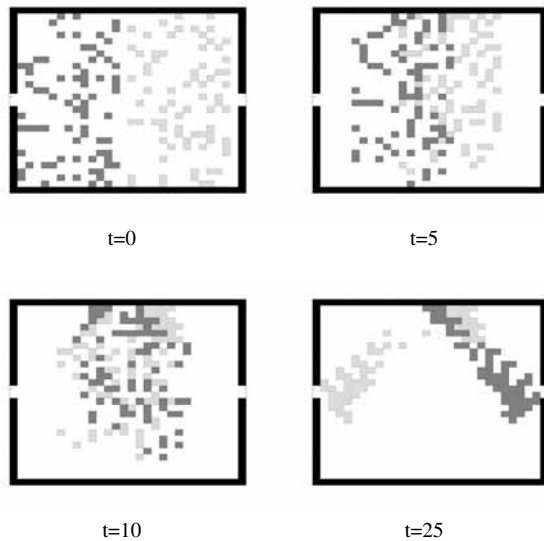


Abbildung 6: Gruppe 1 bewegt sich zum rechten Ausgang, Gruppe 2 zum linken

fortbewegen und dann in geschlossener Formation die Tür anstreben. Dieses auftretende Verhalten lässt sich wieder durch die Verwendung der hier gewählten deterministischen Übergangsregel erklären, wobei dieses Muster in der Realität nicht ungewöhnlich ist, da Menschen dazu tendieren, anderen, sich in die gleiche Richtung fortbewegenden Personen, nachzugehen.

4 Zusammenfassung und Ausblick

In dieser Publikation wurde mittels eines CA Ansatzes Bewegungen von Personen in einem Raum dargestellt. Dabei ließen sich Verhaltensmuster erkennen, die auch in der Realität öfter beobachtet werden, wie zum Beispiel die Entstehung eines Staus vor einem Ausgang mit nur geringer Breite. Um das neuerliche Betreten eines Raumes zu simulieren, wurden periodische Randbedingungen verwendet, wobei sich relativ bald eine Schlange bildete, die in geschlossener Formation den Ausgang ansteuerte. Auch in dem Szenarium mit den zwei Gruppen und den zwei verschiedenen Ausgängen ließ sich erkennen, dass Personen, die sich in die gleiche Richtung bewegten, einander folgten und Kollisionen mit der anderen Gruppe vermieden. Dieses Verhalten lässt sich durch die verwendete CA Regel erklären und wird auch in der Wirklichkeit häufig

beobachtet. Wie sich Hindernisse auf den Bewegungsfluss einzelner Personen auswirken, wurde am Beispiel einer Lagerhalle mit Regalen untersucht. Dabei stellte sich heraus, dass die Position des Ausgangs für die Zeit der Evakuierung eine wesentliche Rolle spielte, da die Regale den leichten Zugang zu den Ausgängen teilweise verhinderten.

Da in dieser Publikation ein deterministischer CA verwendet wurde, versuchten die Personen immer, in die erste Nachbarzelle mit dem kleinsten Bodenfeldwert fortzuschreiten. Des Weiteren wurde bei der Untersuchung des 3×3 großen Zellenblocks immer bei den oberhalb angeordneten Zellen begonnen. Dies führte dazu, dass sich die Individuen, wenn der Ausgang auf der rechten oder linken Seite lag, häufig am oberen Teil des Bildes fortbewegten. Weil das Verhalten der Menschen in der Realität jedoch nicht immer voraussehbar ist, könnte man die Übergangsregel wie in [3] erweitern und somit einen stochastischen Zellulären Automaten erhalten.

Der hier verwendete CA Ansatz, um die Bewegung von Personen zu simulieren, entspricht keinem klassischen Zellulären Automaten und erinnert an ein Agenten-basiertes Modell mit einem einfacheren Regelwerk. Um das betrachtete Modell noch wirklichkeitsgetreuer zu machen und unterschiedliche Eigenschaften von Personen zu berücksichtigen, könnte nun ein agentenbasierter Ansatz gewählt werden. Mit diesem Ansatz wäre es zum Beispiel möglich, unterschiedliche Geschwindigkeiten zu simulieren, die im CA Modell für alle gleich und auf einen Schritt pro Zeiteinheit beschränkt sind.

References

- [1] Chopard B., Droz M. *Cellular automata modeling of physical systems*. Cambridge: Cambridge University Press, 1998.
- [2] Renner, U. *Zelluläre Automaten. Ein ausgewählter Überblick*. Leipzig: 2000. www.synatech.de/Digital_ZA/PDF/ZA_Blick.pdf [Zugriff: 4.1.2018].
- [3] Varas A., Cornejo M., Mainemer D. et al. Cellular automaton model for evacuation process with obstacles. *Physica A: Statistical Mechanics and its Applications*. 2007, 382(2): 631-642. doi: 10.1016/j.physa.2007.04.006.
- [4] Weimar, J.R. *Simulation with Cellular Automata*. Berlin: Logos-Verlag, 1997.

Von der Co-Simulation zu neuronalen Netzmodellen: automatisierte Erzeugung von schnellen, schaltbaren Systemmodellen

Philipp Ebeling, Julian Kalmus, Wilhelm Tegethoff

TLK-Thermo GmbH, Hans Sommer Straße 5,-38106 Braunschweig

*p.ebeling@tlk-thermo.com

Abstract. Co-Simulationsmodelle können komplexe Fragestellungen bei der Entwicklung von Fahrzeugkomponenten und -systemen beantworten. Für einige spezifische Anwendungsfälle, wie das Testen von Reglern, ist der Detaillierungsgrad der Modelle unnötig hoch. Darüber hinaus besteht der Wunsch nach höherer Simulationsschwindigkeit und der Reduktion notwendiger Simulationsumgebungen. Mit der Überführung von Co-Simulationsmodellen zu schaltbaren Systemmodellen aus neuronalen Netzen können diese erfüllt werden. Bislang waren die hierfür notwendigen Arbeitsschritte nicht automatisiert und damit aufwändig und teuer. Der Arbeitsablauf kann durch ein neu-entwickeltes Rahmenwerk durchgängig automatisiert werden. In diesem Artikel wird der automatisierte Prozess von einer Co-Simulation zu einem schaltbaren neuronalen Netzmodell am Beispiel der beiden Systeme Kälte- und Kühlkreislauf vorgestellt.

Motivation

Um komplexe, systemübergreifende Fragestellungen bei der Entwicklung von Fahrzeugkomponenten und -systemen beantworten zu können, bedarf es detaillierter Modelle. Diese werden vorwiegend in Fachabteilungen in ihrer jeweils spezialisierten Simulationsumgebung entwickelt. Die Modelle können in einer Co-Simulationsumgebung [1-3] zu einem Gesamtfahrzeugmodell verschaltet und über ihre jeweilige Software berechnet werden. Durch die Notwendigkeit mehrerer Software-Installationen und der Verfügbarkeit entsprechender Lizenzen ist der Aufwand für eine Portierung des Gesamtfahrzeugmodells zu anderen Fachbereichen groß. Zudem erfüllt die Geschwindigkeit der Co-Simulation wegen des hohen Detaillierungsgrads der Modelle oftmals nicht die Wünsche des Anwenders. In vielen Fällen weisen diese

Modelle für konkrete Fragestellungen an die Gesamtfahrzeugsimulation unnötig hohe Detaillierungsgrade auf. Weiterhin besteht der Wunsch nach einer Verringerung der Anzahl notwendiger Simulationsumgebungen, ohne nennenswerte Qualitätseinbußen in der Simulation des Systemverhaltens hinnehmen zu müssen. Mögliche Einsatzbereiche sind z.B. der Test von Reglern in SiL- oder HiL-Umgebungen mit gekoppelten Modellen des Gesamtfahrzeugs oder die Verwendung der Modelle für modellbasierte prädiktive Regelungen. Für diesen Einsatzbereich ergibt sich die besondere Anforderung, dass die konvertierten Modelle alle möglichen Stellgrößeneingänge in vollem Umfang abbilden müssen.

Ein möglicher Ansatz zur Erfüllung der genannten Anforderungen ist die Erstellung von datengetriebenen Modellen der Teilsysteme mit Hilfe von neuronalen Netzmodellen. Neuronale Netzmodelle verringern die Rechendauer im Vergleich zu den physikalisch motivierten Ursprungsmodellen signifikant und ermöglichen die Berechnung des Gesamtfahrzeugs in einer einzigen Simulationsumgebung. Über die Verschaltung vieler dynamischer, neuronaler Netzmodelle, die jeweils für bestimmte Betriebsbereiche gültig sind, zu einem Gesamtsystemmodell kann die Regelbarkeit des Gesamtmodells für alle Betriebszustände erhalten werden.

1 Arbeitsschritte von Co-Simulations- zu neuronalen Netzmodellen

Bislang erfordert der Prozess zur Erstellung der neuronalen Netzmodelle aus Co-Simulationsmodellen eine hohe Anzahl von manuellen, fehleranfälligen und zeitaufwändigen Zwischenschritten. Aufgrund des damit verbunde-

nen hohen Zeitaufwands wird dieser Prozess nur in seltenen, ausgewählten Fällen durchgeführt. Das Potential von schnellen Gesamtfahrzeugmodellen in peripheren Entwicklungsprozessen wird daher bislang nicht genutzt. Mit Hilfe eines neu-entwickelten Rahmenwerks können die notwendigen Arbeitsschritte durchgängig automatisiert werden. In Abbildung 1 ist der Arbeitsablauf schematisch abgebildet. Im ersten Schritt müssen die Eingangsgrößen abzubildender Betriebspunkte als Datengrundlage zum Trainieren der neuronalen Netzmodelle definiert (Design of Experiment, DoE) und Gültigkeits- bzw. Plausibilitätsgrößen für Simulationsergebnisse festgelegt werden. Die definierten Betriebspunkte werden mit den detaillierten Modellen der Co-Simulation berechnet. Im nächsten Arbeitsschritt müssen alle Ergebnisse hinsichtlich der Einhaltung der Gültigkeits- bzw. Plausibilitätsgrößen und der Stationarität überprüft werden. Die neuronalen Netzmodelle werden anhand der gefilterten Simulationsergebnisse trainiert. Die Zuordnung der Eingangs- und Ausgangsgrößen der neuronalen Netzmodelle erfolgt über eine Systembeschreibungsdatei. Über die Systembeschreibung erfolgt außerdem die Zuordnung der Modelle zu Systemen und deren Systemklassen, wie z.B. das Kältekreislaufsystem mit der Schaltungsvariante Wärmepumpe. Über ein Matlab-Programm wird aus den einzelnen neuronalen Netzmodelle und der Systeminformation automatisiert ein Matlab/Simulink-Systemmodell erzeugt. Das Simulink-Systemmodell beinhaltet somit alle Teilsysteme und deren Klassen. Über zusätzlich instanziierte Steuermodelle können die einzelnen Netzmodelle aktiviert und zwischen den Systemklassen umgeschaltet werden. So bleiben die Regelbarkeit und die Umschaltbarkeit von z.B. Ventilen erhalten. Die Validierung der Modelle erfolgt anschließend über einen Vergleich der Simulationsergebnisse des Simulink-Systemmodells mit den Trainingsergebnissen. In einem letzten Simulationsschritt vor Freigabe des Modells (Roll out) wird die Plausibilität des Modells bei Umschaltvorgängen in einer transienten Simulation überprüft. Dies kann in einer Simulationsprüfstand-Umgebung oder in einer Co-Simulation erfolgen. Zu Dokumentationszwecken werden alle Simulationsergebnisse und erzeugten Modelle automatisiert in einer Datenablage abgelegt. Für relevante Variablen werden automatisiert Diagramme mit Zeitverläufen erstellt und zusammen mit Metadaten in einer PowerPoint-Präsentation oder als PDF abgespeichert und abgelegt.

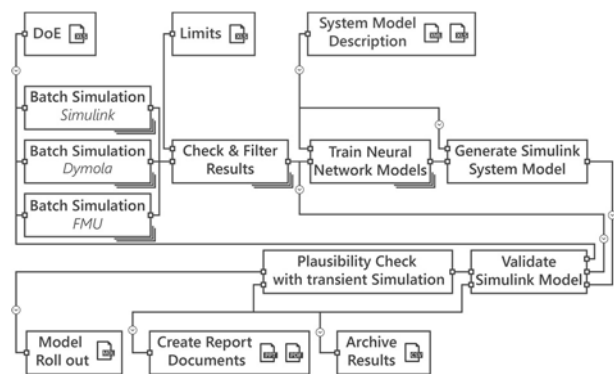


Abb. 1: Arbeitsablauf zum Erstellen eines umschaltbaren, datengetriebenen Systemmodells aus Einzelmodellen der Co-Simulation, der über das entwickelte Rahmenwerk automatisiert wird.

2 Rahmenwerk zur Automatisierung des Arbeitsablaufs

Mit einem neuartigen Python-basierten Rahmenwerk [4] können die in Abbildung 1 gezeigten Arbeitsschritte durchgängig automatisiert werden. Eines der Ziele bei der Entwicklung des Rahmenwerks war die einfache Generierung und Anpassung von nutzerspezifischen Arbeitsabläufen aus objektorientiert programmierten Arbeitsschritten (Tasks) mit definierten Schnittstellen. Die Tasks können Aufgaben unterschiedlichster Anwendungsbereiche sein, wie z.B. Datenverarbeitung und -analyse, Dateimanagement, Simulation, Optimierung oder Visualisierung. Der Nutzer kann bei der Erstellung von individuellen Arbeitsabläufen eigene Tasks in Form von Python-Programmen implementieren oder auf Tasks einer Standard-Bibliothek zurückgreifen. Für viele Software-Werkzeuge, wie z.B. Matlab, Dymola, Excel und PowerPoint, wird eine Python-Programmierschnittstelle (API) ausgeliefert. Dies ermöglicht eine unkomplizierte Einbindung und die Steuerung durch das Rahmenwerk. Ein weiteres Hauptmerkmal der Automatisierungsumgebung ist die automatisierte Distribution und Ausführung der Arbeitsschritte an angebundene Netzwerk- oder Cloudrechner. Innerhalb des Rahmenwerks werden Arbeitsabläufe, Modelle und Ergebnisse in sogenannte Projekte zusammengefasst, verwaltet und archiviert. Die Bearbeitung durch einen Rechnerverbund und die Portierbarkeit auf andere Rechensysteme wird durch die Verwendung von relativen Pfadstrukturen ermöglicht. Die

Batch-Simulationen des vorgestellten Arbeitsablaufs können durch das Rahmenwerk ressourceneffizient durchgeführt werden. Alle an das Projekt angebundene Netzwerkrechner können beliebig vorhandene Rechenkapazität zuteilen und zeitlich limitieren. Eine Bearbeitung einer umfangreichen Simulationsstudie mit bspw. zehn Mitarbeiter-Rechnern über Nacht ist daher realistisch.

3 Beispiel: Fahrzeugteilsysteme Kühl- und Kältekreislauf

Der durchgängige Arbeitsablauf wird an der Überführung eines Systemmodells, bestehend aus zwei Fahrzeugteilsystemen des Kälte- und Kühlkreislaufs, beispielhaft durchgeführt. Das Kältekreislaufsystem ist in diesem Beispiel mit Hilfe von zwei physikalisch motivierten Modelica-Modellen aus zwei unterschiedlichen Entwicklerteams abgebildet.

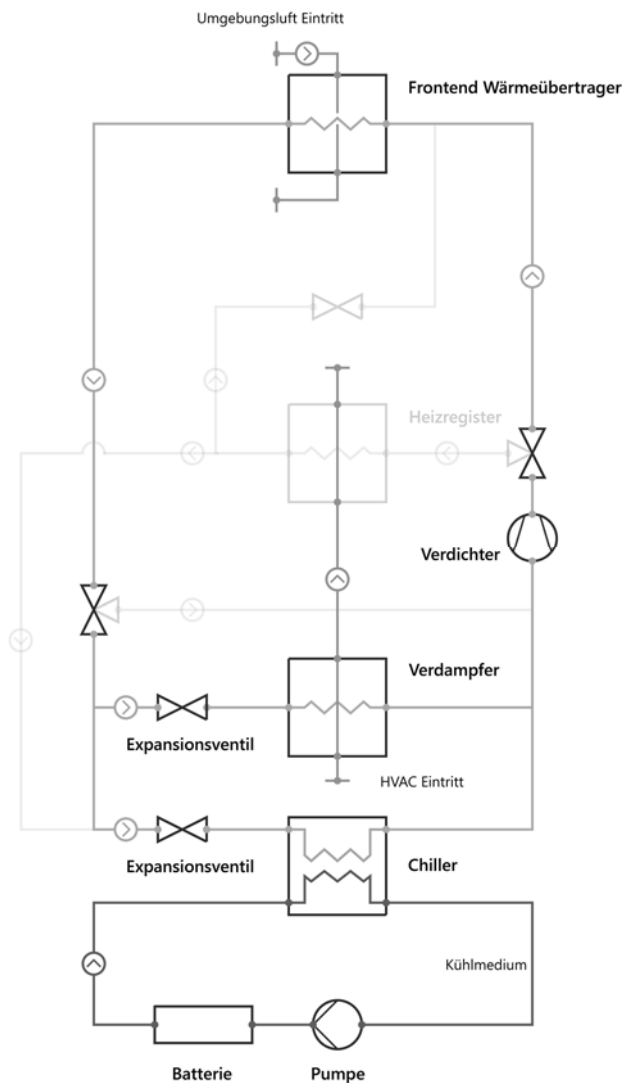


Abb. 2: Kältekreislaufmodell im Schaltzustand Fahrgast-
raumkühlung und Kühlkreislauf zum Kühlen der
Batterie.

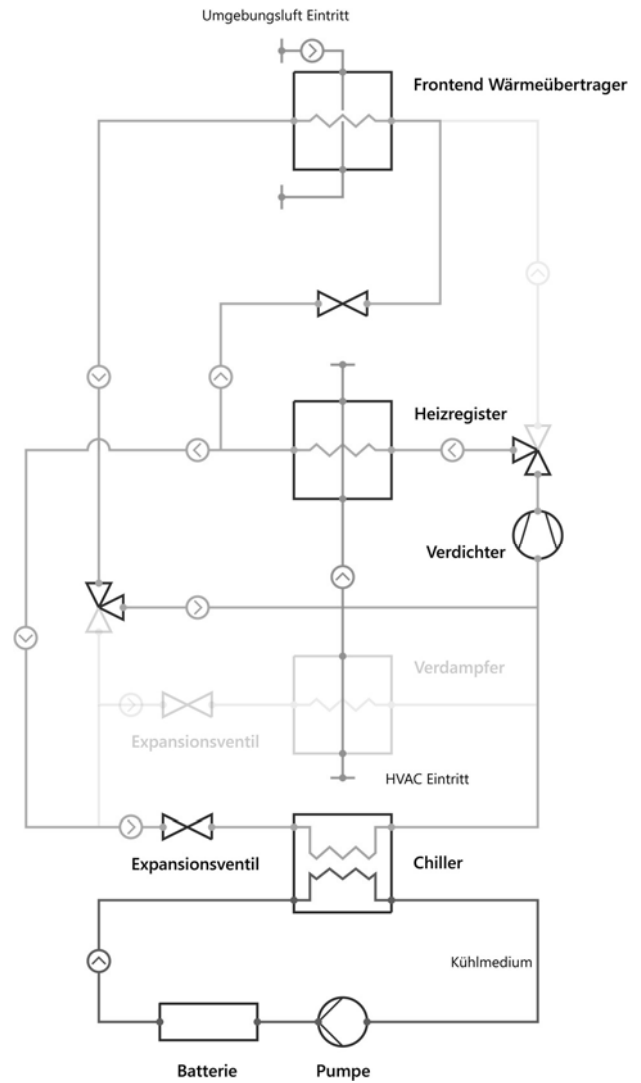


Abb. 3: Kältekreislaufmodell im Schaltzustand Fahrgast-
raumheizung und Kühlkreislauf zum Kühlen der
Batterie.

Das erste Modell umfasst Funktionen für die Kühlung des Fahrgastraums und Antriebskomponenten (siehe Abbildung 2). Das zweite Modell bildet den Schaltzustand Wärmepumpe zur Erfüllung des Heizbedarfs des Fahrgastraums ab (siehe Abbildung 3). Der Kühlkreislauf ist in der Simulationsumgebung Flowmaster abgebildet. Über den übertragenen Wärmestrom am Chiller sind die beiden Teilsysteme physisch miteinander verbunden.

Die Werte für die Stellgrößen der Systeme können für das Gesamtmodell von externen Reglern vorgegeben werden. In diesem Beispiel benötigt der Kältekreislauf die externe Vorgabe der Stellgrößen Verdichterdrehzahl und Expansionsventilquerschnitte. Die Umschaltung zwischen Heiz- und Kühlbetrieb erfolgt über die schaltbaren Ventile, die ebenfalls von extern gesteuert werden. Beim Kühlkreislauf ist die Pumpendrehzahl regelbar.

4 Erstellung von neuronalen Netzmodellen aus Co-Simulationsmodellen

Für die Umsetzung des Arbeitsablaufs stehen Tasks in der Automatisierungsumgebung als Python-Skripte zur Verfügung. Aus den vorher definierten Betriebsbedingungen wird ein Versuchsplan mit zu simulierenden Betriebspunkten generiert, die von nachgelagerten Tasks gelesen und zur Parametrierung der einzelnen Simulationen verarbeitet werden. Die Modelle des Kältekreislaufs können aus Dymola direkt als FMU exportiert werden. Die FMUs werden in einem eigenen Simulator, der in dem Python-Task programmatisch ausgeführt wird, simuliert. Modelle, die nicht als FMU exportiert werden können, werden ebenfalls über den Simulator berechnet. Der Simulator nutzt hierbei die Schnittstelle der Co-Simulations-Software. Die Ergebnisse der ca. 10.000 Einzelsimulationen werden nach Durchführung in einem nachgelagerten Arbeitsschritt automatisiert gefiltert, indem abzubildende Größen auf die Einhaltung von Betriebsgrenzen, physikalische Plausibilität und Stationarität überprüft werden. Zur Erstellung der neuronalen Netzmodelle stehen im Allgemeinen mehrere Methoden zur Verfügung. Es können durch das Automatisierungsrahmenwerk auch direkt Python-basierte Module eingesetzt werden. In dieser Arbeit wird das Training der neuronalen Netzmodelle anhand der aufbereiteten Simulationsergebnisse und der von Mathworks vertriebenen Neu-

ral Network Toolbox durchgeführt und in einem entsprechenden Task automatisiert. Für jeden Ausgabewert des Gesamtmodells wird jeweils ein neuronales Netz trainiert. Die Eingangsgrößen sind vereinfachend für jedes neuronale Netz identisch. 70 % der erzeugten Ergebnisse werden zum Trainieren der neuronalen Netze, 15 % zum Testen und 15 % zum Validieren der erzeugten Netzmodelle verwendet. Die Automatisierung ermöglicht es, eine Vielzahl von weiteren Trainingsoptionen und Modellvarianten zu testen und zu evaluieren.

Da die Trainingsdaten nicht durch Messungen erzeugt wurden, sondern durch Simulationen von physikalisch motivierten Modellen, weisen die Trainingsdaten kein Rauschen auf. Außerdem ist die Gefahr des sogenannten Overfittings durch die sehr große Anzahl an Trainingsdaten minimiert [5, 6]. Nichtsdestotrotz muss beim Trainieren der neuronalen Netze sichergestellt werden, dass die Netze ausreichend generalisiert sind. Beispielsweise kann es passieren, dass die Netze die Trainingsdaten optimal wiedergeben, jedoch Betriebspunkte, die nur minimal von den Trainingsdaten abweichen nicht mehr korrekt bestimmt werden können. Durch das Testen der neuronalen Netze an Betriebspunkten, die nicht zum Anlernen der Netze verwendet werden, wird gezeigt, dass das Netz ausreichend generalisiert. Auch sollten Extrapolationen über den Bereich, der von den Trainingsdaten abgedeckt wird, vermieden werden, da unter Umständen unplausible Ergebnisse resultieren.

Die Automatisierung gewährleistet nicht, dass der Gütegrad der trainierten neuronalen Netzmodelle besser als einzelne, mit Erfahrung und Augenmaß trainierte Modelle ist. Jedoch können verschiedenen Methoden der Erstellung der Modelle automatisiert ausprobiert werden, sodass letztendlich das Modell mit der besten Qualität ausgewählt werden kann. So können beispielsweise neuronale Netzmodelle für die gleichen Trainingsdaten mit jeweils verschiedener Anzahl von sogenannten *hidden layers* trainiert werden [7] oder beliebige Wiederholungen durchgeführt und miteinander verglichen werden.

5 Erstellung eines umschaltbaren Matlab/Simulink-Systemmodells

Die generierten, validierten neuronalen Netzmodellen werden in einem Matlab/Simulink-Modell zu einem Gesamtmodell vereint. Die Modellerstellung erfolgt über ein entwickeltes Matlab-Programm, das über das Automatisierungs-Rahmenwerk gestartet und parametrierbar wird. Das erzeugte Gesamtmodell beinhaltet alle neuronalen Netzmodelle der abzubildenden Systeme, als auch deren Verschaltungsvarianten. Die Umschaltung zwischen Systemvarianten ist im Gesamtmodell durch das Wechseln des Modellausgangs von den neuronalen Netzmodellen eines Systemzustands zum anderen abgebildet. Die Schaltzustände werden über boolesche Eingangsgrößen bestimmt, die durch externe Regelungsmodelle vorgegeben werden. Die Handhabung der Modellausgänge bei Schaltvorgängen bedarf spezieller Blöcke. Bei einem Umschaltvorgang werden die Größen über sogenannte Smooth-Transition-Funktionen vom bisherigen Ausgangswert in den Wert des aktivierten Modells in einem spezifisch parametrierbaren Zeitfenster überblendet. Das nunmehr nicht aktive Netzmodell wird eingefroren. Der Vorteil bei der Verwendung einer Smooth-Transition-Funktion gegenüber einem PT1-Glied besteht darin, dass die Dauer des Aus- und Umschaltvorgangs unabhängig von der Dynamik des Systems während des Betriebs gewählt werden kann. Beim Ausschalten des Systems werden die Ausgangsgrößen entsprechend zwei Kategorien gehandhabt:

1. Größen, die bei ausgeschaltetem System den Wert Null annehmen, wie bspw. elektr. Leistungen und Ströme, Fluidmassen- und -volumenströme
2. Größen, die bei ausgeschaltetem System definierte Stillstandswerte annehmen, wie z.B. Temperaturen, Drücke, Enthalpien

Die Dynamik der datengetriebenen Modelle wird durch Modelle in den jeweiligen Systemklassen approximiert. In diesem Fall handelt es sich um geschätzte Zeitkonstanten. Das Rahmenwerk hält grundsätzlich die Möglichkeit bereit, diese Zeitkonstanten anhand von Mess- oder Simulationsdaten anhand einer Optimierung anzupassen oder direkt dynamische neuronale Netzmodelle zu erzeugen.

Das Programm zur Erstellung des verschalteten Gesamtmodells benötigt eine Systembeschreibung. Diese beinhaltet die Zuweisung der einzelnen neuronalen Netzmodelle zu den Systemen und deren Schaltungsvarianten,

interne Datenverbindungen, die Zuteilung der Kategorien, Überblendungszeitfenster bei Umschaltung und Initialwerte. Die Systembeschreibung basiert auf einer MS Excel-Datei, die durch das Rahmenwerk vorausgefüllt und vom Nutzer vervollständigt wird.

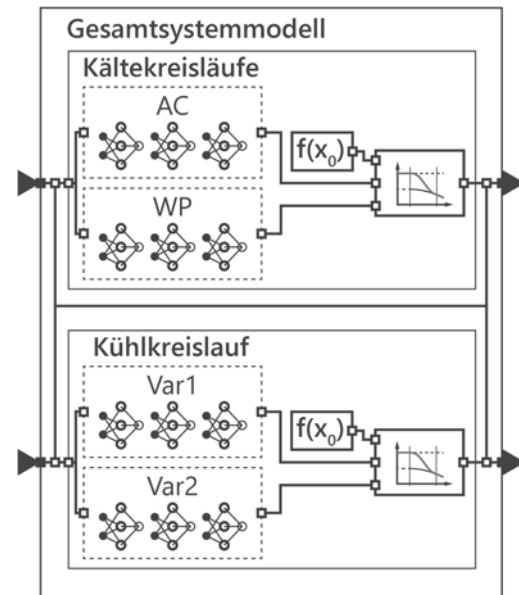


Abb. 4: Schema des Gesamtsystemmodells mit Systemmodellen und deren Varianten. Die Ausgangswerte werden bei Aus- oder Umschaltung über Smooth-Transition-Funktionen überblendet.

6 Evaluierung des Systemmodells

Das erzeugte Systemmodell wird anhand automatisierter Simulationsrechnungen und Auswertungen bezüglich der DoE-Betriebspunkte validiert und in transienten Simulationen plausibilisiert. Die automatisierte Erstellung der Ergebnisvisualisierung und der Modell-Dokumentation sowie die Ablage der Ergebnisse in bspw. Archivordnern erfolgt in einem abschließenden Arbeitsschritt. Zur Evaluierung der neuronalen Netzmodelle wird sowohl ein Teil der 10000 stationären Simulationen verwendet. Abbildung 5 zeigt Simulationsergebnisse des physikalisch motivierten Co-Simulationsmodells und die des datengetriebenen Gesamtfahrzeugmodells für die Fahrgastraumkühlung bei transienten Luftmassenströmen am Verflüssiger im Frontend des Fahrzeugs. Insgesamt zeigt das verschaltete Gesamtsystemmodell mit den datengetriebenen Modellen ein plausibles Verhalten. Jedoch ist die Modellgüte noch nicht in Gänze zufriedenstellend. In dieser Arbeit sind einige Optionen bei der Erstellung der

neuronalen Netzmodelle nicht untersucht und einige Vereinfachungen getroffen worden.

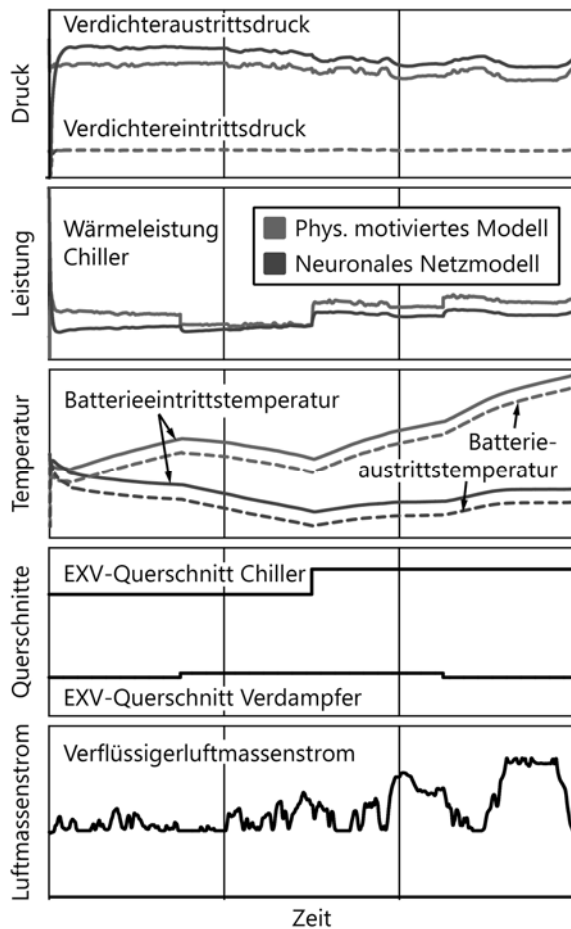


Abb. 5: Simulationsergebnisse für das Szenario Fahrgastraumklimatisierung für die physikalisch motivierten und das verschaltete Gesamtsystemmodell aus neuronalen Netzmodellen

Die Verbesserung der Modellgüte wird im weiteren Projektverlauf verfolgt. Der Schwerpunkt dieser Arbeit liegt zum aktuellen Projektfortschritt auf der Automatisierung der Arbeitsabläufe. Die erstellten neuronalen Netzmodelle sind zum Nachweis der Machbarkeit der Automatisierung ohne Anspruch auf eine zufriedenstellende Modellgüte erzeugt worden.

7 Fazit und Zusammenfassung

Das neu-entwickelte Rahmenwerk zur Automatisierung von Arbeitsabläufen ermöglicht die Einbindung von speziellen Arbeitsschritten bzw. Tasks, die für die Erstellung eines datengetriebenen Gesamtfahrzeugmodells aus einem Co-Simulationsmodell notwendig sind. Am Beispiel der zusammengefassten Fahrzeugteilsysteme Kühl- und Kältekreislauf wird der Prozess umgesetzt und durchgeführt. Der Zeitaufwand gegenüber dem bisher größtenteils manuell durchzuführenden Prozesses verringert sich signifikant. Durch das schaltbare, datengetriebene Systemmodell wird die Simulationsgeschwindigkeit deutlich erhöht. Außerdem ist durch die Möglichkeit eines FMU-Exports des verschalteten Gesamtsystemmodells die Notwendigkeit für die ursprünglich notwendigen Softwarelizenzen nicht mehr gegeben. Daher wird die Nutzung von Gesamtfahrzeugmodellen in neuen Entwicklungsbereichen ermöglicht. Durch die automatisierte Plausibilisierung, Auswertung und Berichterstellung wird die Modellqualität gesichert und dokumentiert.

References

- [1] R. Kossel. *Hybride Simulation thermischer Systeme am Beispiel eines Reisebusses*. Dissertation. TU Braunschweig, 2012
- [2] R. Kossel, C. Correia, M. Loeffler, M., M. Bodmann and W. Tegethoff. *Verteilte Systemsimulation mit TISC*. ASIM-Workshop 2009
- [3] R. Kossel, M. Loeffler, M. Bodmann and W. Tegethoff. *Verteilte Simulation von Energiesystemen im Fahrzeug*. Berechnung und Simulation im Fahrzeugbau 2010, VDI-Verlag, November 2010
- [4] MoBA Automation. <https://www.tlk-thermo.com/index.php/de/softwareprodukte/moba-lab>
- [5] A. P. Piotrowski and J. J. Napiorkowski. *A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling* J. Hydrol., vol. 476, pp. 97–111, 2013.
- [6] M. Sturm. *“Neuronale Netze zur Modellbildung in der Regelungstechnik”* München, 2000.
- [7] H. Shimodaira, *Multi-Layer Neural Networks*. 2015.

LON-Bussimulation mit SimEvents zur Auslastungs- und Fehlerabschätzung

Alexander Martens*, Olaf Simanski, Christian Bock, Olaf Hagendorf

Hochschule Wismar – University of Applied Sciences: Technology, Business and Design
Research Group Computational Engineering and Automation (CEA); <http://www.cea-wismar.de>

*Kontakt: Alexander.Martens@hs-wismar.de

Abstract. In Kooperation mit der Dr. Diestel GmbH aus Rostock, wird zurzeit an einem FuE Projekt „Entwicklung eines universellen Regelungsmoduls für die Druck- und Volumenstromregelung, sowie die Bewertung der Regelung auf Busebene“ im Bereich Reinraumtechnik, gearbeitet.

Ein Arbeitspaket innerhalb des Projektes beinhaltet die Entwicklung und Nutzung eines Simulators für die Bussimulation. Aus den zuvor mit einem Projektierungstool erstellten Projekten sollen automatisch Simulationsmodelle generiert werden, die dann auf die Projektierung rückwirken.

Die durch die Dr. Diestel GmbH erzeugten Projekte beinhalten durchschnittlich mehrere hundert Meter Leitungslänge mit mehreren hundert Teilnehmern in einem Bussystem.

Der Simulator der Hochschule Wismar, dient der Unterstützung bei diesen Projektierungen.

Die Kommunikation der einzelnen Busteilnehmer (Knoten) erfolgt mittels LON-Bus. Eine Vorhersage der erzeugten Buslast ist aufgrund der Anzahl und Komplexität der einzelnen Teilnehmer und der von ihnen versendeten Datenpakete manuell nicht möglich.

Die Simulation soll es ermöglichen, die Buslast zu schätzen, sowie eine Bewertung des Netzwerkes hinsichtlich der auftretenden Fehler und damit eine Entscheidungshilfe bei der Projektierung zu geben.

Einleitung

Im Zuge eines Forschungsprojektes arbeiten die Projektpartner Hochschule Wismar und Dr. Diestel GmbH am Projekt „Entwicklung eines universellen Regelungsmoduls für die Druck- und Volumenstromregelung, sowie die Bewertung der Regelung auf Busebene“ miteinander. Dr. Diestel GmbH ist im Bereich Lüftungs-, Klima-, Kälte- und Reinraumtechnik tätig. Um die einzelnen Bereiche stetig zu verbessern, werden Forschungen zum Beispiel im Bereich Regelung von Lüftungsanlagen durchgeführt. [1]

Die verwendeten Reglereinrichtungen werden zum Teil über den LON-Bus (Local Operating Network) an-

geschlossen. Nach der Installation von großen Netzwerken kann es aufgrund der Komplexität und Anzahl der Baugruppen, sowie der Leitungslängen zu Problemen führen, da die Buslast in der Projektierung nicht abschätzbar ist.

Um eine Unterstützung bei der Projektierung zu geben, wurde sich für die Integration eines Simulators entschieden. Bereits existierende Simulatoren für ähnliche Protokolle nutzen für die Datenübertragung unter anderem ein Protokoll der Kollisionserkennung oder auch Kollisionsvermeidung. [2, 3]

Nach Recherchen ist bislang kein Simulator mit dem genutzten Protokoll des LON-Busses verfügbar. Im folgenden Paper wird auf die Erstellung und Funktionsweise eines selbst entwickelten Simulators eingegangen.

Bei der Implementierung wurde sich für SimEvents entschieden. Mittels SimEvents und der Modellbibliothek können ereignisdiskrete Simulationen aufgebaut und Latenz, Datenrate, Fehlerrate und weitere analysiert werden. [11]

1 LON - Local Operating Network

Herkömmliche Feldbussysteme verfügen häufig über ein zentrales Leitsystem, das das Einlesen von Sensoren, Informationen bzw. Zustände von Geräten und das Steuern von Aktoren übernimmt. Einzelne Baugruppen besitzen dabei keine Intelligenz. Zustände einzelner Baugruppen werden zum Leitsystem übertragen und dort verarbeitet. Nach der Verarbeitung werden neue Sollwerte bzw. Zustände zu den einzelnen Baugruppen gesendet. Durch diese Kommunikation kann eine hohe Busauslastung des vorliegenden Netzes entstehen. [4]

Um die Buslast zu senken, werden im LON-Netz intelligente Knoten verwendet. Diese Knoten kom-

munizieren mit den angeschlossenen Baugruppen und entscheiden selbständig, entsprechend ihrer Vorgaben, über neue Zustände. Zusätzlich können beispielsweise Regelungen knotenlokal umgesetzt werden. Der Einsatz eines Leitrechners ist optional.

1.1 Allgemein

Der LON-Bus wurde 1990 entwickelt und verwendet das Protokoll LonTalk. Als Netzstrukturen können einfache Ring-, Stern- oder Linien- bis hin zu komplexen Mischstrukturen verwendet werden. Die maximale Buslänge beträgt für Linienstrukturen 2700m (siehe Figure 1), für freie Netzstrukturen 500m (siehe Figure 2). Bei Linienstrukturen ist darauf zu achten, dass ein maximaler Abstand von 3m der einzelnen Stiche nicht überschritten wird. Pro Segment können 64 passive Knoten (ohne eigenes Netzteil) bzw. 128 aktive Knoten (mit eigenem Netzteil) eingesetzt werden. Zur Vermeidung bzw. Dämpfung von Signalreflexionen an den Leitungsenden müssen diese mit Abschlusswiderständen (Terminatoren) versehen werden (siehe Figure 1 und Figure 2 Terminator). [5]

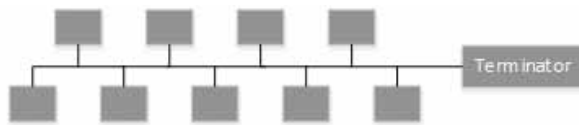


Abbildung 1: Beispiel für eine Linienstruktur

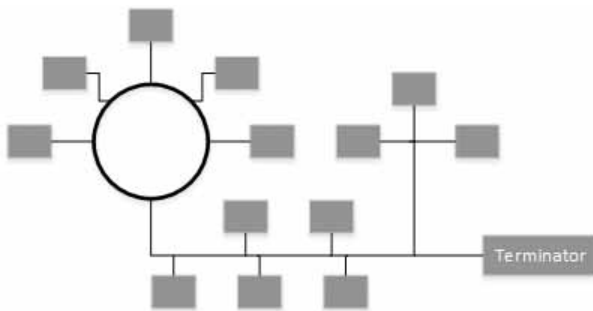


Abbildung 2: Beispiel für eine freie Netzstruktur

Für den LON-Bus nutzbare Übertragungsmedien sind unter anderem verdrehte Zweidrahtleitungen (Twisted Pair – kurz: TP), Niederspannungsnetze (Powerline) und Funk. Die maximale Paketgröße (249 Bytes) kann bei einer Übertragungsgeschwindigkeit zwi-

schen 5kbit/s mit Powerline und 1,25Mbit/s auf einer TP Linie mit maximal 130m Länge betragen. Eine gängige Geschwindigkeit bei einer TP Verkabelung ist 78kbit/s. Die Signalübertragung erfolgt auf Grundlage des Differential-Manchester-Code. [6]

1.2 Buszugriffsverfahren

LonTalk nutzt ein prädiktives p-persistentes CSMA-Verfahren zum Versenden erzeugter Datenpakete. Das CSMA-Verfahren steuert den Mehrfachzugriff mit Trägerprüfung gleichberechtigter Teilnehmer auf einem gemeinsam genutzten Medium. Datenpakete werden unter Berücksichtigung der Wahrscheinlichkeit eines prädiktiven vorläufigen Datenaufkommens versandt.

Die Wartezeit (t_w) zwischen den einzelnen Sendeversuchen wird in Slots unterteilt. Sie besteht aus einem festen Ruheslot β_1 , 128 optionalen Prioritätsslots (m) und bis zu 1008 Zufallslots (n) (siehe Figure 3).

Die Länge des Ruheslot β_1 ist gerätespezifisch und setzt sich zusammen aus der internen Datendekodierungsmethode (1 bit time), physikalische Ausbreitungsverzögerung (t_p) und der internen Verarbeitungszeit des MAC-Controllers (t_m) (siehe Formel 1). Die interne Datendekodierungsmethode garantiert das Senden und Empfangen von Daten zu jeder Bitzeit. Die Ausbreitungsverzögerung wird durch die physikalische Übertragung der Medienlänge definiert. Die interne Verarbeitungszeit des MAC-Controllers definiert die Zeit von der Überprüfung des nicht belegten Busses bis hin zur ersten Ausgabe der Daten. Die β_2 -Slots haben bei einer Geschwindigkeit von 78125Bit/s eine feste Größe von 5,6Bit. Was eine Wartezeit von 71,68 μ s pro Slot ergibt. [7, 8]

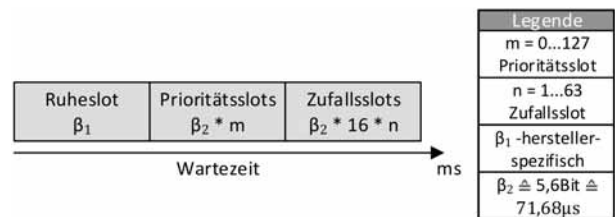


Abbildung 3: Übersicht Wartezeit

$$\beta_1 > 1 \text{ bit time} + (2 * t_p + t_m) \tag{1}$$

Versucht ein Knoten zu senden, hört dieser nach der Wartezeit den Bus ab. Stellt der Knoten dabei keine Busaktivität fest, beginnt der Knoten mit dem Versand

des Datenpaketes. Wird hingegen Busaktivität festgestellt, wird das Senden ausgesetzt und die Wartezeit erneut berechnet.

Wurde einem Knoten eine Priorität zugeordnet, dann besteht die Wartezeit (siehe Formel 2) aus dem Ruheslot und dem Prioritätsslot basierend auf der zugewiesenen Priorität. Wurde dem Knoten keine Priorität zugeordnet (siehe Formel 3), wird in Summe der Ruheslot, die maximale Länge des Prioritätsslots und bis zum zufällig gewählten Zufallslot gewartet. Der Zufallslot wird anhand einer prädiktiven Buslast geschätzt. Dazu hört jeder Knoten den Datenverkehr am Bus mit und bestimmt anhand der Informationen in den aufgenommenen Datenpaketen, wie viele Knoten in unmittelbarer Zukunft voraussichtlich senden werden.

$$t_w = \beta_1 + \beta_2 * m \tag{2}$$

$$t_w = \beta_1 + \beta_2 * 127 + \beta_2 * 16 * n \tag{3}$$

Der Vorteil der Verwendung von Prioritäten für Knoten besteht darin, dass diese vor Knoten ohne oder niedrigerer Prioritäten senden können. Um zu vermeiden, dass Knoten mit Priorität den Bus mit Datenpaketen auslasten, lässt das LonTalk-Protokoll nur jede zweite Nachricht eines Knoten als priorisiertes Datenpaket zu. [6]

Dieses Prinzip kann die Häufigkeit von Kollisionen minimieren, aber nicht ausschließen. Im Falle einer Kollision, beenden die sendenden Knoten den Sendevorgang eigenständig und beginnen nach einer Wartezeit mit einem erneuten Sendeversuch. [4, 9]

2 Umsetzung eines LON-Bus-Simulator mit SimEvents

Um Buslast und Fehlerrate in LON-Bus-Systemen mit vielen Knoten abzuschätzen, wird eine Modellsimulation in SimEvents entwickelt. Somit können mögliche Überlastungen schon bei der Projektierung erkannt und behoben werden.

Für die Erstellung des Simulationsmodells sind die Elemente Terminator, T-Abzweigung, Leitung und Knoten (bestehend aus MAC-Controller und Applikation) notwendig (siehe Figure 4).

Aufbauend auf einem Mathworks-Beispiel von Anuja Apte zum CSMA/CD Verfahren [2] konnten die Elemente Terminator und T-Abzweigung ohne weitere Anpassung übernommen werden. Die weiteren Ele-

mente wurden an die Besonderheiten des LON-Bus angepasst bzw. erweitert.

Figure 4 zeigt ein Beispiel eines Netzwerkes mit drei Knoten. Die grauen Blöcke sind für den Ablauf der Modellsimulation notwendig und können nicht editiert werden. Die weißen Blöcke sind applikationsspezifisch und werden zu Beginn hinsichtlich der gewünschten Funktionsweise parametrisiert.

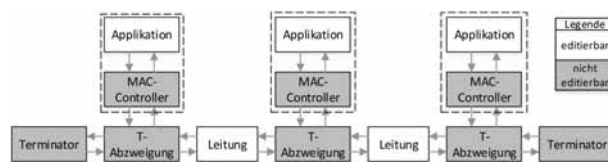


Abbildung 4: Beispiel eines LON-Netzwerkes mit 3 Bus-Knoten

2.1 Modellkomponenten

Im Folgenden wird auf Elemente in der Simulation hinsichtlich Besonderheiten und Funktionsweise eingegangen.

Terminatoren werden an den Leitungsenden positioniert und löschen ankommende Datenpakete (siehe Figure 5).

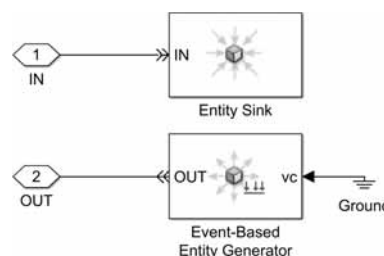


Abbildung 5: Terminator

Um Knoten mit dem Bus zu verbinden, werden T-Abzweigungen eingesetzt. Diese vervielfältigen intern die ankommenden Datenpakete und leiten diese an die Ausgänge weiter (siehe Figure 6).

Die Leitung, d.h. der damit verbundene Längenparameter, ruft eine zur Länge proportionale zeitliche Verzögerung der Datenpakete hervor. Die Verzögerungszeit wird für beide Richtungen in den Single Servern 1 und 2 (siehe Figure 7 – grau) eingestellt. Zusätzlich ist eine Kollisionserkennung im Leitungselement implementiert.

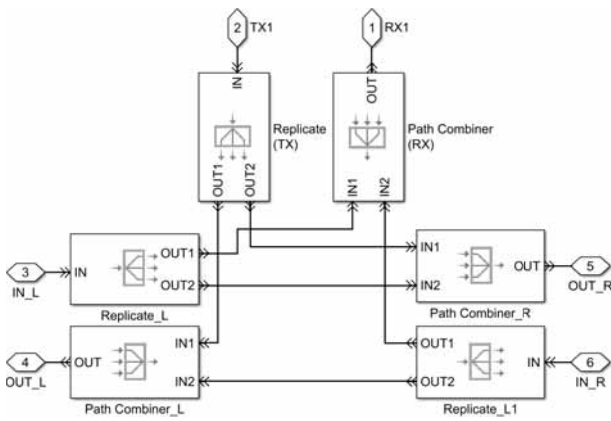


Abbildung 6: T-Abzweigung

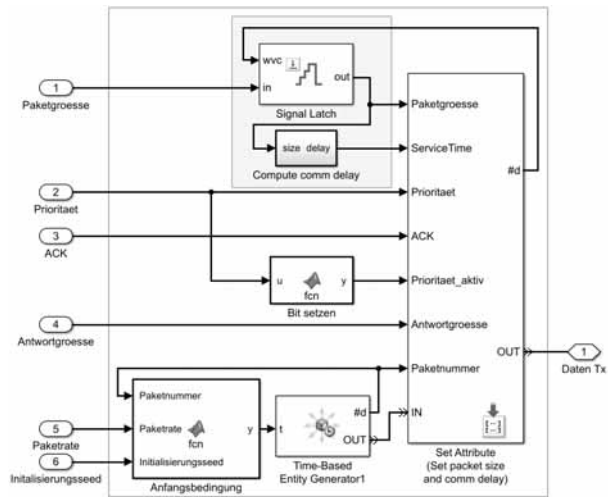


Abbildung 8: Applikation - Paketgenerierung

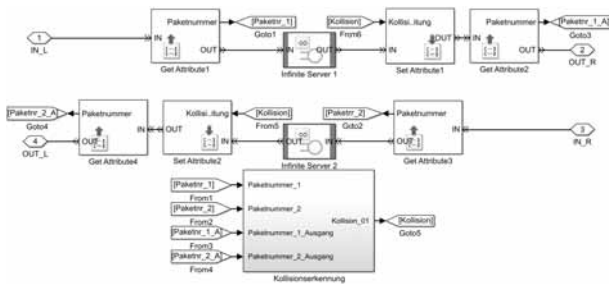


Abbildung 7: Leitung (interner Aufbau)

Die Kollisionserkennung ist notwendig, da sich nur ein Datenpaket gleichzeitig auf dem Bus befinden kann. Beim gleichzeitigen Versand mehrerer Datenpakete auf dem Bus kommt es in den jeweiligen Leitungsabschnitten zu Kollisionen (siehe Kapitel 2.3).

Die Applikationen stellen die verschiedenartigen Teilnehmer mit unterschiedlichen Arten und Anzahlen von Datenpunkten dar und erzeugen Datenpakete in zuvor eingestellten Zeitenintervallen. Für die Simulation werden die nötigen Datenpunkte aus der Projektierung in die entsprechenden Knoten und Leitungslängen übernommen. Dazu zählen Paketgröße, Sendeintervall, Priorität und Paketquittierung (siehe Figure 8).

Die Paketgröße gibt an, wie viele Bytes in einem Datenpaket enthalten sind, die vom Knoten versandt wird. Weiter wird aus der Paketgröße die Abarbeitungszeit (ServiceTime) des jeweiligen Datenpaketes berechnet und als Attribut im versendeten Datenpaket gesichert. Mit dem Sendeintervall wird festgelegt wie viele Datenpakete in einer Sekunde erzeugt und über den Bus versendet werden. Mittels Priorität können Datenpakete bevorzugt behandelt werden. Ausgesendete Datenpa-

kete können optional vom Empfänger quittiert werden. Die verschiedenen Varianten der Quittierung sind: einmaliges Senden ohne Acknowledgement (ACK), dreimaliges Senden ohne ACK, Senden mit ACK und optionalen anwendungsspezifischen Daten. Bei der Verwendung von ACK mit anwendungsspezifischen Daten wird die Größe der Zusatzdaten mit in der sendenden Applikation angegeben. Dies hat zur Folge, dass die Größe der Antwort variieren kann. Werden Datenpakete zum Start der Simulation zur selben Zeit in den Applikationen erzeugt und versandt, kollidieren diese in den Leitungssegmenten. Um dieses zu verhindern wird in jeder Applikation ein Initialisierungsseed gesetzt, der einen zeitlichen Offset auf die Paketgenerierung beaufschlagt.

Zusätzlich zum Erzeugen von Datenpaketen überprüft die Applikation, ob der Sender eines ankommenden Datenpaketes eine Antwort (ACK, ACK mit zusätzlichen Daten) oder nicht erwartet. Wird kein ACK erwartet wird das Datenpaket gelöscht. Wird ein ACK erwartet, wird eine Kopie des eingegangenen Datenpaketes an den Absender mit geänderten Attributen zurückgeschickt. Dafür wird im ersten Schritt dem empfangenden MAC-Controller mit einer Variable signalisiert, dass es sich bei dem Datenpaket um eine Antwort handelt. Danach wird eine interne Verarbeitungszeit gesetzt, die angibt wie lange der Knoten für die Bearbeitung des ankommenden Datenpaketes benötigt. Die Zeit ist gerätespezifisch und wurde auf 15ms gesetzt. Im nächsten Schritt wird die Antwortgröße aus dem Datenpaket ausgelesen und als neue Größe des Datenpakets

gesetzt. Dann wird es an den MAC-Controller übergeben, der das Datenpaket auf den Bus legt (siehe Figure 9).

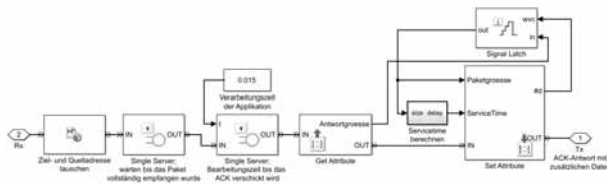


Abbildung 9: ACK-Antwort mit zusätzlichen Daten

Der MAC-Controller implementiert das prädiktive p-persistentes CSMA Verfahren und realisiert das Lon-Talk Protokoll. Dazu wurden der MAC-Controller in Empfangs- (siehe Figure 10 - oben) und Sendeteil (siehe Figure 10 - unten) aufgeteilt. Zusätzlich verfügt der MAC-Controller über die Möglichkeit, Kollisionen zu detektieren und zu behandeln (siehe Figure 10 - mittig).

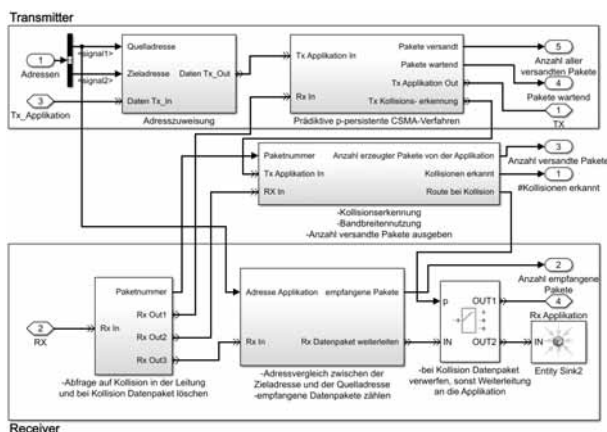


Abbildung 10: MAC-Controller (interner Aufbau)

Im Empfangsteil (Receiver) wird im ersten Schritt überprüft, ob das ankommende Datenpaket im Laufe der Übertragung mit einem weiteren Datenpaket kollidierte. Fand eine Kollision statt, wird das Datenpaket verworfen und ein Fehlerzähler wird inkrementiert.

Weiter wird geprüft, ob das Datenpaket für den MAC-Controller mit der entsprechenden Zieladresse vorgesehen ist. Stimmt die Zieladresse nicht mit der des MAC-Controllers überein, wird das Datenpaket verworfen. Bei Übereinstimmung der Adressen wird das Datenpaket an die Applikation weitergeleitet.

Im Sendeteil (Transmitter) wird das Datenpaket, das zuvor mit den gewünschten Parametern in der Applikation generiert wurde, verarbeitet. Im MAC-Controller

erfolgt zu Beginn eine Adresszuweisung. Handelt es sich bei dem Datenpaket um eine Antwort, die an den ursprünglichen Sender zurückgesendet wird, werden Ziel- und Quelladresse getauscht. Bei neuen Datenpaketen hingegen werden die Wunsch-Zieladresse und Quelladresse hinzugefügt. Danach wird das Datenpaket an den „Prädiktive p-persistente CSMA-Verfahren“ Block weitergeleitet (siehe Kapitel 2.2).

Nach Abarbeitung der prädiktiven p-persistenten CSMA Steuerung, wird das Datenpaket auf dem Bus an die weiteren angeschlossenen Knoten verteilt. Um zu signalisieren, dass der Bus von einem Datenpaket belegt wird, werden ankommenden Datenpakete anhand ihrer Größe und der Servicetime in einem Single Server abgearbeitet. Die Belegung des Single Servers kann von der CSMA Steuerung abgefragt werden. [6]

2.2 Prädiktives p-persistentes CSMA-Verfahren

Das prädiktive p-persistente CSMA-Verfahren zum Versenden erzeugter Datenpakete besteht aus einer Sequenzsteuerung, Berechnung der erwarteten Buslast und dem CSMA-Verfahren selbst (siehe Figure 11).

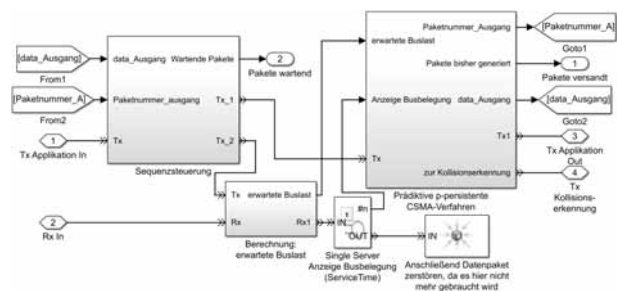


Abbildung 11: Prädiktive p-persistente CSMA-Verfahren

Die Sequenzsteuerung (siehe Figure 12) ist notwendig, damit nur ein Datenpaket gleichzeitig durch das CSMA-Verfahren abgearbeitet wird, auch wenn die Applikation beispielsweise mehreren Teilnehmern mit einem ACK antworten will, es aber aufgrund erhöhter Buslast nicht unmittelbar möglich ist. Die Sequenzsteuerung gibt den Bus frei, wenn ein Datenpaket die Steuerung verlässt.

Ein Indikator für eine hohe Buslast ist in der Sequenzsteuerung der Ausgang „Wartende Pakete“. Dieser signalisiert das das Datenpaket nicht über den Bus versandt werden konnte und in der Sequenzsteuerung auf Abarbeitung wartet. Ein zeitliches Verwerfen von Da-

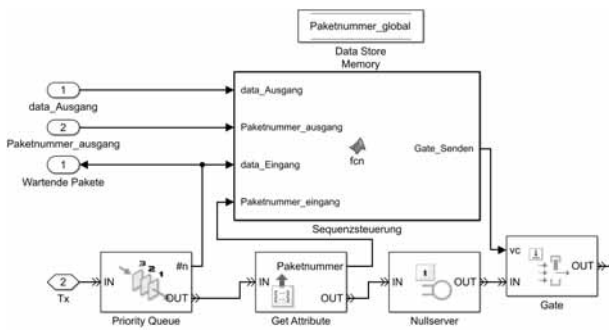


Abbildung 12: Sequenzsteuerung im MAC-Controller

tenpaketen die sehr lange in der Priority Queue liegen ist derzeit nicht implementiert (siehe Figure 12).

Befindet sich ein Datenpaket in der CSMA-Steuerung (siehe Figure 13) wird im ersten Schritt anhand der erwarteten Buslast die Wartezeit berechnet und gewartet. Parallel werden ankommende Datenpakete anhand ihrer Servicezeit in einem Single Server abgearbeitet und signalisieren einen belegten Bus. Nach Ablauf der Wartezeit wird im Output Switch überprüft, ob der Bus belegt ist. Ist der Bus frei, wird das Datenpaket versendet. Wird hingegen ein belegter Bus festgestellt, wird die Wartezeit anhand der erwarteten Buslast erneut berechnet und erneut gewartet. Dieser Vorgang wird solange wiederholt, bis das Datenpaket erfolgreich versendet werden konnte.

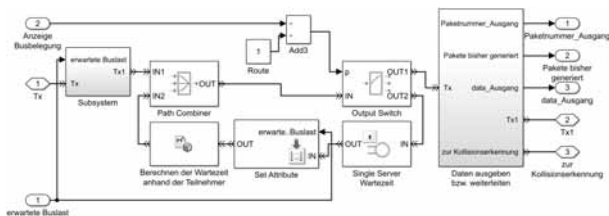


Abbildung 13: Prädiktives p-persistente CSMA Steuerung

Die erwartete Buslast wird parallel zur CSMA Steuerung in einem Stateflowmodell berechnet. Hierfür werden Informationen aus den Datenpaketen, die auf dem Bus versandt werden, gesichert und ausgewertet. Dazu zählen die Informationen der Paketquittierungen (ohne ACK, mit ACK, mit ACK und zusätzlichen Daten), Ziel- und Quelladressen. Aus diesen Informationen wird abgeschätzt, wie viele Teilnehmer in naher Zukunft anderen Teilnehmern antworten werden. Um die Vorhersage aktuell zu halten, verfallen gesicherte Informationen bei Nichtbenutzung nach einer vorher definierten Zeit.

nierten Zeit.

2.3 Kollisionserkennung

Um Kollisionen festzustellen wurden zwei Kollisionserkennungen in den Elementen Leitung und im Element MAC-Controller implementiert.

Im MAC-Controller findet parallel zum Sendee- und Empfangsteil eine Kollisionserkennung statt. Diese überprüft, ob die Applikation ein neues Datenpaket versendet, während über den Bus ein bereits existierendes Datenpaket im MAC-Controller eintrifft. Dieser Fall hätte eine Kollision zur Folge und beide Datenpakete werden daraufhin verworfen.

Die Kollisionserkennung in den Leitungen ist nötig, wenn mehrere Datenpakete gleichzeitig in den Leitungen verzögert wurden und diese Kollision aufgrund der Leitungslänge nicht im MAC-Controller detektiert werden kann. Für die Feststellung, dass sich zwei Datenpakete gleichzeitig in der Leitung befinden, wird ein Stateflow-Modell verwendet (siehe Figure 14). Jedes erzeugte Datenpaket enthält eine vorher festgelegte Paketnummer. Die Paketnummern dienen als Indikatoren für das Eintreffen und Verlassen von Datenpakete im Leitungssegment.

Im Stateflow-Modell wurde für jede Richtung eine „belegt“ Variable definiert. Beim Betreten der Leitung wird die zugehörige Richtungsvariable gesetzt. Nach Ablauf der Verzögerungszeit im jeweiligen Single Server verlässt das Datenpaket die Leitung und setzt dabei die Variable zurück. Sind beide Variablen gleichzeitig gesetzt, entspricht dies einer Kollision, da nicht mehr als ein Datenpaket gleichzeitig in der Leitung aktiv sein kann. Nachdem die Kollision erkannt wurde, werden die Datenpakete markiert, damit diese im MAC-Controller erkannt und verworfen werden.

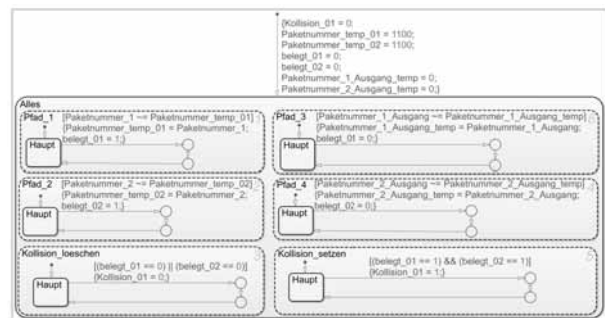


Abbildung 14: Stateflow-Modell

3 Modellvalidierung

Nachdem der Simulator mit dem prädiktiven p-persistentes CSMA-Verfahren einen nutzbaren Stand erreicht hatte, wurden Stabilität und Verhalten mit einer festen Anzahl von Knoten und unterschiedlichen Einstellungen getestet. Hierfür wurden identische Systeme aufgebaut, parametrisiert, ausgemessen und mit den Simulationsergebnissen verglichen.

Im Folgenden werden zwei reale Systeme mit unterschiedlichen Spezifikationen näher betrachtet und auf das Vorgehen bei der Validierung der Simulation genauer eingegangen.

3.1 Versuchsaufbauten

Im ersten Aufbau wurden auf Geräte von Dr. Diestel GmbH zurückgegriffen, die in Linie verdrahtet wurden. Die Länge zwischen den Baugruppen konnte durch Steckverbinder verlängert bzw. verkürzt werden. Als Geräte wurden ein Raumluft-, ein Raumabluftregler und ein Temperaturfühler mit Steuerungsfunktionalität verwendet. [16, 17]

Der zweite Aufbau wurde mit drei Embedded-Busnoten der Firma Deutschmann durchgeführt [15]. Diese können frei programmiert werden. Sie ermöglichen die Entwicklung eigener Programme mit denen unterschiedliche Buslasten erzeugt werden können.

In Figure 15 sind die Aufbauten schematisch dargestellt. Die einzelnen Busknoten sind die Geräte 1 bis 3. Der NIC-USB ist die Schnittstelle zum PC. [13]

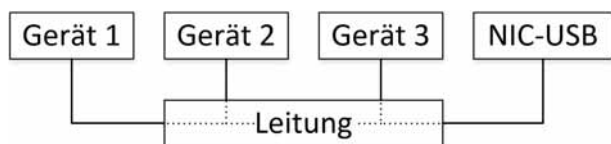


Abbildung 15: Testaufbau (schematisch)

3.2 Datengewinnung

Um die internen Variablen der Knoten zu parametrieren wurde die Software LonMaker, sowie die Hardware NIC-USB verwendet. Mit dem LonMaker können die internen Variablen der Busknoten einzeln, in Gruppen oder auch alle zyklisch gepollt werden. Der NIC-USB verbindet einen PC mit dem LON-BUS. Für die Aufzeichnung der auftretenden Buslast wurde das Programm LPA (LOYTEC Protocol Analyzer) genutzt.

Mit dem Protokollanalysator können die aufgezeichneten Daten als CSV-Dateien gesichert werden. Dieses ermöglicht eine nachträgliche Auswertung. [12, 14]

Im ersten Aufbau wurde mit dem LonMaker die internen Variablen gepollt. Das Pollintervall wurde bei diesem Versuchsaufbau verändert. Im zweiten Aufbau wurde die Programmierfähigkeit der Busknoten genutzt. Hierbei wurden die Embedded-Busknoten so programmiert, dass diese zyklisch Daten zwischen den weiteren Embedded-Busknoten versenden. Die erzeugten Daten in beiden Aufbauten wurde mitgeloggt und ausgewertet.

Die Mitschnitte beinhalten Quell-, Zieladressen, Zeitstempel, Paketgrößen, Prioritäten und Paketquittierungen. Die CSV-Dateien wurden mit MATLAB eingelesen und die einzelnen Informationen anhand der Quelladressen den einzelnen Applikationen zugeordnet. Für jede Applikation wurden die durchschnittliche Größe der Daten, Zeitstempel der Datenpakete, Mittelwert der Zeiten zwischen den einzelnen Aufrufen und die gesamte Messzeit gesichert. [10]

Die extrahierten Informationen der MATLAB-Auswertung werden automatisch in die Simulation übernommen und ausgeführt. Hierfür wurden die Applikationen und die Leitungslängen parametrisiert. In Figure 16 ist der Simulator mit SimEvents dargestellt. Zur Busgeschwindigkeit werden neben auftretenden Fehlern, wartenden Paketen, die Anzahl der erzeugten Datenpakete dargestellt. Mit den angezeigten Informationen können Rückschlüsse auf Buslast und Fehlerrate erfolgen.

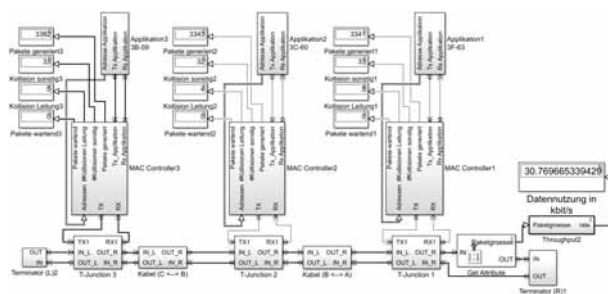


Abbildung 16: Programm LonMaker mit den Baugruppen von Deutschmann

3.3 Auswertung und Fehlerbetrachtung

Im ersten Versuchsaufbau wurden 56 Versuche durchgeführt. Mit dem LonMaker wurde eine maximal Bus-

last von 7,483kBit/s (ca. 10%) gemessen. Die Abweichung zwischen dem realen System und der Simulation liegen zwischen 0,036% und 5,026%. Der mittlere Fehler der einzelnen Versuche beträgt 1,5772. Die Standardabweichung beträgt 1,302 (siehe Figure 17).

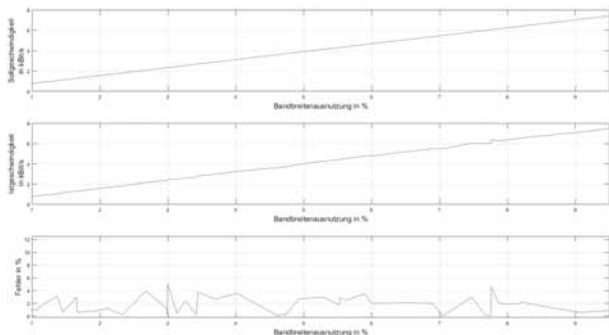


Abbildung 17: Ergebnis: Versuchsaufbau 1

In Figure 17 sind die Ergebnisse des zweiten Versuchsaufbaus dargestellt. Sie stellt die Soll-, Istgeschwindigkeit und den Fehler in Prozent der Busauslastung gegenüber. Zu erkennen sind Schwankungen des Fehlers von 0,036% bis 5,026% über die gesamte betrachtete Bandbreite. Eine Ursache der Schwankungen ist ein früher Stand des Simulators. Dieser wurde mit fortlaufenden Test optimiert.

Im zweiten Versuchsaufbau wurde die Anzahl der Versuche konstant gehalten. Durch das Senden von Datenpaketen untereinander konnte eine maximale Buslast mit 31,114kBit/s gemessen werden. Die Abweichung zwischen dem realen System und der Simulation liegen zwischen 0,337% und 11,96%. Der mittlere Fehler der einzelnen Versuche beträgt 1,2118. Die Standardabweichung beträgt 3,177 (siehe Figure 18).

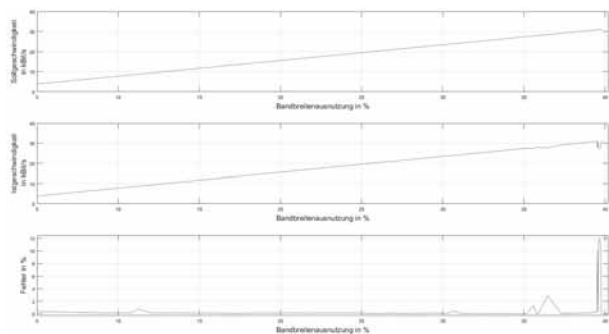


Abbildung 18: Ergebnis: Versuchsaufbau 2

In Figure 18 sind die Ergebnisse des zweiten Ver-

suchsaufbaus dargestellt. Sie stellt die Soll-, Istgeschwindigkeit und den Fehler in Prozent der Busauslastung gegenüber. Bis ca. 35% Bandbreitenausnutzung sind die Abweichungen zwischen dem realen System und der Simulation kleiner als 1%. Ab 35% Bandbreitenausnutzung steigt der Fehler auf bis zu 12%. In der Simulation ist zu erkennen, dass der Zähler für die Anzahl der wartenden Paket steigt und nicht alle erzeugten Datenpakete über den Bus versandt werden können. Dies lässt sich auf einen ausgelasteten Bus zurückführen.

In Tabelle 1 werde die beiden Versuchsaufbauten gegenübergestellt. Nach anpassen des Simulators konnte der mittlere Fehler von Versuchsaufbau 1 zu Versuchsaufbau 2 verringert werden.

	Versuchsaufbau 1	Versuchsaufbau 2
maximale Geschwindigkeit	7,483kBit/s	31,114kBit/s
minimaler Fehler	0,036%	0,337%
maximaler Fehler	5,026%	11,96%
mittlerer Fehler	1,5772	1,2118
Standardabweichung	1,302	3,177

Tabelle 1: Darstellung Ergebnisse der Versuchsaufbauten

Mit dem 2. Versuchsaufbau konnte eine maximale Busauslastung von 40% empirisch ermittelt werden.

4 Fazit und Ausblick

Das Paper beschreibt den Aufbau und die Funktionsweise eines Simulators für den LON-Bus mit dem prädiktiven p-persistente CSMA-Verfahren.

Mit dem Simulator wurden Tests mit 2 Versuchsaufbauten für zu bis 10% und bis zu 40% Busauslastung durchgeführt.

Die ermittelten Ergebnisse zeigen, dass der Simulator bis 35% Bandbreitenausnutzung geringe Abweichungen zum realen System aufweist. Ab 35% steigt die Abweichung sprunghaft auf max.12% an. In der Si-

mulation ist zu erkennen, dass nicht alle erzeugten Datenpakete versandt werden können, was auf einen ausgelasteten Bus zurückzuführen ist.

Der Simulator kann für die erste Einschätzung von LON-Netzwerken hinsichtlich Fehlerrate und Busauslastung genutzt werden.

Die nächsten Schritte beinhalten weitere Optimierungen des Simulators und die automatisch Modellgenerierung. Die Modellgenerierung soll es ermöglichen Simulationsmodelle automatisch aus der Projektkonfiguration zu erzeugen und diese zu simulieren. Mit den Simulationsergebnissen kann die erwartete Buslast geschätzt, sowie eine Bewertung hinsichtlich der auftretenden Fehler erfolgen.

Literatur

- [1] Dr. Diestel GmbH, *Unternehmensprofil*, <https://www.dr-diestel.de/unternehmensprofil/>, Abruf: 01.08.2018
- [2] A. Apte, *Modeling System Architecture and Resource Constraints Using Discrete-Event Simulation*, <https://de.mathworks.com/company/newsletters/articles/modeling-system-architecture-and-resource-constraints-using-discrete-event-simulation.html>, Abruf: 01.08.2018
- [3] MathWorks, *ALOHA and CSMA/CA Packetized Wireless Networks*, <https://de.mathworks.com/help/comm/examples/aloha-and-csma-ca-packetized-wireless-networks.html>, Abruf: 01.08.2018
- [4] LON Nutzer Organisation e. V., bfe-Oldenburg, *LonWorks – Installationshandbuch 2. Auflage*, Publisher: VDE VERLAG GMBH, Berlin und Offenbach 2004
- [5] H. Merz, T. Hansemann, Ch. Hübner, *Gebäudeautomation*, Carl Hanser Verlag München 2007
- [6] LON Nutzer Organisation e. V., D. Dietrich (Hrsg.), P. Fischer (Hrsg.), *LONWORKS-Planerhandbuch*, VDE Verlag, Berlin und Offenbach 2001
- [7] ECHELON, *LonTalk Protocol Specification*, <http://www.enerlon.com/JobAids/Lontalk%20Protocol%20Spec.pdf>, Version 3.0, Abruf: 01.08.2018
- [8] ECHELON, *Introduction to the LonWorks Platform*, https://www.echelon.com/assets/blt893a8b319e8ec8c7/078-0183-01B_Intro_to_LonWorks_Rev_2.pdf, Version 2.0, Abruf: 01.08.2018
- [9] D. Dietrich, D. Loy, H. Schweinzer (Hrsg.), *LON-Technologie 2. überarbeitete Auflage*, Hüthig GmbH, Heidelberg 1999
- [10] MathWorks, *MATLAB Version 9.2.0 - R2017a*, <https://de.mathworks.com/products/matlab.html>, Abruf: 01.08.2018
- [11] MathWorks, *SimEvents - Model and simulate discrete-event systems Version: 5.2*, <https://de.mathworks.com/products/simevents.html>, Abruf: 01.08.2018
- [12] LOYTEC, *LPA Version: 3.7.2*, <https://www.loytec.com/de/produkte/software/lpa>, Abruf: 01.08.2018
- [13] LOYTEC, *NIC-USB (NIC709-IP1E100C)*, <https://www.loytec.com/de/produkte/router-nic/nic>, Abruf: 01.08.2018
- [14] ECHELON, *LonMaker Version 3.22.12 SP2*, <https://www.metz-connect.com/de/products/11021603e>, Abruf: 01.08.2018
- [15] ECHELON, *UNIGATE IC LonWorks*, <https://www.deutschmann.de/de/produkte/embedded-solutions/unigate-ic/unigate-ic-lonworks/>, Abruf: 01.08.2018
- [16] TROX GmbH, *LonWorks-Schnittstelle für EASYLAB Regler und Adaptermodule*, <https://www.trox.de/erweiterungsmodule/serie-em-lon-f8b9912d13b5c6aa>, Abruf: 01.08.2018
- [17] Kieback&Peter GmbH & Co. KG, *technolon Room Module*, <https://extranet.kieback-peter.de/download/110-51110-81-en.pdf?23015>, Abruf: 01.08.2018

Leitungsmodell für Bordnetzuntersuchungen

Leonard Gysen¹, Joachim Haase²

¹ Universität Kassel, Fachgebiet Fahrzeugsysteme und Grundlagen der Elektrotechnik, Wilhelmshöher Allee 73, 24109 Kassel; gysen@uni-kassel.de

² Fraunhofer IIS/Entwicklung adaptiver Systeme (EAS), Zeunerstr. 38, 01069 Dresden; joachim.haase@eas.iis.fraunhofer.de

Abstract. Aktuelle Entwicklungen im Automobilbau sind durch eine zunehmende Komplexität der Elektrik/Elektronik (E/E)-Systeme gekennzeichnet. Das hat Auswirkungen auf die Gestaltung des Bordnetzes von Fahrzeugen. Zur Untersuchung unterschiedlicher Bordnetzkonzepte wurde unter Nutzung der universellen Verhaltensbeschreibungssprache Modelica eine Modellbibliothek erstellt. Diese Bibliothek enthält u.a. Leitungsmodelle, die neben der Berücksichtigung thermisch-elektrischer Wechselwirkungen auch die Aufbereitung von Aussagen zu nichtfunktionalen Merkmalen wie Masse und Preis unterstützen. Die Modelle sind für den Vergleich unterschiedlicher Bordnetzarchitekturen mittels Simulation verwendet worden.

Einleitung

Die Entwicklung im Automobilbau ist aktuell durch Richtungen wie die Elektrifizierung des Antriebsstrangs, die Vernetzung innerhalb des Fahrzeugs und zwischen Fahrzeug und Umgebung sowie Aktivitäten zur Automatisierung von Fahrfunktionen gekennzeichnet. Erprobte Sicherheitskonzepte setzen auf das korrigierende Eingreifen des Fahrers. Bei künftigen steigen die Anforderungen an die für einen sicheren Betrieb entscheidenden Komponenten des E/E-Systems. So muss deren Funktionsfähigkeit ggf. durch eine redundante Energieversorgung unter Verwendung intelligenter Stromverteiler gewährleistet werden. Kritische Situationen wie beispielsweise die Überhitzung einzelner Leitungen müssen rechtzeitig erkannt und entschärft werden. Das Abschalten unkritischer Verbraucher oder auch die Freigabe alternativer Versorgungswege sind mögliche Maßnahmen, die eingeleitet werden können.

Um diesen Anforderungen gerecht werden zu können, sind neue Bordnetzarchitekturen zu untersuchen. Neben der Sicherstellung der funktionalen Anforderungen fließen in deren Bewertung auch die Konsequenzen für die Fertigung ein, speziell die Beherrschung der Variantenvielfalt des Bordnetzes, spielen die Auswirkungen auf Masse und Preis eine Rolle.

Ein Hilfsmittel in diesem Bewertungsprozess sind simu-

lationsgestützte Verfahren. Unter Verwendung der Verhaltensbeschreibungssprache Modelica und der Simulationsumgebung OpenModelica ist dafür eine Modellbibliothek erstellt worden [1].

1 Leitungsmodell

1.1 Modellierungsansatz

Das thermisch-elektrische Leitungsmodell ist aus Segmenten aufgebaut, deren Modellstruktur in Bild 1 dargestellt ist. Der elektrische Leitungswiderstand R ist temperaturabhängig. Der im j . Leitungssegment generierte Wärmestrom $P = R(T[j]) \cdot i^2$ wird in radialer Richtung über den durch den thermischen Widerstand $R_{thConSur}$ und die Wärmekapazität C_{thIns} beschriebenen Isolator der Leitung an die Umgebung und in axialer Richtung über Wärmeleitung an benachbarte Leitungssegmente oder Kontaktstellen abgeführt.

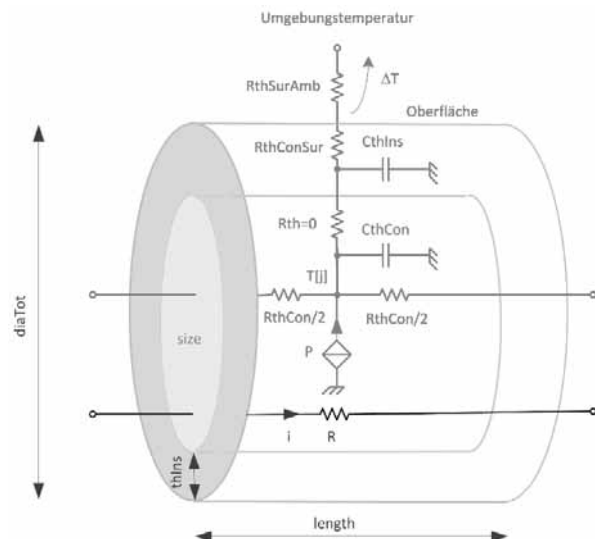


Bild 1: Modell für ein Leitungssegment

Die Parameter des Leitungsmodells werden mit Ausnahme des Konvektionswiderstandes $R_{thSurAmb}$ aus

Geometrie- und Materialparametern, diese ggf. aus dem Leitungstyp abgeleitet (siehe [1]). Es ist

$$R_{thSurAmb} = \frac{1}{\alpha \cdot dia_{Tot} \cdot \pi \cdot length} \quad (1)$$

Kritische Größe in dieser Beziehung ist der Wärmeübergangskoeffizient α . Er hängt stark vom Leitungsdurchmesser dia_{Tot} und der Temperaturdifferenz ΔT zwischen der Leitungsoberfläche und der Umgebungstemperatur ab (siehe Bilder 1 und 2). Eine Änderung von α beeinflusst stark die Wärmeabfuhr durch Konvektion und damit die Temperatur des Leiters. Je größer α desto besser kann Wärme durch Konvektion abgeführt werden und umso geringer ist in der Folge die Selbsterwärmung des Leitungsstücks. Somit führen zu große Werte des Wärmeübergangskoeffizienten u.U. zu einer zu optimistischen Temperaturabschätzung, d.h. die ermittelte maximale Temperatur des Leiters ist ggf. zu klein.

Im einfachsten Fall kann α vor Beginn der Simulation auf einen konstanten Wert gesetzt werden. Bei vorsichtiger und konservativer Betrachtung ist α eher zu klein als zu groß anzusetzen. Ist beispielsweise eine Temperaturdifferenz von 30K unkritisch und der Leitungsdurchmesser kleiner als 1 cm, so ist $\alpha = 10 \frac{Ws}{m^2K}$ mit Blick auf die mittleren Kurve in Bild 2 eine akzeptable Wahl für erste Simulationsexperimente.

Bei genauerer Betrachtung wird die Abhängigkeit des Wärmeübergangskoeffizienten vom Leitungsdurchmesser dia_{Tot} und der Temperaturdifferenz ΔT zwischen Leitungsoberfläche und Umgebungstemperatur bei freier laminarer Konvektion entsprechend [2] (Gleichung 2.19) während der Simulation berücksichtigt

$$\alpha(dia_{Tot}, \Delta T) = \left(\frac{0.1254}{\sqrt{dia_{Tot}}} + 1.0932 \cdot \sqrt[6]{\Delta T} \right)^2 \quad (2)$$

Im Modell werden Temperaturdifferenzen $\Delta T < 0$ wie der Fall $\Delta T = 0$ behandelt. Um unrealistisch große Werte des Wärmeübergangskoeffizienten zu vermeiden, wurde er bei der Umsetzung im Modell begrenzt.

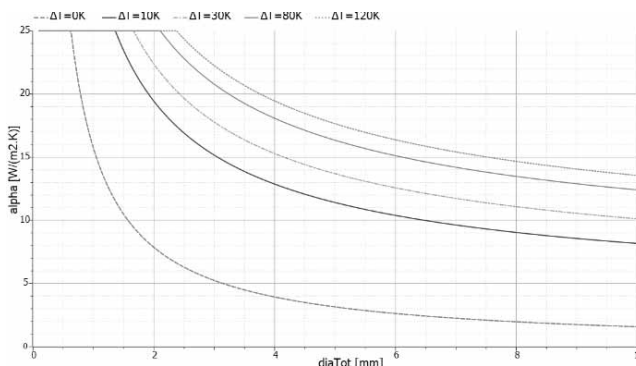


Bild 2: Wärmeübergangskoeffizient i.A. von Leitungsdurchmesser und Temperaturdifferenz ΔT bei freier Konvektion

Das Leitungsmodell wird aus einer Kette von Leitungssegmenten aufgebaut.

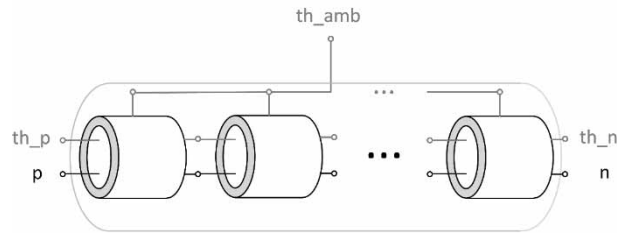


Bild 3: Aufbau des thermisch-elektrischen Leitungsmodells aus Leitungssegmenten

Die Modelica-Funktion `writeMassPrice2File` ermöglicht es, die Werte von Masse und Preis (soweit verfügbar) von Komponenten in eine csv-Datei zu schreiben. Dazu werden folgende Zeilen in das Modelica-Modell eingefügt:

```
// sum up of masses and prices
parameter String name = getInstanceName();
parameter Real PriceComp = ... ;
parameter Modelica.SIunits.Mass
    MassComp = ... "mass [kg]";

initial algorithm
writeMassPrice2File(MassComp, PriceComp, name);
```

Die implementierte Lösung ermöglicht es, beim Start eines Simulationslaufes die Werte für alle Leitungen in einer Datei zu erfassen.

1.2 Modellvarianten

Fall	Modell
1	WireGeneralExtendedThermal (useHeatPorts=true) Thermische Klemmen müssen beschaltet werden. Der Wärmeübergangskoeffizient wird entsprechend (2) während der Simulation angepasst oder auf einen konstanten, vorgegebenen Parameter gesetzt.
2	WireGeneralExtendedThermal (useHeatPorts=false) Die Temperatur an allen thermischen Klemmen wird auf eine vorgegebene Temperatur gesetzt. Ansonsten entspricht das Verhalten dem Fall 1.
3	WireThermal Geometrie- und Materialparameter werden aus dem Typ der Leitung ermittelt. Verhalten ansonsten wie in den Fällen 1 und 2 in Abhängigkeit von useHeatPorts.
4	WireGeneralExtended Nur ein Leitungssegment wird, Wärmefluss in axialer Richtung wird nicht berücksichtigt. Der Wärmeübergangskoeffizient muss vorgegeben werden.
5	Wire Vorgabe von Parametern über den Leitungstyp. Sonst wie Fall 4.

Auf Basis des skizzierten Ansatzes können die in der vorangegangenen Tabelle aufgeführten Modellvarianten abgeleitet werden. Sie unterscheiden sich im Aufwand für die Parametrisierung, in der Detaillierung der Modellbeschreibung und daraus abgeleitet dem erforderlichen Simulationsaufwand. Speziell der Übergang zu einer Beschreibung mit konstantem Wärmeübertragungskoeffizienten kann den Aufwand erheblich reduzieren. Ein mögliches Vorgehen besteht darin, mit einem Modell gemäß den in der Tabelle beschriebenen Fällen 1, 2 oder 3 zu beginnen, damit eine Abschätzung für den Wärmeübertragungskoeffizienten (siehe Bild 2) abzuleiten und für Variantensimulationen auf ein Modell entsprechend Fall 4 oder 5 zu verwenden.

1.3 Anwendungsmöglichkeiten

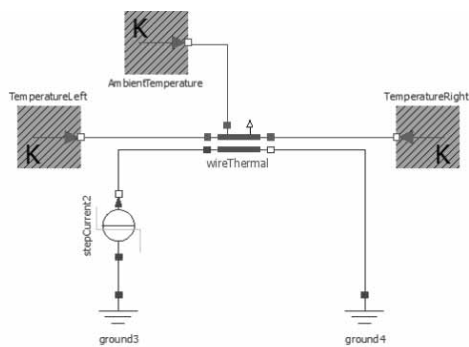


Bild 4: Simulationsmodell zur Erfassung des radialen und axialen Wärmeflusses

Bild 4 zeigt eine Simulationsanordnung mit einem Modell entsprechend Fall 1. Eingeschaltet wird ein Strom von 1,5 A, der durch eine 1 m lange Leitung mit einem Durchmesser des Leiters von 1 mm und einem Außendurchmesser von 2,3 mm fließt. Die Umgebungstemperatur und die Temperatur an den Leitungsenden werden auf 20 °C gesetzt.

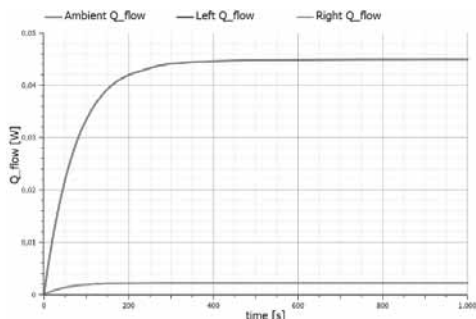


Bild 5: Wärmefluss in radialer und axialer Richtung

Das Modell ermöglicht eine Abschätzung des Wärmeflusses in axialer Richtung durch das linke und rechte Leitungsende und durch Konvektion an die Umgebung. Wie zu erwarten sind die Wärmeflüsse durch linkes und rechtes Leitungsende in diesem Fall gleich und sehr viel kleiner als die Wärmeableitung durch Konvektion.

Durch die diskrete Beschreibung einer Leitung mit einer großen Anzahl Leitungsstücken kann auch die örtliche Verteilung der Temperatur auf einer Leitung ermittelt werden. Entscheidend für die Beurteilung des thermischen Verhaltens ist die dabei erreichte maximale Temperatur. Bild 6 zeigt die Ergebnisse einer Beispielsimulation mit einem Modell entsprechend Fall 2 bei Vorgabe der Umgebungstemperatur von 20 °C.

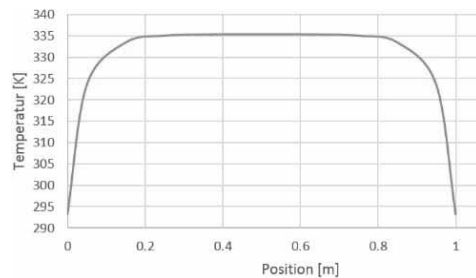


Bild 6: Örtliche Verteilung der Temperatur bei vorgegebener Temperatur an den Leitungsende

Auch die Leitungsenden sind auf dieser Temperatur fixiert worden. Durch die 1 m lange Leitung mit einem Außendurchmesser von 2,3 mm fließt ein Strom von 15 A.

2 Bibliothek OnBoardSystem

Es wurden neben den beschriebenen Varianten des Leitungsmodells Verhaltensmodelle für typische Komponenten im elektrischen Bordnetz von Kraftfahrzeugen erstellt und unter Verwendung der Modellierungssprache Modelica erstellt und mit der Version 1.9.6 der OpenModelica Software [3] des Open Source Modelica Consortium (OSMC) [4] erprobt.

Die Modelle sind in der Bibliothek OnBoardSystem zusammengefasst. Die Bibliothek ist wie aus Bild 7 ersichtlich gegliedert. Das Verhalten eines Teils der Modelle wird gleichungsbasiert beschrieben. Durch die Änderung typischer Parameter kann es modifiziert werden. Daneben können für ausgewählte Komponenten Leistungsprofile vorgegeben werden. Für die intelligenten Stromverteiler wurden unterschiedliche Varianten modelliert, mit denen ein Schließen oder eine Unterbrechung des Stromflusses nachgebildet werden kann. Öffnen oder Schließen

kann durch Steuersignale oder auf Grund der Überwachung oder der aus dem Stromfluss abgeleiteten Temperatur der angeschlossenen Leiter erfolgen.

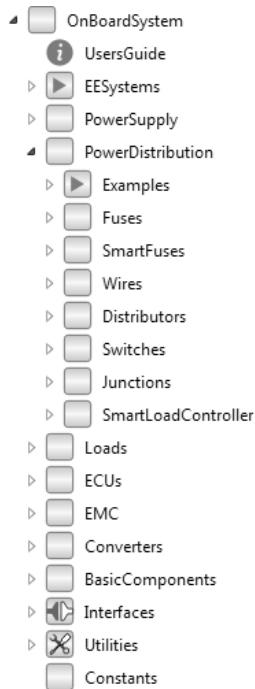


Bild 7: Modellbibliothek OnBoardSystem

3 Bordnetzuntersuchungen

Mit den Modellen für Leitungen, Lasten und intelligente Stromverteiler wurden ein konventionelles Bordnetz, ein konventionelles Bordnetz mit SmartFuses, eine Energy-Backbone-Topologie und eine Ringtopologie exemplarisch untersucht. SmartFuses werden dabei zum Trennen und Verbinden von Verbrauchern genutzt. SmartFuses werden zu Gruppen zusammengefasst, die als SmartLoadController oder intelligenter Stromverteiler bezeichnet werden. Bei den Bordnetzuntersuchungen wurden maximale und mittlere Endtemperatur der Leitungen und der tiefste Spannungseinbruch ermittelt. Neben den unterschiedlichen Ansätzen zur Energiesteuerung mit Schmelzsicherungen und intelligenten Stromverteilern können alternative Bordnetztopologien mit Unterschieden hinsichtlich Leitungsmasse, Leitungstemperatur und auch des Preises für die Leitungen erstellt werden. Einfache Beispiele für Bordnetze wie eine konventionelle Topologie, eine Energy-Backbone-Topologie und eine Ringtopologie mit intelligenten Stromverteilern sind im Package OnBoardSystem.EESystems zusammengestellt.

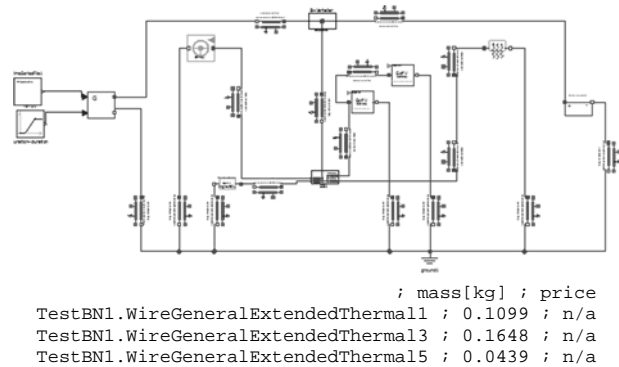


Bild 8: Bordnetz TestBN1 und Ausschnitt aus csv-Datei

Berücksichtigt werden können dabei Annahmen für die Umgebungstemperatur in den unterschiedlichen Bereichen für die Verlegung der Leitungen durch Vorgabe unterschiedlicher Umgebungstemperaturen. Einzelheiten können in Teil 1 von [1] entnommen werden.

Vergleichsrechnungen mit dem hier vorgestellten Leitungsmodell und dem in [5] vorgestellten haben bei gleichen Randbedingungen eine gute Übereinstimmung ergeben.

Danksagung

Der Forschungsvereinigung Automobiltechnik (FAT) im VDA und dem Arbeitskreis VDA/FAT AK30 danken die Autoren für die Unterstützung der Arbeiten.

References

- [1] *Simulationsgestützte Methodik zum Entwurf intelligenter Energiesteuerung in zukünftigen Kfz-Bordnetzen*. Berlin: FAT-Schriftenreihe 305; 2018. 210 p. Online: <https://www.vda.de/de/services/Publicationen/fat-schriftenreihe-305.html>
- [2] Audrius Ilgevicus: *Analytical numerical analysis and simulation of heat transfer in electrical conductor and fuses*. Hochschule der Bundeswehr München, Dissertation, 2004. Online: <http://athene-forschung.unibw.de/doc/85275/85275.pdf>
- [3] OpenModelica Release 1.9.6. Online: <https://build.openmodelica.org/omc/builds/windows/releases/1.9/6/>
- [4] Informationen zu OpenModelica und zum Open Source Modelica Consortium (OSMC). Online: <https://www.openmodelica.org/>
- [5] Ludwig Brabetz, Mohamed Ayeb, and Leonard Gysen: Model-Based Circuit Protection Using Solid State Switches. *SAE Int. J. Engines*. 2017; 10(3):1375-1382. doi: 10.4271/2017-01-1641

Model-based Analysis of Maintenance-induced Availability of Aircraft in an Airline Network

Marie Bieber^{1*,3}, Barbara Glock², Alexander Plagemann¹, Nikolas Popper³

¹Airbus Operations GmbH

**marietheresebieber@gmail.com*

²dwh GmbH simulation services, Neustiftgasse 57-59, 1070 Vienna, Austria

³Institute for Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria;

Abstract. Optimized aircraft maintenance concepts usually aim at increasing the aircraft availability and reducing costs. A model is presented that quantifies the effect of the increased aircraft availability when applying different maintenance concepts on an entire airline fleet. An agent-based approach is suggested: The aircraft with their maintenance tasks are dynamically assigned to the flight schedule. One major advantage of the presented methodology is that constraints for specific aircraft tails can be added easily. This is potentially of interest when testing new maintenance concept that are more focused on optimizing maintenance for single aircraft, by e.g. a stronger dependency on aircraft usage and findings.

Introduction

Currently a lot of studies are done on aircraft maintenance and its optimization to create more availability. In further consequence this leads to an economical and operational advantage for the airline operating the aircraft. Aircraft make money when they are flying. So, the baseline for airline planning is the sentence “The schedule is the king.” Once the airline has flights scheduled and tickets sold, the most important thing is to operate those flights. Usually airlines plan in the following way:

1. **Master Schedule design:**
An optimal set of flight legs is selected.
2. **Fleet Assignment:**
The aim of the fleet assignment is to determine which type of aircraft should fly each flight leg.
3. **Aircraft Scheduling:**
The schedule of each aircraft is determined includ-

ing all maintenance tasks and the commercial constraints.

4. **Tail assignment:**

According to the schedule and the aircraft schedules a tail number is assigned to flight legs. The master schedule is translated into an operational flight schedule.

5. **Crew Assignment:**

To each flight leg a crew is assigned according to the working constraints of the crews.

Once the schedule is planned it is operated and continuously updated. Operational interruptions might occur and lead to delays or cancellations. This might have as a consequence changes in the operational flight schedule.

1 Methods

Clearly, changing the number of revenue flights means that the master schedule has to be changed. However, changing the master schedule has a big impact on airline operations and depends on a lot more factors than considered in this paper, like the airline policy. The approach used is to not change the master schedule itself but instead work with the currently used master schedule containing n flight legs and fix the fleet size a_c as the smallest number of aircraft, such that all flight legs of the master schedule can be operated without any maintenance constraints. When taking into account maintenance the aircraft are able to operate $m \leq n$ flight legs of the master schedule. The number m is the model output and can be used as a measurement of the effect of introducing the maintenance concept in the airline fleet. In fact, the problem that has to be dealt with is the tail assignment problem.

1.1 Linear Programming approach

In the classical approach, which is broadly used in the existing literature, the first task faced in solving the tail assignment is to identify a set of feasible flight routes that can be operated by a single aircraft. Often the maintenance constraints are already considered in this step of the tail assignment. In [1] this is described as "maintenance routing". Maintenance is often done at night due to flight restrictions. Considering this aspect [2] propose algorithms to model the aircraft routing due to these maintenance requirements. Another method to solve the maintenance routing problem is presented by [3]. As an input a fixed flight schedule with aircraft assigned to it is given, which is then optimized by re-assigning the aircraft to flight routes (set of flight legs) in order to undergo different maintenance checks. A heuristic approach is provided to solve the problem of minimizing the maintenance cost and costs incurred during the re-assignment of aircraft to flight segments.

The most common mathematical model used to describe the tail assignment problem is an integer linear program. This was already presented by [4]. The practise today is to combine the mathematical models with constraint programming techniques. While mathematical programming is more focused on optimization rather than on quickly obtaining a solution, constraint programming obtains a feasible solution quicker. The first approach combining the mathematical and constraints programming techniques to solve the tail assignment problem was presented by [1]. It is a very convenient approach since achieves both accuracy and is quick enough for practical use. In fact, the work has been the basis for various further studies on the tail assignment problem and is used by several airlines.

The main objective of an airline is to operate the master schedule. Operational interruptions of any kind should be avoided. The problem is examined by [5] in his doctoral thesis where two mathematical models for the robust tail assignment are proposed. They are solved by an extension of the standard solution methods presented in [1]. Another approach introduced by [6] addresses operational uncertainty by extending the one-day routes aircraft maintenance implicitly for the subsequent two days. However, here also the flight routes are given as an input, but can be modified by an iterative algorithm.

1.2 Agent-based modelling

As opposed to the linear programming model, which is a top-down solution approach, an agent-based model provides a bottom-up approach. Rather than looking at the entire model system the focus lies upon each individual entity and its behaviour within the system. An agent-based system consists of one or more autonomous individuals, the agents. In [7] an 'agent' is characterized by autonomy, reactivity, pro-activeness and social ability, i.e. interaction with other agents. An agent-based model, as [8] states that 'An agent-based model is a simulation model that employs the idea of multiple agents situated and acting in a common environment as central modelling paradigm.'

The model presented in this paper is based on an agent-based approach mainly due to the adaptiveness of agent-based models. Maintenance tasks can be added to each aircraft, which allows the flexibility to define specific tasks depending on the aircraft usage or other factors. Another advantage of the agent-based approach is that instead of separating the airline planning process from the real aircraft operations, both are dynamically combined. Probabilities of delay, delay propagation and operational interruptions can be considered directly in the airline planning. This enables one to optimally plan airline operations in a robust and realistic way.

2 The Agent-based Model

The model consists of aircraft agents and the airline network represented through the master schedule that should be operated by the aircraft. The model returns the number of revenue flights operated in a specified period of time.

The main scope of the agent-based model is to show the effect of a maintenance induced increased availability of aircraft on airline operations. This is done by modelling the aircraft tails, their behaviour and movements and embedding them in the airline network. This can be seen in Figure 1, where the agents behaviour is described in a state chart. An aircraft agent can be assigned to a flight leg. After landing, if a maintenance task package is due, the aircraft undergoes maintenance. When the maintenance is finished, it is assigned a buffer time to be prepared for the next flight. If it does not undergo maintenance, the aircraft is directly assigned a buffer time when it landed. As

soon as the buffer time passed, it becomes available for flights again.

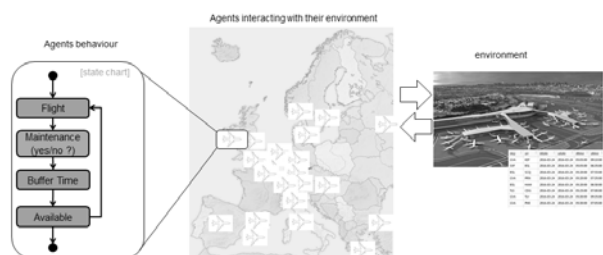


Figure 1: ABM generic architecture

One of the characteristics of the model is that the operational flight schedule is created dynamically during a simulation run. The aircraft follow specific rules defining their behaviour. Flight legs, turn around times and maintenance tasks are assigned to the aircraft tails in the moment they occur. Therefore, not only the flight routes are created dynamically but also the ground times between flights and the maintenance ground times.

3 Results

Based on in-service data and real airline data the model was tested for various maintenance scenarios. One of those scenarios aimed at analysing the effect of check overruns. Scheduled maintenance includes regulated maintenance tasks, that have to be performed regularly. The most common maintenance packaging practise is to perform checks in regular intervals, such as daily or weekly checks. A typical task packaging setup is illustrated in Table 1.

maintenance type	frequency (days)	duration (hours)
daily	1.5	2
weekly	8	3
A	190	8
1C	730	93
2C	1460	140
6Y	2190	350
12Y	4380	730

Table 1: Typical scheduled block type maintenance packages

The term *check overrun* refers to the phenomena that performing a maintenance task package took for an unexpected reason longer than it was scheduled. But does reducing these overruns have a considerable impact on airline operations? The scenario of a 1C check overrun is simulated for an example airline with a calibrated fleet size of 31 aircraft. Instead of the usual duration of 93 hours one assumes the check to take 98 hours (which equals an overrun of around 0.2 days). The overall results in terms of the number of revenue flights and the average ground times for the simulated period of time are given in Table 2.

	no overrun	overrun
Number of revenue flights	6 325	6 325
Average ground time (seconds)	12 100.29	12 095.13

Table 2: Results for the Scenario 1C Check overrun

In Figure 2 the number of revenue flights for every aircraft tail are listed where the tails that had to undergo a 1C checks are highlighted. Similarly, Figure 3 displays the average ground times for every aircraft tail.

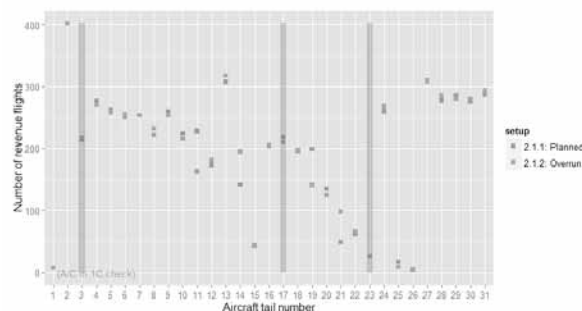


Figure 2: Number of revenue flights per aircraft tail for the 1C check overrun

The simulation reveals that increasing the duration for a 1C check overrun does not have a significant impact, neither on the total number of revenue flights operated during the time frame nor on the average ground times. For the aircraft tails that had to undergo the 1C check the results are more in alignment to what is expected: For two aircraft tails the number of revenue flights

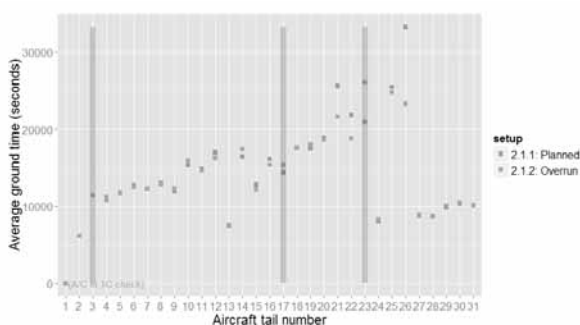


Figure 3: Average ground times per aircraft tail for the 1C check overrun

decreased in the overrun setup and the average ground time increased.

A reasonable explanation for this is that the model dynamics causes an aircraft that is delayed unexpectedly to be replaced by another aircraft dynamically. As a consequence, delays are compensated directly and with a simplicity that is not feasible in reality. The fleet size of the airline is big enough to compensate a downtime of a single aircraft tail efficiently.

4 Challenges and future developments

During simulations it was observed that some unexpected results are due to a poor fleet size calibration. As a next step, it would be suggested to work on that matter.

One major limitation of the model was observed to be how it handles an unexpected delay, in this case for maintenance tasks. Rather than reflecting the effect of the very complicated process of adopting to unexpected changes, as it would happen in real airline operations, the model assigns a different aircraft tail to the flight that should be operated and therefore dynamically solves the interruption caused in airline operations. To see the effects of e.g. an overrun as it happens in reality better, a possibility would be to first run the model and than use the thereby created tail assignment as a constraint for the second run, where the aircraft operations are simulated.

5 Conclusion

The model performs quite well when tuning the input parameters to be very close to what is used for real flight operations. This is also true for the maintenance setup. When changing the maintenance scenario significantly, the model adapts to the new setup. Then, the results for the number of operated revenue flights and average ground times are much closer to the classical maintenance scenario than what is observed for real flight operations.

All in all, the approach has the potential to be developed further for performing tail assignments that are adoptable to a wide range of maintenance scenarios and is adaptive to changes. Through using an agent-based approach a new methodology was devised that provides an alternative to the classical linear programming approach. In the agent-based approach constraints and attributes can be altered for single aircraft tails. One promising application for the model would be to analyze the effect of maintenance concepts that are more dependent on the usage in flight hours or flight cycles or individual findings or failures for single aircraft tails. The paper provides the framework for further investigations into this field.

References

- [1] Grönkvist M. *The tail assignment problem*. Chalmers tekniska högskola. 2005.
- [2] Gopalan R, Talluri KT. The Aircraft Maintenance Routing Problem. *Operations Research*. 1998; 46:260–271.
- [3] Sriram C, Haghani A. An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice*. 2003;37(1):29–48.
- [4] Levin A. Scheduling and fleet routing models for transportation systems. *Transportation Science*. 1971; 5(3):232–255.
- [5] Dovica I. Robust tail assignment. 2014;.
- [6] Maher SJ, Desaulniers G, Soumis F. The tail assignment problem with look-ahead maintenance constraints. 2015;.
- [7] Wooldridge M. *Agent-based software engineering*, vol. 144. 1997.
- [8] Siegfried R. *Modeling and simulation of complex systems: A framework for efficient agent-based modeling and simulation*. Springer. 2014.

Data Farming und simulationsbasierte Robustheitsanalyse für Fertigungssysteme

Thomas Schulze^{1*}, Niclas Feldkamp², Sören Bergmann², Steffen Straßburger²

¹ Institut für Technische und Betriebliche Informationssysteme, Otto-von-Guericke Universität Magdeburg, Universitätsplatz 2, 39102 Magdeburg, *thomas.schulze@ovgu.de

² Fachgebiet Wirtschaftsinformatik für Industriebetriebe, Technische Universität Ilmenau, PF 10 05 65, 98684 Ilmenau

Abstract. Diskrete Simulation ist eine etablierte Methodik zur Untersuchung des dynamischen Verhaltens von komplexen Fertigungs- und Logistiksystemen. Konventionelle Simulationsstudien fokussieren auf einzelne Modellaspekte und spezifische Analysefragen. Der Umfang der ausgeführten Szenarien ist häufig gering. Das Konzept des Data-Farming verwendet das Simulationsmodell als Datengenerator für eine breite Skala von Experimenten und ermöglicht unter Nutzung von Data-Mining-Methoden eine wesentlich breitere Untersuchung des simulierten Systems sowie eine höhere Komplexität in den abgeleiteten Erkenntnissen. Anforderungen an Simulationssysteme und -modelle zur Durchführung von Data-Farming werden erläutert. Eine Erweiterung des Ansatzes ist die simulationsbasierte Robustheitsanalyse auf der Basis von Verlustfunktionen nach Taguchi. Beide Vorgehensweisen werden an einer Fallstudie aus dem Fahrzeugbau demonstriert.

1 Einführung

Simulation ist ein etabliertes Werkzeug zur Planung und Steuerung von Fertigungs- und Logistiksystemen. Traditionelle Simulationsstudien versuchen ein vorgegebenes Projektziel zu erfüllen, wie z. B. Verbesserung des Hallenlayouts bzw. Reduzierung von Pufferkapazitäten. Zum Erreichen der Ziele werden häufig einzelne Simulationsexperimente manuell definiert oder es werden simulationsbasierte Optimierungsverfahren verwendet.

Mit der Verfügbarkeit von höherer Rechenleistung durch schnelle Prozessoren und paralleles Abarbeiten sowie der Anwendung von Data-Mining-Verfahren werden die Anwendungsmöglichkeiten von Simulationen erweitert. Somit können sehr viele ausgewählte Simulationsexperimente durchgeführt werden um im Folgenden, bspw. mittels Data-Mining-Methoden, unbekannte Wirkzusammenhänge aufzudecken. Der Simulator agiert hierbei als Datengenerator mit dem Ziel, eine umfangreiche Datenmenge zu erzeugen. In einem Folgeschritt werden diese Datenmengen unabhängig vom Simulator analysiert. Diese Analyse ermöglicht die Ableitung von Erkenntnissen zum simulierten System,

die sich aus einem klassischen Simulationsprojekt mit wenigen Experimenten nicht ableiten lassen. [1], [2].

Ein Produktionsprozess wird i.A. als robust bezeichnet, wenn er gegen unerwünschte Einflussgrößen stabil bleibt und die Produktionsziele unter Einhaltung wirtschaftlicher Zielgrößen erreicht. In diesem Beitrag wird unter robustem Fertigungssystem ein System verstanden, welches gegenüber Schwankungen im operatives Produktionsprogramm stabil bleibt.

Allgemein ist festzustellen, dass bei der Planung von Fertigungssystemen die Robustheit der Systeme eine steigende Bedeutung erfährt, häufig wird aber zur simulationsgestützten Planung nur ein vorgegebenes Produktionsprogramm verwendet. [3], [4]. An diesem antizipierten Produktionsprogramm wird die gesamte Planung ausgerichtet. Dagegen sind robuste Systeme in der Lage, auch bei sich verändernden Produktionsprogrammen, die geforderten Leistungsparameter zu sichern. Die Methode des Data Farmings [5] in Kombination mit der Anwendung von Verlustfunktionen nach Taguchi [6] erlaubt eine Bestimmung der Robustheit unterschiedlicher Systementwürfe.

Ziel dieses Beitrags ist das Aufzeigen von ausgewählten Ergebnissen aus einer Data-Farming-Analyse sowie aus der Ermittlung von robusten Systementwürfen an Hand einer Fallstudie aus einem Industrieprojekt. Darüber hinaus werden allgemeine notwendige Anforderungen an Simulationssysteme und -modelle zur Durchführung einer großen Anzahl von Simulationsexperimenten abgeleitet.

Der Beitrag ist wie folgt aufgebaut: Nach der Einleitung erfolgt ein kurzer Abriss zu den Themen Data Farming und simulationsbasierter Robustheitsanalyse. Anschließend werden Anforderungen an Simulationssysteme und -modelle zur Anwendung von Data Farming und Robustheitsanalyse diskutiert. In einer Fallstudie werden die erläuterten Methoden veranschaulicht. Fazit und Ausblick schließen den Beitrag ab.

2 Data Farming

Die Grundidee des ursprünglichen Data-Farming-Konzeptes [7],[8],[9] ist die Verwendung eines Simulationsmodells als Datengenerator, wobei das Ziel ist, mit Hilfe von effizientem Experimentdesign und High Performance Computing, ein möglichst vollständiges Spektrum von Ergebnisdaten zu erzeugen und somit den Informationsgewinn zu verbessern [5],[7]. Die „Farming“-Metapher drückt hierbei aus, dass ähnlich einem Farmer, der sein Land möglichst ertragreich kultiviert, der Datenertrag des Simulationsmodells maximiert werden soll [8]. Die hierbei eingesetzten Ansätze zur Gestaltung von Simulationsexperimenten, z. B. die häufig verwendete Methode Nearly Orthogonal Latin Hypercubes (NOLH) [10], erlauben bei vertretbaren Datenmengen respektive Experimentzahlen die umfassende Abbildung möglicher Wertekombinationen von Eingabeparametern [11]. Ursprünglich wurde Data Farming für militärische Gefechtssimulationen entwickelt, da bei der Anwendung von klassischen Simulationsstudien die gestellten komplexen Fragestellungen nicht ausreichend beantwortet werden konnten [7].

Aufbauend auf dem originären Data-Farming-Konzept wurden weitere Methoden zur Analyse der Daten entwickelt bzw. angewendet. Eine grundlegende Methodik, welche das Auffinden von versteckten, potenziell nützlichen Wirkzusammenhängen in Ergebnisdaten von nicht-militärischen Simulationsmodellen, insbesondere im Kontext von Produktion und Logistik, erlaubt, wurde von Feldkamp et al. unter dem Namen „Knowledge Discovery in Simulation Data“ entwickelt [12],[1]. Hierbei kommen zur Verarbeitung der vom Simulator generierten Daten Methoden des Data Mining sowie Methoden der interaktiven, visuellen Analyse zum Einsatz. Diese von Feldkamp [12] entwickelte Methodik strebt nach der konsequenten Verzahnung von Datenanalyse und -Visualisierung, wobei deren Verbindungsglied die menschliche Fähigkeit zur Interpretation und Schlussfolgerung darstellt. Gefördert wird dies durch ein hohes Maß an Interaktivität des Gesamtprozesses der Datenauswertung [13]. Diese Methodik zur Wissensentdeckung in Simulationsdaten wurde in Industrieallstudien erprobt.

3 Simulationsbasierte Robustheitsanalyse

Im Allgemeinen versteht man unter Robustheit eines Systems, in wie weit sich nicht direkt vom Systembetreiber kontrollierbare Störgrößen auf die interessierenden Leistungskennzahlen eines Systems auswirken. Hierbei ist ceteris paribus die Robustheit hoch, wenn die Auswirkungen der Störgrößen möglichst gering ist, vice versa ist diese gering wenn die Auswirkungen hoch sind [15]. Des Weiterem wird oft, wenn auch von der eigentlichen, ursprünglichen Definition abweichend, eine hohe Leistung der als robust gekennzeichneten Systeme unterstellt; d. h. im engeren Sinn sind auch Lösungen mit sehr geringer Leistung robust, wenn diese interessierenden Leistungskennzahlen unter verschiedenen Werten für die Störgrößen nicht oder nur schwach schwanken. Im Folgenden sollen aber nur Lösungen als robust bezeichnet werden, die neben dieser nötigen Bedingung auch ein definiertes Leistungskennzahlmaß, z. B. Durchsatz und Durchlaufzeiten ausreichend erfüllen. Ziel ist es, die durch den Planer kontrollierbaren Systemeigenschaften, auch als Systemkonfiguration bezeichnet, so zu setzen, dass die mittlere Systemleistung über alle zu berücksichtigen Störungen möglichst hoch ist und die Varianz der Systemleistung möglichst klein ist.

Zur Bewertung der Robustheit wird unter anderen die Methode der Verlust-Funktionen (Loss Functions) nach Taguchi [6] genutzt. Taguchis Idee resultiert aus seinen umfangreichen Erfahrungen aus dem Bereich des Qualitätsmanagements und hierbei insbesondere auf der Beobachtung, dass jede Abweichung von einem Zielwert einen monetären Verlust nach sich zieht. Zur Bewertung einer Systemkonfiguration sind verschiedene Verlustfunktionen abhängig der Rahmen- und Zielparаметer möglich, die in Tabelle 1 aufgezeigt sind.

Art der Verlustfunktion	Formel
Nominal-the-best	$\bar{L} = k[\sigma^2 + (\bar{y} - \tau)^2]$
Smaller-the-better	$\bar{L} = k[\bar{y}^2 + \sigma^2]$
Larger-the-better	$\bar{L} = k \left[\sum (1/y^2) \right] / n$

Tabelle 1: Verlustfunktion für verschieden Zwecke [16].

\bar{L} steht jeweils für den durchschnittlichen Verlust einer bestimmten Systemkonfiguration über alle Konfigurationen von Störgrößen, wobei \bar{y} und σ^2 den Mittelwert bzw. die Varianz der betrachteten Zielgröße einer Systemkonfiguration darstellen. Bei der „nominal-the-best“-Verlustfunktion, welche Abweichungen von einem vorgegebenen Zielwert sowohl nach oben als auch nach unten bestraft, wird dieser Zielwert als τ angegeben. Zusätzlich kann die sog. Qualitätsverlustkonstante k verwendet werden, um den Qualitätsverlust in geeigneter Weise umzurechnen und monetär bewerten zu können.

Taguchis Ansatz wurde in mehreren Publikationen auf Simulationsexperimente angewendet. So skizzierte Sanchez einen Ansatz zur Integration des Konzepts der Robustheit mit der Response-Surface-Metamodellierung zur Optimierung von Modellen [17]. Dellino et al. nutzte einen ähnlichen Ansatz zur simulationsbasierten Optimierung von Robustheitsproblemen unter Verwendung von Response-Surface-Methodik und Kriging-Metamodellen [18]. Auch im Bereich der Gefechtsfeldsimulation wurden die Potentiale der Verlustfunktion Taguchis erkannt [19].

In der hier vorgestellten Methodik wird das Ziel verfolgt, eine Robustheitsanalyse basierend auf den Verlustfunktionen von Taguchi mit einem großangelegten Experimentdesign und visuell unterstützten Methoden zur Wissensentdeckung für Fertigungssimulationen zu kombinieren [14]. Hierzu kommen gekreuzte Experimentpläne zum Einsatz.

Der Experimentplan zur Variation der Systemkonfigurationen wird gekreuzt mit einem Experimentplan zur Abbildung der Störszenarien. Dies erlaubt neben der Analyse der Konfigurationen je Störszenario auch Bias frei die Analyse der Robustheit einer Systemkonfiguration gegenüber den Störfaktoren mittels den beschriebenen Verlustfunktionen. Abschließend können mittels Methoden des Knowledge Discovery in Simulation Data Rückschlüsse bzgl. der relevanten Systemfaktoren und deren Ausprägungen gezogen werden.

4 Anforderungen an Simulationssysteme und -modelle zur Durchführung von Data-Farming

Die verwendeten Simulationssysteme und -modelle müssen bestimmten Anforderungen genügen, die im

Folgenden erläutert werden. Darüber hinaus wird ein Data-Farming-Management-Tool mit den folgenden Basisfunktionen benötigt:

- Erstellung des Experimentdesigns unter Verwendung geeigneter Methoden wie NOLH, Automatisches Verteilen, Starten und Beenden der Experimente sowie deren Monitoring und
- Erfassen und Aufbereiten der Resultate aus den einzelnen Experimenten. Fehlerhafte Experimente müssen erkannt und in Abhängigkeit von der Fehlerart gegebenenfalls automatisch neu gestartet werden.

Die zu verwendeten Simulationssysteme müssen die Modellgenerierung bzw. -modifikation und das Setzen von Eingabeparametern ohne Nutzereingriffe erlauben. Schnittstellen müssen vorhanden sein, um die Eingabedaten aus den definierten Experimentplänen zu importieren. Es sind Datenschnittstellen zu verwenden, die ein laufzeiteffizientes Einlesen sicherstellen. Des Weiteren müssen die Simulationssysteme durch das Data-Farming-Management-Tool steuerbar sein.

Existierende kommerzielle Simulationssysteme zur Simulation von Fertigungssystemen, wie Plant Simulation [20] oder FlexSim [21] verfügen nicht über alle benötigten Basisfunktionen eines Data-Farming-Management-Tools. Für das Data-Farming in der Fallstudie wurde daher ein proprietäres Softwarewerkzeug entwickelt.

Eine der Limitationen in der Anwendung der Data-Farming-Methodik ist durch die Validität der Simulationsmodelle begründet. Für klassische Simulationsstudien mit wenigen Szenarien sind entsprechende Validierungsverfahren entwickelt worden [22]. Im Ergebnis des Data-Farming-Experimentdesigns werden Wertekombinationen für Eingabedaten verwendet, die unter Umständen den validierten Eingabedatenraum aus der klassischen Studie überschreiten. Bei der Analyse der Ergebnisdaten ist dieser Umstand zu berücksichtigen.

Zur Durchführung von Data-Farming muss das existierende Simulationsmodell aus der klassischen Simulationsstudie angepasst werden. Zum einen muss der Umfang der interessierenden Zusammenhänge an das Spektrum der Ausgabewerte angepasst werden. Das bedeutet ggf. weitere Ausgabewerte zu modellieren oder auch nicht benötigte Werte aus dem Modell zu entfernen. Des Weiteren müssen Schnittstellen zur Kommunikation mit dem Data-Farming-Management-Tool zum Einlesen der Eingabedaten und der Ergebnisausgabe implementiert werden.

Eine Modellanalyse zur Erhöhung der Abarbeitungsgeschwindigkeit wird empfohlen. Nur die Modellvariablen, die als Ergebnisgrößen für das Data Farming verwendet werden, sind zu berücksichtigen. Wertaufzeichnungen, statistische Berechnungen, grafische Aufbereitungen und das Speichern auf externen Geräten von nicht benötigten Modellvariablen erhöhen in unnötiger Weise die Rechenzeit. Bei der Durchführung von Tausenden von Experimenten beim Data-Farming hat die Rechenzeit eines einzelnen Experimentes einen Einfluss auf die Anzahl der auszuführenden Experimente, wenn für die Durchführung der Untersuchungen nur ein definierter Zeitrahmen zur Verfügung steht.

5 Fallstudie

Für eine Demonstration der Vorgehensweisen zum Data-Farming und der simulationsbasierten Robustheitsanalyse wird ein Simulationsmodell aus einem Industrieprojekt verwendet, welches die Montage von Fahrzeugen nachbildet. Das „legacy“-Simulationsmodell wurde vom Industriepartner erstellt und in einer klassischen Simulationsstudie verwendet. Dieses Modell wurde an die Anforderungen zum Data-Farmings angepasst. Die ursprüngliche Rechenzeit für ein Experiment konnte dabei signifikant gesenkt werden.

Das zu simulierende System umfasst über 50 Montagestationen mit über 50 Werkern und 14 Puffer. Die Stationen werden in 5 verschiedenen Zonen logisch zusammengefasst. Das Produktspektrum enthält 720 verschiedene Fahrzeugtypen, die an den einzelnen Stationen unterschiedliche Montagezeiten aufweisen. Die Ergebnisgrößen für die klassische Simulationsstudie sind u. a. der mittlere tägliche Durchsatz und die Auslastung der Werker.

Im ersten Schritt des Data-Farming-Konzeptes werden die Eingabefaktoren definiert. Diese Faktoren werden klassifiziert in Entscheidungs- und Störfaktoren. Entscheidungsfaktoren lassen sich bei der Planung des zu untersuchenden Systems durch den Planer beeinflussen. Störfaktoren hingegen sind durch den Planer nicht steuerbar. Tabelle 2 gibt einen Überblick zu den verwendeten Eingangsfaktoren und ihren Wertebereichen. Die Eingabefaktoren mit den entsprechenden Wertebereichen wurden gemeinsam mit dem Industriepartner ausgewählt.

Zwischen 13 ausgewählten Montagestationen der

Hauptlinie sind Puffer vorgesehen. Mit der Faktorengruppe „Pufferkapazität“ wird die Kapazität eines einzelnen Puffers bezeichnet, wobei insgesamt 13 Faktoren dieser Gruppe zugeordnet sind.

Faktorname	Anzahl	Typ	Wertebereich
Pufferkapazität	13	E	1 - 10
Skill Level Werker	5	E	1 oder 2
Normzeitkoeffizient	5	E	1,0 – 1,3
Produktmix	30	S	30 Varianten

Tabelle 2: Eingabefaktoren ; Erläuterung der verwendeten Faktoren (Typ E: Entscheidungs-; Typ S: Störfaktor).

Mit dem Faktorengruppe „Skill Level Werker“ wird die Fähigkeit der Werker beschrieben, an mehr als an einer Station arbeiten zu können. In dieser Studie wurden zwei diskrete Ausprägungen 1 oder 2 verwendet. Der Wert 1 bedeutet hierbei, dass ein Werker nur an einer ihm explizit zugewiesenen Station arbeiten kann. Ein Wert von 2 erlaubt eine höhere Flexibilität, indem Werker die Fähigkeiten besitzt, sowohl an seiner direkt zugewiesenen Station, sowie an den vor- bzw. nachgelagerten Stationen zu arbeiten.

Die Skill-Level-Faktoren werden hierbei nicht für einzelne Werker, sondern insgesamt je Zone vergeben. Für die fünf Zonen ergeben sich somit fünf verschiedene Faktoren aus dieser Gruppe. Die einzelnen Faktoren dieser Gruppe werden mit Skill_Zone* bezeichnet, wobei für * die Werte 1 bis 5 gelten.

Die Faktorengruppe „Normzeitkoeffizient“ (Real Time Calibration Coefficient – RTCC) beschreibt, in welchem Verhältnis die geplante zur tatsächlichen Montagezeit steht. Ein Wert von 1 bedeutet hierbei das die tatsächliche Bearbeitungszeit der geplanten entspricht. Ein höherer Wert zeigt an, dass die tatsächlichen Montagezeiten kürzer als die geplanten Montagezeiten sind. Der Faktor „Normzeitkoeffizient“ kann ebenfalls nur je Zone differenziert werden. Auch hier ergeben sich somit für die fünf Zonen die entsprechenden einzelnen Faktoren, die mit RTCC_Zone* bezeichnet werden.

Der Faktor „Produktmix“ ist ein qualitativer Faktor, d. h., die numerischen Werte für diesen Faktor dienen nur der Kennzeichnung von verschiedenen Produktspektren, auch als Produktmixe bezeichnet. Unter einem Produktmix wird eine Liste prozentualer Anteile

aller zu produzierenden Produkttypen aufgefasst, die in einem bestimmten Zeitraum zu produzieren sind. Das verwendete Simulationsmodell verwendet einen derartigen Produktmix als Eingabeparameter und generiert aus selbigen ein Produktionsprogramm, welches die Reihenfolge der zu fertigenden Fahrzeuge nachbildet. Es werden 30 verschiedene Produktmixe verwendet und der Wert für den Faktor „Produktmix“ kennzeichnet die laufende Nummer des Produktmixes.

Die Fallstudie wurde in zwei Phasen durchgeführt. Phase 1, als Data-Farming-Analyse bezeichnet, verwendet nur die Entscheidungsfaktoren. In Ergänzung dazu werden in Phase 2 auch die Störfaktoren mit einbezogen. Diese Phase wird als Robustheitsanalyse bezeichnet.

5.1 Data-Farming-Analyse

In Phase 1 der Fallstudie wurden zunächst nur die 23 (13 + 5 + 5) verschiedenen Entscheidungsfaktoren verwendet. Die Einbeziehung der Störfaktoren, d.h. unterschiedlicher Produktmixe, erfolgt in der zweiten Phase.

Der Experimentplan für diese Entscheidungsfaktoren wurde auf Basis der Nearly-Orthogonal-Latin-Hypercube-Methode (NOLH) erstellt [10]. Basierend auf dieser Methode wurden 510 verschiedene Kombinationen über die verschiedenen Entscheidungsfaktoren erzeugt. Eine einzelne Kombination wird auch als Konfiguration bezeichnet. Nach der Durchführung der 510 Experimente können erste Auswertungen vorgenommen werden. Unter der Festlegung der Simulationszeit mit 5 Wochen betrug die Rechenzeit für jeden einzelnen Experiment 260 bis 280 sec.

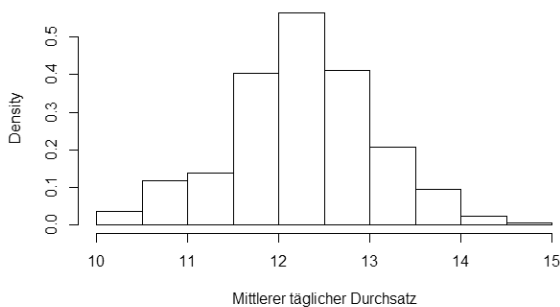


Bild 1: Histogramm der Ergebnisgröße „Mittlerer täglicher Durchsatz“ über 510 Systemkonfigurationen.

Eine erste Analyse untersucht den Einfluss der Entscheidungsfaktoren auf die Ergebnisgrößen. Für diesen Beitrag wird sich auf die Ergebnisgröße „Mittlerer täglicher

Durchsatz“ beschränkt. Bild 1 zeigt ein Histogramm über die 510 Simulationsexperimente für die Ergebnisgröße „Mittlerer täglicher Durchsatz“, wobei jedes Experiment eine Systemkonfiguration beschreibt. Der Wertebereich erstreckt sich von minimal 10,35 bis maximal 14,6 Fahrzeuge pro Tag. Diese Ergebnisgröße wird von den Konfigurationen und somit von den Entscheidungsfaktoren beeinflusst.

Weitere Analysen untersuchen, welche der 23 Entscheidungsfaktoren den stärksten bzw. einen starken Einfluss auf die Ergebnisgröße „Mittlerer täglicher Durchsatz“ haben. Erste Hinweise gibt eine Korrelationsanalyse zwischen der Ergebnisgröße und den Entscheidungsfaktoren. Hohe Korrelationskoeffizienten weisen auf einen tendenziell starken Einfluss hin. Die Faktoren der Gruppe „Pufferkapazität“ weisen allesamt sehr niedrige Korrelationswerte auf. Damit wird ein sehr geringer Einfluss dieser Faktorengruppe auf den Durchsatz vermutet. Die fünf Entscheidungsfaktoren mit den höchsten Korrelationskoeffizienten werden als Hauptentscheidungsfaktoren bezeichnet und sind in Tabelle 3 aufgelistet. Es ist abzulesen, dass die Faktorengruppe „Pufferkapazität“ keinen signifikanten Einfluss auf die Ergebnisgröße hat.

Faktorname	Korrelationskoeffizient
RTCC_Zone1	0,532
Skill_Zone1	0,507
RTCC_Zone4	0,409
RTCC_Zone3	0,165
RTCC_Zone2	0,098

Tabelle 3: Korrelationskoeffizienten für die Hauptentscheidungsfaktoren.

Ohne weitere Betrachtung von Interaktionseffekten zwischen einzelnen Faktoren scheint es, dass der Entscheidungsfaktor RTCC_Zone1, d.h. der Normzeitkoeffizient für Zone 1, den größten Einfluss auf die Ergebnisgröße „Mittlerer täglicher Durchsatz“ hat. Der Wert des Korrelationskoeffizienten von 0,532 lässt vermuten, dass noch weitere Entscheidungsgrößen bzw. Interaktionseffekte den täglichen Durchsatz beeinflussen. Das Streudiagramm zwischen diesem Entscheidungsfaktor und der Ergebnisgröße in Bild 2 indiziert die Tendenz, dass ein hoher Wert für RTCC_Zone1 zu einem hohen täglichen Ausstoß führt.

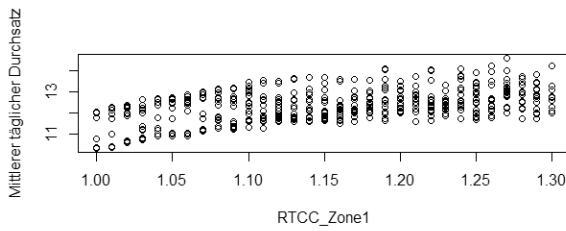


Bild 2: Scatterdiagramm für den Entscheidungsfaktor RTCC_Zone1 und der Ergebnisgröße „Mittlerer täglicher Durchsatz“.

Aus der Liste der Korrelationskoeffizienten in Tabelle 3 lässt sich weiter ableiten, dass die Entscheidungsfaktoren RTCC_Zone1 und Skill_Zone1 den höchsten Einfluss auf die Ergebnisgröße „Mittlerer täglicher Durchsatz“ haben. Für den Planer ergibt sich die Erkenntnis, dass zur Erreichung eines hohen Durchsatzes die Zone1 von besonderer Bedeutung ist und ggf. sogar einen Engpass darstellt.

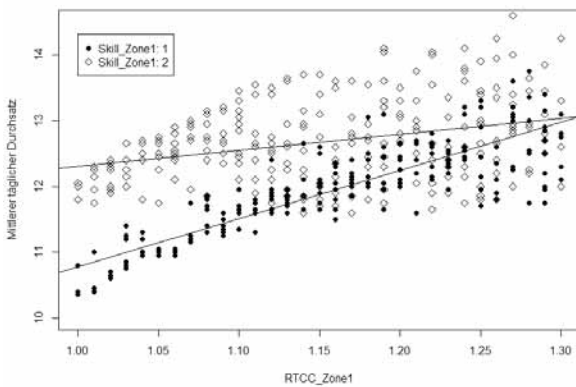


Bild 3: Scatterdiagramm für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ in Abhängigkeit von den Entscheidungsfaktor RTCC_Zone1 und Skill_Zone1.

In einer weiteren Analyse wird untersucht, ob es eine Interaktion zwischen beiden Entscheidungsfaktoren auf diese Ergebnisgröße gibt. Dabei ist zu berücksichtigen, dass der Entscheidungsfaktor Skill_Zone 1 nur die Werte 1 oder 2 hat. Bild 3 zeigt ein Streudiagramm zwischen der Ergebnisgröße „Mittlerer täglicher Durchsatz“ und den beiden Einflussfaktoren. Die Linien sind Regressionsgeraden für die Ergebnisgröße in Abhängigkeit vom Wert des Entscheidungsfaktors Skill_Zone1. Aus diesem Bild wird geschlossen, dass bei niedrigen Werten von bis zu bis 1,15 für den Einflussfaktor RTCC_Zone1 der Durchsatz gesteigert wird, wenn der Entscheidungsfaktor Skill_Level1 den Wert

gleich 2 hat. Des Weiteren kann für die Planer abgeleitet werden, dass eine höhere Flexibilität der Werker in Zone 1 vorteilhaft ist, wobei sich dieser Effekt bei hohen Werten für den Normzeitkoeffizienten wieder relativiert.

Entscheidungsbäume sind eine weitere Methodik zur Analyse von Einflussfaktoren und ggf. sogar Interaktionen zwischen diesen. Das Anwendungsziel eines Entscheidungsbaumes, hier eines Regressionsbaumes ist die Vorhersage eines Zielgrößenwertes in Abhängigkeit von Eingabewerten. Für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ wurde ein Regressionsbaum in Abhängigkeit von den identifizierten Hauptentscheidungsfaktoren erstellt (vgl. Bild 4). Die Prozentzahlen an den Knoten geben an, wieviel Prozent der untersuchten Fälle durch diesen Knoten beschrieben werden.

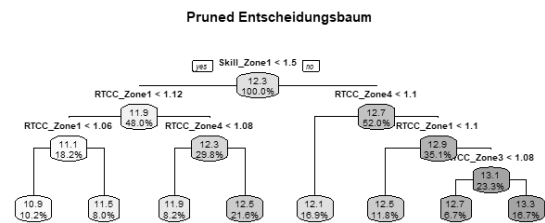


Bild 4: Pruned Entscheidungsbaum für die Ergebnisgröße „Mittlerer tägliche Durchsatz“.

Der Entscheidungsfaktor Skill_Zone1 bildet die Wurzel dieses Baumes, d.h. dieser Faktor hat den größten Einfluss im Zusammenspiel mit den anderen Entscheidungsfaktoren. Ein Wert von 1 für diesen Faktor bedeutet, dass dann der Mittelwert für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ gleich 11,9 über alle Kombinationen der restlichen Einflussfaktoren ist. Wird hingegen für den Einflussfaktor ein Wert von 2 verwendet, so ist der Mittelwert über die restlichen Wertekombinationen gleich 12,7. Aus diesem Baum lässt sich an die Planer des Montagesystems folgende Empfehlung für einen hohen Durchsatz ableiten:

$$(Skill_Zone1 = 2) \ \&\& \ (RTCC_Zone4 > 1,4) \ \&\& \ (RTCC_Zone1 > 1,1)$$

5.2 Robustheitsanalyse

Die Robustheit des Montagesystems steht im Zentrum von Phase 2 der Fallstudie. Bei der Planung von Fertigungssystemen spielt die Robustheit der Systeme gegenüber Störeinflüssen, wie z. B. Schwankungen der Produkteanteile im geplanten Produktionsprogramm, eine große Rolle. Mit der Robustheitsanalyse soll unter-

sucht werden, wie robust die 510 verschiedenen Systemkonfigurationen gegenüber solchen unterschiedlichen Produktmischen sind. Bei diesem Simulationsmodell kann sich auf Produktmische als Eingabeparameter beschränkt werden, da die Reihenfolge der Produkte durch das Simulationsmodell erzeugt wird.

Eine Robustheitsanalyse kann sich auf eine oder mehrere Ergebnisgrößen beziehen. Für diese Fallstudie wird sich auf die Analyse bezüglich der Ergebnisgröße „Mittlerer täglicher Durchsatz“ beschränkt. Ziel dieser Robustheitsanalyse ist es, die Systemkonfigurationen zu identifizieren, welche eine hohe Robustheit hinsichtlich dieser Ergebnisgröße aufweisen. Zur Durchführung der Analyse müssen die notwendigen Produktmische erzeugt werden. Diese Produktmische sind als Werte für den Störfaktor Produktmix (Tabelle 1) zu betrachten.

Der im klassischen Simulationsprojekt verwendete Produktmix wird als originärer Produktmix bezeichnet. Dieser Mix ist der Ausgangspunkt zur Generierung der benötigten ähnlichen Produktmische, die Derivate genannt werden. Der originäre Produktmix enthält 720 verschiedene Produkttypen. Die Anteile der Produkttypen am originären Produktmix sind unterschiedlich. Auf der Basis der folgenden Schrittfolge werden die Derivate abgeleitet.

1. Festlegung eines relativen Bereiches, z. B. 20 %, für alle Anteilswerte der Produkttypen im originären Produktmix.
2. Ermittlung der neuen Anteilswerte der Produkttypen aus den in Schritt 1 bestimmten Bereichen mittels stochastischer Experimente. Dabei muss die Bedingung eingehalten werden, dass die Summe der prozentualen Anteile 100 ergibt.

Schritt 2 muss für jeden neuen Produktmix wiederholt werden. In dieser Fallstudie wurden 29 zusätzliche Produktmische generiert.

Die Anzahl der zu verwendenden Produktmische wird in dieser Fallstudie stark durch die benötigte Rechenzeit für die Simulationsexperimente bestimmt, denn beim Experimententwurf für die Robustheitsanalyse werden die 510 Systemkonfigurationen mit den unterschiedlichen Produktmischen gekreuzt. Bei insgesamt 30 Produktmischen ergibt sich ein Experimentumfang von $30 \times 510 = 15300$ Experimenten.

Die Experimentbedingungen in Abschnitt 5.1 für die Analysen in Phase 1, mit 5 Wochen Simulationszeit, führten zu einer mittleren Rechenzeit von 5 min pro

Experiment. Unter Beibehaltung dieser Bedingung ergibt sich zur Durchführung der 15300 Experimente ein Rechenzeitbedarf von schätzungsweise 76500 min, das entspricht 1275 Stunden oder 53 Tagen. Der Rechenzeitaufwand für die Robustheitsanalyse ist unter diesen Bedingungen zu hoch.

Zur Reduzierung des Rechenzeitaufwandes wurden zwei Techniken eingesetzt:

1. Reduzierung der Rechenzeit für ein Experiment auf 1 min durch Verkürzung der Simulationszeit.
2. Parallelisierung der Experimente durch die Anwendung des proprietären Data-Farming-Management-Tools.

Durch die Verkürzung der Simulationszeit auf eine Woche konnte die Rechenzeit für ein Experiment auf 1 min verkürzt werden. Daraus ergibt sich für die Ergebnisgrößen der Simulation eine stärkere Beeinflussung durch den Initialisierungs-Bias. Als Konsequenz daraus sind beispielsweise die ermittelten Werte für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ kleiner als die Werte aus der Analyse mit einer Simulationsdauer von 5 Wochen. Da das Ziel der Robustheitsanalyse in einem Vergleich der Konfigurationen besteht, wurde diese Einschränkung akzeptiert, um alle 15300 geplanten Experimente durchführen zu können.

Durch die Anwendung eines proprietären Data-Farming-Management-Tools konnte die Experimentdurchführung automatisiert und parallelisiert werden. Die Parallelisierung kann wahlweise auf den Prozessorkernen eines PCs oder auf einer virtuellen Maschine mit mehreren Prozessoren oder Kernen erfolgen. Bei der Durchführung der Experimente auf 8 Kernen einer leistungsstarken virtuellen Maschine wurde ein Durchsatz von 360 Experimenten pro Stunde erreicht. Die theoretische Dauer beträgt annähernd 42 Stunden. Dieser Wert wurde bei der praktischen Durchführung der Experimente überschritten, da eine gewisse Anzahl von Experimenten wiederholt werden musste. Die reale Dauer für die 15300 Experimente betrug annähernd 60 Stunden.

In einem weiteren Schritt der Robustheitsanalyse werden für jede Konfiguration über alle Produktmische die Werte für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ zusammengestellt. Für diese 30 Werte je Konfiguration werden der Mittelwert und die Standardabweichung berechnet. Das Ziel der Robustheitsanalyse für die obige Ergebnisgröße besteht darin, die Konfigurationen zu identifizieren, die einen hohen Mittelwert und eine geringe Standardabweichung aufwei-

sen. Bild 5 zeigt ein Scatterdiagramm für die Mittelwerte und Standardabweichungen über alle 510 Konfigurationen. Der Bereich von Interesse ist eingekreist.

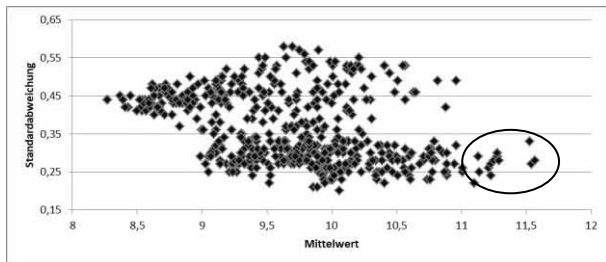


Bild 5: Scatterdiagramm für Mittelwerte und Standardabweichungen der Ergebnisgröße „Mittlerer täglicher Durchsatz“ über alle 510 Konfigurationen.

Zur Quantifizierung der Robustheit wird auf die Anwendung von Verlustfunktionen nach Taguchi [6] zurückgegriffen. Für jede Konfiguration wird ein Verlust ermittelt. Die Berechnung des Verlustwertes für eine Konfiguration erfolgt nach Formel 1

$$L = \sum_{i=1}^n (\tau - y_i)^2 \quad (1)$$

mit L als Wert der Verlustfunktion, τ als Schwellwert, y_i als Wert für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ für den Produktmix i und n der Anzahl der Produktmixe.

Die angewandte Analyse beinhaltet die folgenden Schritte:

1. Definition des Schwellwertes τ für die Ergebnisgröße „Mittlerer täglicher Durchsatz“. Unter Einbeziehung der errechneten Mittelwerte wird ein Schwellwert τ von 15 verwendet. Dieser Wert ist größer als das Maximum der ermittelten Werte.
2. Berechnung des Verlustes für jede Konfiguration.
3. Aufsteigende Sortierung der Liste aus Schritt 2.

Tabelle 4 zeigt die auf Basis der Verlustfunktion errechneten Ergebniswerte der Robustheitsanalyse für die sieben Konfigurationen mit den geringsten Verlusten. Es ist zu erkennen, dass die Werte für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ nicht wesentlich differieren. Ein Systemplaner hat nun die Möglichkeit, diese ausgewählten Konfigurationen auf ihre praktikable Umsetzung hin zu untersuchen.

Konfiguration	Mittl. Durchsatz	Verlust
448	11,57	354,48
373	11,54	361,60
103	11,53	363,60
410	11,29	414,48
237	11,18	417,52
256	11,28	417,84
169	11,25	423,44

Tabelle 4: Liste mit robusten Konfigurationen, dem mittleren täglichen Durchsatz und dem ermittelten Verlust.

Anschließend wurde in einer Korrelationsanalyse untersucht, welche Eingangsgrößen die Robustheit stark beeinflussen. Die Eingangsgrößen mit dem größten Einfluss wurden selektiert. Die ermittelten Werte der signifikanten Eingangsgrößen für robuste Konfigurationen sind in Tabelle 5 aufgelistet.

Die Ergebnisse der Robustheitsanalyse unterstützen den Systemplaner bei der Auswahl geeigneter Konfigurationen. Für robuste Konfigurationen werden die folgenden Werte empfohlen, die Tabelle 5 zusammengestellt sind.

Faktorname	Wertebereich
RTCC_Zone1	1,19 – 1,30
Skill_Zone1	2,0
RTCC_Zone4	1,23 – 1,30
RTCC_Zone3	1,10 – 1,24
RTCC_Zone2	1,16 – 1,28

Tabelle 5: Empfehlungen für die Entscheidungsfaktoren zum Erreichen robuster Konfigurationen.

6 Fazit und Ausblick

Dieser Beitrag präsentiert das methodische Vorgehen zum Data Farming in Verbindung mit simulationsbasierter Robustheitsanalyse und erläutert dieses Vorgehen an einer realen Fallstudie für die Simulation einer Montagelinie. Hierzu werden Anforderungen an die zu verwendenden Simulationssysteme dargestellt, damit diese Systeme im Kontext des Data-Farming operieren können. Des Weiteren werden Hinweise gegeben, wie

existierende Simulationsmodelle aus klassischen Simulationsprojekten für Data-Farming adaptiert werden sollten. In der Fallstudie werden die Ergebnisse ausgewählter Analysemethoden aus dem Data-Farming und Data-Mining für das Simulationsmodell vorgestellt. Für die Robustheitsanalyse hinsichtlich einer Ergebnisgröße wurde beispielhaft eine Verlustfunktion erstellt, die zur Bestimmung robuster Systemkonfigurationen verwendet wurde.

Entwicklungsbedarf besteht sowohl in der Automatisierung des Ablaufs zur Wissensentdeckung durch Data-Farming als auch in der erweiterten Nutzung interaktiver Werkzeuge zur visuellen Analyse. Forschungsbedarfe ergeben sich auch bzgl. des Experimentdesigns von Produktmischen bzw. Produktionsprogrammen unter Beachtung von Qualitätskriterien, wie der Orthogonalität, Ausgeglichenheit sowie der Anzahl der Experimente.

References

- [1] Feldkamp N; Bergmann S; Straßburger S; Schulze T. Data Farming im Kontext von Produktion und Logistik. In Wenzel S, Peter T, editors. *Simulation in Produktion und Logistik 2017. Simulation in Produktion und Logistik*; 2017. 20.-22. September 2017; Kassel:Kassel:Kassel university press. 169-178.
- [2] Feldkamp N; Bergmann S; Strassburger S; Schulze T. Knowledge Discovery in Simulation Data: A Case Study of a Gold Mining Facility. In Roeder TM, Frazier P I; Szechtmann R, Zhou E, Huschka T; Chick S E, editors. *Winter Simulation Conference 2016. Winter Simulation Conference*; 2016 December, Washington D.C.. Piscataway,N.J.: IEEE Inc.1607-1618.
- [3] Kleijnen, J P C; Sanchez S M, Lucas Thomas W, Cioppa T M. State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. *INFORMS Journal on Computing*. 2005; 17 (3): 263–289.
- [4] Painter M K, Erraguntla M, Hogg G L, Beachkofski B. Using Simulation, Data Mining, and Knowledge Discovery Techniques for Optimized Aircraft Engine Fleet Management. In: Perrone L F, Wieland F P, Liu J, Lawson B G, Nicol D M, Fujimoto R M, editors. *Winter Simulation Conference*; 2006 December; Monterey. Piscataway, N.J.: IEEE inc. 1250-1257.
- [5] Sanchez S M. Work Smarter Not Harder: Guidelines for Design Simulation Experiments. In Henderson S G, Biller B, Hsieh M H, Shortle J, Tew J D, Barton R R, editors. *Winter Simulation Conference 2007. Winter Simulation Conference*; 2007 December, Washington D.C.. Piscataway,N.J.: IEEE Inc. 805-816.
- [6] Taguchi, G. Quality engineering (Taguchi methods) for the development of electronic circuit technology. *IEEE Transactions on Reliability* 44, 1995. 225–229.
- [7] Horne G E; Meyer T E. Data Farming: Discovering Surprise. In Kuhl M E, Steiger N M, Armstrong F B, Joines J A, editors. *Winter Simulation Conference 2005. Winter Simulation Conference*; 2005 December; Orlando FL. Piscataway,N.J.: IEEE Inc. 1082–1087.
- [8] Sanchez S M. Simulation Experiments: Better Data, Not Just Big Data. In: Tolk A, Diallo S D, Ryzhov I O, Yilmaz L, Buckley S, Miller J A , editors. *Winter Simulation Conference 2014. Winter Simulation Conference*; 2014 December; Savannah GA. Piscataway,N.J.: IEEE Inc. 805–816.
- [9] Brandstein, A G, Horne G E. Data Farming: A Meta-technique for Research in the 21st Century. In Hoffman F G, Horne G E, editors. *Maneuver Warfare Science. Maneuver Warfare Science*; 1998 Dec; Quantico VA. Marine Corps Combat Development Command: 93–99.
- [10] Hernandez A S, Lucas T W, Carlyle M. Constructing nearly orthogonal latin hypercubes for any nonsaturated run-variable combination. *ACM Transactions on Modeling and Computer Simulation*. 2012 22, 1–17.
- [11] Kleijnen, J P, Sanchez S M, Lucas, T W, Cioppa T M. State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. *INFORMS Journal on Computing* 2005; 17: 263–289.
- [12] Feldkamp N, Bergmann S, Strassburger S. Knowledge Discovery in Manufacturing Simulations. In Taylor S, editors. *Proceedings of the 2015 ACM SIGSIM PADS Conference. 2015 ACM SIGSIM PADS Conference*; 2015 June; London UK. New York: ACM New York. 3–12.
- [13] Keim D A, Mansmann F, Schneidewind J, Thomas J, Ziegler H. Visual Analytics: Scope and Challenges. In: Simoff S, Boehlen M H, Mazeika A, editors. *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. Heidelberg: Springer; 2008. p 76-90.
- [14] Feldkamp N, Bergmann S, Strassburger S, Schulze T. Knowledge Discovery and Robustness Analysis in Manufacturing Simulations. In Chan V, D'Ambrogio A, Zacharewicz G, Mustafee N, editors. *Winter Simulation Conference. Winter Simulation Conference*; 2017 December; Las Vegas. Piscataway, N.J.: IEEE , p 3952-3963.
- [15] Park G J, Lee T H, Lee K H, Hwang K H. Robust Design: An Overview. *AIAA Journal*.2006; (44):181–191.
- [16] Phadke M S. *Quality engineering using robust design*. Englewood Cliffs, NJ: Prentice Hall, 1989.
- [17] Sanchez S. Robust design: seeking the best of all possible worlds. In Joines R R, Barton K, Kang K, Fishwick P A, editors. *Winter Simulation Conference. Winter Simulation Conference*; 2000 December; Orlando. Piscataway, NJ: IEEE Inc. 69–76.
- [18] Dellino G, Kleijnen J P C, Meloni J P C. Robust Simula-

- tion-Optimization using Metamodels. In Rossetti M D, Hill R R, Johansson B, Dunkin A, Ingalls R G, editors. Winter Simulation Conference. *Winter Simulation Conference*; 2009 Dec; Austin. Piscataway, NJ: IEEE Inc, 540–550.
- [19] Horne G, Åkesson B, Meyer S, Anderson S. Data farming in support of NATO. Final Report of Task Group MSG-088, Neuilly-sur-Seine Cedex: North Atlantic Treaty Organisation, 2014.
- [20] Plant Simulation: Siemens PLM Software. Available: <https://www.plm.automation.siemens.com/de/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml>. Accessed: 17-July-2018.
- [21] FlexSim. Available: <https://www.flexsim.com/de/>. Accessed: 17-July-2018.
- [22] Rabe M, Spieckermann S, Wenzel S. Verifikation und Validierung für die Simulation in Produktion und Logistik. Heidelberg: Springer Verlag, 2008. 237.

Simulation in Wood Science: a Novel Model for the Process Gluing

Carina Rößler^{1*}, Felix Breiteneker², Martin Riegler¹

¹ Wood K plus, Competence Centre for Wood Composites and Wood Chemistry, Area Wood Materials Technologies, Konrad-Lorenz-Straße 24, 3430 Tulln, Austria; *c.roessler@kplus-wood.at

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

Abstract. In this paper, a mathematical model for the sub-process of gluing in the manufacturing process of particleboards is presented. The model describes the application of resin droplets onto wood particles on laboratory scale. At the moment the model and simulations are carried out for the problem considered in two dimensions. The developed mathematical model uses lattice gas cellular automata for the movement of the wood particles, and random walk for the movement of the resin droplets. Feasibility was shown with simulation results of a greatly simplified mathematical model in two dimensions. This model was developed further for a better fitting to the real process. First simulation results of this more detailed model are presented. The next step will be the specification of how validation will be performed. In future, it is planned to create a mathematical model in three dimensions.

Introduction

The industrial manufacturing process of wood-based composites consists of several consecutive sub-processes. The quality of the final product depends on the raw material and the different processing steps. To ensure a high quality of the product, while optimizing the use of energy and resources, adjustable process parameters need to be continuously adapted [1]. Simulations of potential measures can predict the impact of these measures and hence avoid possible costs of rejects. This generates a deeper understanding of the manufacturing process and enables a more efficient usage of resources.

Based on a value benefit analysis, the sub-process gluing was selected for modelling and simulation.

The wood particles are located in a mixer and thus are moved. Due to the position in the mixer the particles have a local boundary condition. Their movement is locally limited (chips are located at the bottom of the mixer), unless they are transported upwards from the mixer. The resin is sprayed from above into the mixer using nozzles. The resin drops hit the moving wood particles. The aim is to describe the movement and

interactions of the wood and resin particles, which results in computing the resin distribution. Appropriate geometries for resin particles and wood particles needed to be selected and different size distributions to be applied. The rules for the different kinds of collisions needed to be defined. In the case of a collision between wood and resin, a statistical distribution was used for calculating the quantity of resin, which penetrates, adheres and passes to another wood particle (lubrication effect).

1 Gluing

In this section, the process of gluing is described according to [2], [3], [4] and [5]. Gluing is a sub-process of the manufacturing process of particleboards. In this process step the wood particles are glued with the resin formulation, which is sprayed into the mixer using nozzles. Based on [2] the typical dimensions of the wood chips are: length 8-10 mm, width 1.5-3.5 mm and thickness 0.25-0.4 mm. In [3] it is stated that the diameter of the resin droplets is between 30 μm and 100 μm . One goal of gluing is to equally distribute the resin droplets on the surface of the wood particles, which is important for the bonding of the wood particles. This has a strong impact on the properties of the final particleboard. In the following, the process of gluing is considered on laboratory scale by using a so-called ploughshare mixer, which is shown in Figure 1.

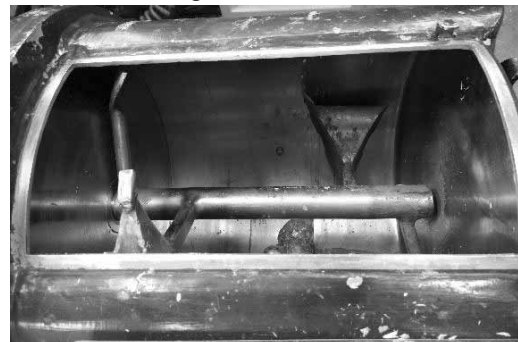


Figure 1: Picture of a ploughshare mixer.

Within the mixer used on laboratory scale, there are four ploughshares (also called mixing arms), a milling head, and an opening for the nozzle. First, the wood chips are located at the bottom of the mixer. Due to the mixing arms the wood particles are moved within the mixer, which constitutes a local boundary condition. The resin is sprayed into the mixer using a nozzle. The resin droplets hit the moving wood particles and thus the wood particles are glued.

2 Model

At the moment the model is developed for the two dimensional problem. For modelling the movement of the wood chips a lattice gas cellular automaton (LGCA) is used. The common LGCA is described in [6]. For using LGCA in this setting, the method has to be extended and modified, which will be explained later. For the LGCA a lattice, states and the evolution ε , where C is the collision operator and S is the streaming operator,

$$\varepsilon = S \circ C \quad (1)$$

have to be defined. The movement of the resin droplets is realised by a random walk across the lattice used in the LGCA.

The elements, which are used within this model, and the properties of the environment are described in the following.

- wood particles: The shape of the wood particles is considered to be a rectangle.
- resin particles: The shape of the resin droplets is considered to be a circle.
- resinating mixer: The geometry of the cross-sectional area of the resinating mixer is assumed to be a circle.
- ploughshares/mixing arms: The ploughshares move along a circle line.
- time interval
- grid width

In the following, the main elements of the LGCA are considered in more detail. In [7], which is a previous publication of the model, these are already described.

2.1 Lattice

The lattice is built by nodes, which are the vertices of equilateral triangles. Therefore, the lattice shows hex-

agonal symmetry. For obtaining “smooth” objects (circle, which builds the resinating mixer; rectangles, which build the wood particles; circles, which build the resin droplets), a suitable grid width has to be chosen.

In Figure 2 above, the black points are the vertices of the equilateral triangles, which are the nodes of the lattice. The red dashed line represents the wall of the resinating mixer. In Figure 2 below, a small part of the cross-sectional area of the mixer is depicted. The black lines are the edges of the equilateral triangles and the smaller colored triangles represent the directions of movement.

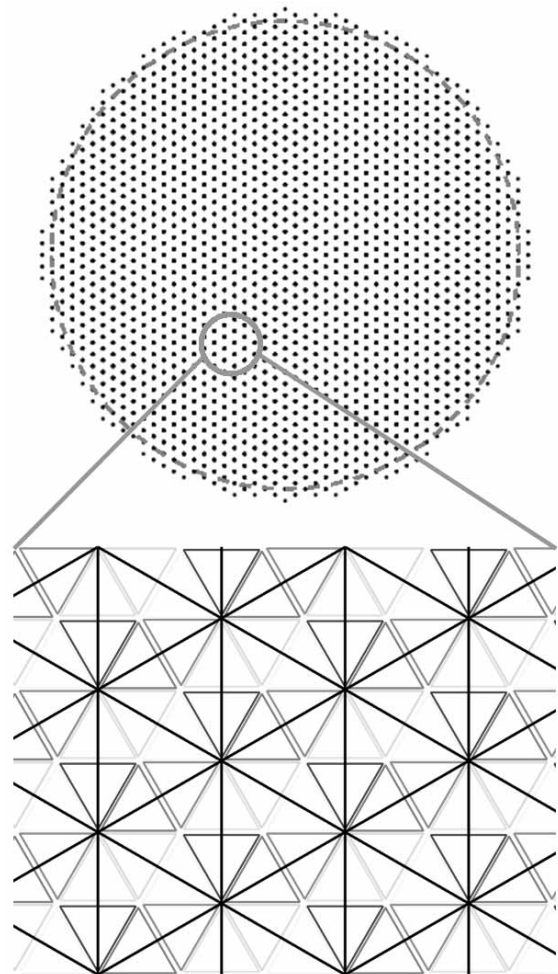


Figure 2: Schematic depiction of the lattice points and the lattice velocities.

According to [6] at each node there are six cells, which represent the six directions of movement. These correspond to the side lengths of the equilateral triangles and are the so-called lattice velocities. These common lattice

velocities are given in equation (2) for $i = 1, \dots, 6$.

$$c_i = \left(\cos\left(\frac{\pi}{3}i + \frac{\pi}{2}\right), \sin\left(\frac{\pi}{3}i + \frac{\pi}{2}\right) \right) \quad (2)$$

On the one hand, wood particles are stationary, if they are not moved by a mixing arm. On the other hand, resin particles can be stationary, if they stick to the wall of the resonating mixer. Due to the fact that particles can have velocity zero, the common LGCA has to be extended regarding the possible lattice velocities. Therefore, at each node a seventh cell is added, whereby its lattice velocity is zero. Thus, the lattice velocity of this cell is defined by equation (3).

$$c_0 = (0,0) \quad (3)$$

Due to the different scale of the wood and resin particle, and the corresponding grid width, the wood particles extend over several nodes of the lattice, which is not provided within the common LGCA. Therefore, some specifications regarding the states, collisions and streaming are necessary.

2.2 State

Each cell of a node must have a so-called state. The state indicates what kind of particle or part of the resinating mixer is located at a certain position. There are certain states, which are described in the following.

- empty: The state “empty” is used for the air, which will not be modelled, and for the definition of the state of a node, which will be explained later.
- wall: The state “wall” is used for the rigid body of the ploughshare mixer.
- ploughshare/mixing arm: For the moving mixing arms this state is used.
- milling head: The state used for the milling head.
- wood: The state for parts of a wood particle without resin.
- resin: The state for the resin droplets.
- glued wood: The state for parts of a wood particle, where resin is adhered, i.e. the state wood is changed to glued wood, if a wood particle and a resin particle collide with each other. After such a collision the resin droplet penetrates into the wood particle over time. Furthermore, if a glued wood particle collides with an unglued wood particle, part of the resin droplet passes to the unglued wood particle, which is called

lubrication effect. In section 2.3 these behaviors will be included in the collision rules.

Corresponding to the dimensions of the wood particle and the grid width, the wood particle expands over several nodes. For each node, which is occupied by the wood particle, the state of the cell, which corresponds to the direction of movement, is set to wood. Due to the fact, that each node, which is occupied by the same wood particle, has to move in the same direction. For the other cells of the nodes, the states are set to empty. According to the described situation the cells, which are set empty, of a node, which belongs to a wood particle, are also occupied by the wood particle. Thus, a modification of the common LGCA is necessary. Therefore, the state of a node is introduced. It is defined as wood, if one cell at a node has state wood or glued wood. In Figure 3 an example for this approach is shown schematically. The wood particle occupies three nodes of the lattice, which is indicated by the brown hexagons. The underlying lattice is depicted by the black lines. The brown circles represent the cells with state wood. These cells are chosen corresponding to the direction of movement, which is a certain lattice velocity, of the wood particle. The states of the other cells have to be empty due to the occupation by the wood particle.

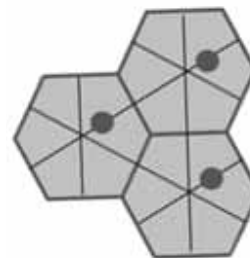


Figure 3: Exemplary illustration of a wood particle, which occupies three nodes.

2.3 Collision

As described in section 2.2 several kinds of states are used. Therefore, the rules for all possible types of collisions have to be defined. In the following, these are illustrated using collisions, where two particles or a particle and a machining part are involved. Of course, it is possible that collision of more than two particles can occur.

First, the collision rule for resin particles is defined. For this collision a perfectly inelastic collision is considered, thus after collision there is one bigger resin particle. In

Figure 4 the collision of two resin particles is depicted schematically.

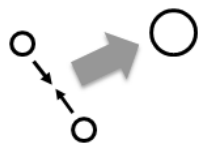


Figure 4: Schematic illustration of the collision of two resin particles.

Next, the collision rule for wood particles is defined. An elastic collision is used for this case, which is shown in Figure 5.

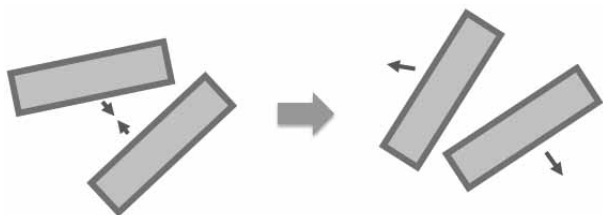


Figure 5: Schematic illustration of the collision of two wood particles.

Due to the porous structure of wood, it can absorb liquids. This is an important fact, which has to be included in the collision rule for wood and resin particles. If such a collision takes place, the resin particle sticks onto the wood particle. Furthermore, a part of the resin particle is absorbed by the wood particle, which is called penetration. In Figure 6 the collision of a wood and resin particle and the penetration are depicted schematically.

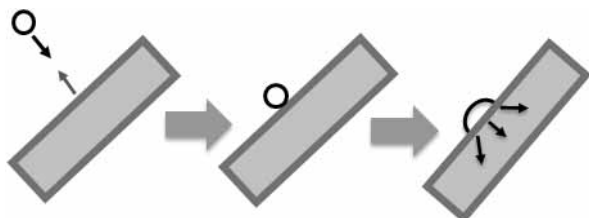


Figure 6: Schematic illustration of the collision of a wood particle and a resin particle.

Next, the collision of a glued wood and an unglued wood particle is considered. This type of collision is modelled as an elastic collision. Furthermore, in this case a part of the resin particle on the surface of the glued wood particle passes to the other wood particle, which is called lubrication effect. In Figure 7 this behaviour is shown schematically. For the collision of two glued wood particles a similar behaviour occurs, i.e. part of the resin drops can pass from one wood particle the other one.

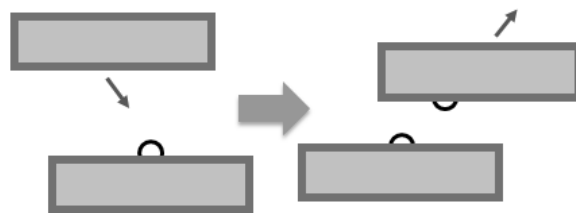


Figure 7: Schematic illustration of the collision of a glued wood and an unglued wood particle.

For all types of collisions mentioned above the velocity of the particles after collision is chosen randomly based on the lattice velocities. This assumption is used for simplification of the implementation.

In the following the collision of a wood particle with parts of the resinating mixer are considered. If a wood particle collides with the wall of the resinating mixer or a mixing arm, an elastic collision takes place. In Figure 8 this type of collision is shown by means of the wall.

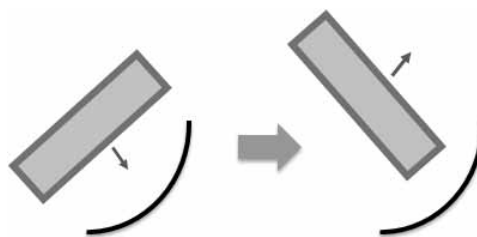


Figure 8: Schematic illustration of the collision of a wood particle and the wall of the resinating mixer.

In contrast to the collision described above, an inelastic collision is used as model for a collision of a resin particle with parts of the mixer. In case of a collision with the wall, the resin particle sticks onto the wall. Therefore, the velocity of the resin particle after collision is zero. In Figure 9 this type of collision is shown. If a resin drop collides with a mixing arm, the velocity of the resin droplet after collision corresponds to the velocity of the mixing arm.



Figure 9: Schematic illustration of the collision of a resin particle with the wall of the resinating mixer.

2.4 Streaming

For the streaming of the wood and resin particles, dif-

ferent approaches are used.

Wood particles. If the ploughshares do not move or do not touch a certain wood particle, the wood particle is stationary. Therefore, in the streaming step the wood particle does not move.

If the velocity is not equal to zero, the cells with state wood move according to the corresponding lattice velocity. Due to the technique described in section 2.2 it is guaranteed, that each node of a specific wood particle moves across the same lattice velocity.

Resin particles. The streaming of the resin particles is modelled by a random walk. Due to gravitational acceleration the resin droplets cannot move upwards. Therefore, not all lattice velocities are allowed for the movement of the particles. A random variable is used for choosing one of the possible lattice velocities. After determination of a permitted lattice velocity, the resin particle moves according to this choice across the lattice.

For a resin particle, which sticks onto the wall, the velocity is zero. If it adheres to a mixing arm, the velocity is set to the velocity of the mixing arm.

3 Simulation

For simplification of the implementation for the two dimensional model, some assumptions were made:

- The milling head and mixing arms are not realized.
- All wood particles have the same dimensions.
- All wood particles, which are not stationary, have the same speed.
- All resin droplets have the same size and velocity until they adhere on a wood particle or a part of the resonating mixer.
- The velocity after collision of the wood particles is determined randomly.
- The penetration of resin into the wood particle is not implemented.

As a further remark, it has to be mentioned, that not all parameters needed for the simulation can be measured by the real system. Therefore, not all parameters correspond to the real system, which has to be included in the interpretation of the simulation results.

The model was implemented using MATLAB R2015b.

In Figure 10 a snapshot of a simulation run is shown, whereas the wood particles are shown in brown and the resin droplets in blue.

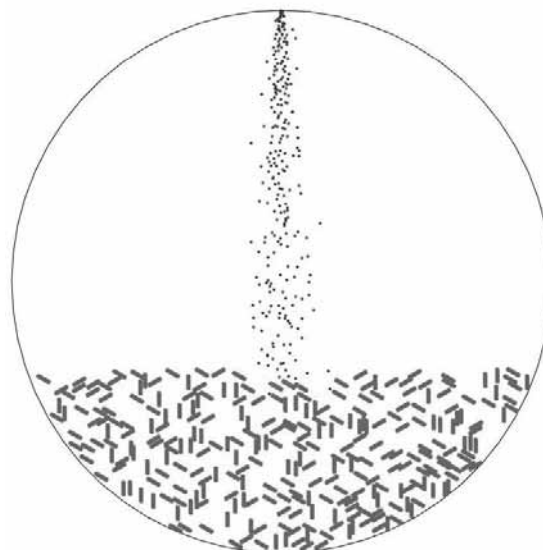


Figure 10: Movement of wood particles (brown) and resin droplets (blue) during simulation.

As illustrated in Figure 10, the resin droplets are moving downwards from the top. Areas of wood particles, which are glued, i.e. a collision of a resin droplet or a glued wood particle and an unglued wood particle took place, are shown in red. For obtaining quantifiable results, the relative frequency of glued wood chips in relation to the total number of wood chips is calculated. First results of this computation are shown in [7].

4 Plans for Validation

It is planned to validate the results using experiments on laboratory scale, since validation is important for the connection between the model and the reality. Regarding the behaviour of wood particles and resin droplets after collision, contact angle measurements will be performed for determining the time dependent behaviour (height, spreading) of the droplet on the wooden surface. In Figure 11 a schematic illustration of the technique for measuring the contact angle θ is shown.

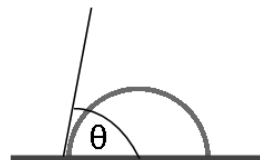


Figure 11: Schematic depiction of contact angle measurement.

Furthermore, for the penetration images of a tiny part of a wood particle, which is glued, can be analysed under a microscope. The lubrication effect will be validated by gluing a wooden board and check the weight before and after it got in contact with an unglued wooden board.

The final result of the sub-process gluing can hardly be measured by a defined criterion, therefore the validation of the final simulation results is limited.

5 Conclusion and Outlook

First simulations showed feasibility of the developed mathematical model. The next steps include improving the mathematical model and simulations in two dimensions. Furthermore, the validation of the simulation results will be carried out. Future plans include creating a mathematical model for the process in three dimensions.

Acknowledgements. The authors gratefully acknowledge the support from the Austrian Research Promotion Agency for funding as well as the Technische Universität Wien and the University of Natural Resources and Life Sciences for providing infrastructure.

References

- [1] Riegler M, Spangl B, Weigl M, Wimmer R, Müller U. Simulation of a real-time process adaptation in the manufacture of high-density fibreboards using multivariate regression analysis and feedforward control. *Wood Science and Technology*. 2013; 47, 1243-1259.
- [2] Wagenführ A, Scholz F. *Taschenbuch der Holztechnik*. Carl Hanser Verlag GmbH Co KG, 2012.
- [3] Rowell RM. *Handbook of wood chemistry and wood composites*. CRC press, 2005.
- [4] Deppe H, Ernst K. *Taschenbuch der Spanplattentechnik*. 4th. Leinfelden-Echterdingen, DRW. 2000.
- [5] Dunky M, Niemz P. *Holzwerkstoffe und Leime: Technologie und Einflussfaktoren*. Springer-Verlag. 2002.
- [6] Wolf-Gladrow DA. *Lattice-gas cellular automata and lattice Boltzmann models: an introduction*. Springer. 2004.
- [7] Rößler C, Riegler M, Breitenecker F. Modeling and simulation of moving wood chips and resin droplets within a resinating mixer using lattice gas cellular automata. *IFAC-PapersOnLine*, 2018; 51(2): 55-60.

Dynamische Modelle und Geographische Informationssysteme:

Bestandsaufnahme und Anforderungen an ein integriertes Modellieren in Raum und Zeit

Jochen Wittmann¹

¹ Hochschule für Technik und Wirtschaft Berlin, Studiengang Umweltinformatik, Wilhelminenhofstraße 75A, 12459 Berlin, Germany, wittmann@htw-berlin.de

Abstract. Mit der zunehmenden Verbreitung von Daten mit geographischem Raumbezug werden auch die entsprechenden dynamischen Modelle zur Spezifikation und zur Simulation von dynamischen Prozessen im geographischen Raum zunehmend nachgefragt. Häufig werden derartige Modelle in interdisziplinären Teams entwickelt, deren Mitglieder nicht unbedingt über vertiefte Kenntnisse in Simulationsmethoden verfügen. Eine Unterstützung durch geeignete Softwaresysteme ist daher besonders wünschenswert. Sowohl aus dem Bereich der Simulations-Software als auch aus dem Bereich der Geographischen Informationssysteme (GIS) finden sich Angebote. In dieser Situation untersucht der vorliegende Artikel zunächst die bestehenden Ansätze zur Spezifikation und zur Simulation von Prozessen in Raum und Zeit sowohl auf Seiten der Simulationstechnik als auch auf Seiten der GIS. Anschließend werden die grundsätzlichen Möglichkeiten einer Integration von Simulationsmethodik und der Behandlung von Dynamik in Raum und Zeit in einem geographischen Raum sowohl auf methodologischer als auch auf Softwareebene analysiert. Eine allgemeingültige Empfehlung für ein bestimmtes SW-System gelingt jedoch nicht. Vielmehr stehen am Ende des Artikels drei Anforderungskomplexe, deren Umsetzung eine problembezogene Behandlung von raumbezogenen Simulationsaufgaben vereinfachen und auch für Personen des Anwendungsfeldes handhabbar machen könnten.

Motivation

Das Potenzial einer Verbindung der Funktionalitäten von Simulationssystemen und Geographischen Informationssystemen erscheint groß, gerade, wenn man den Trend beobachtet, dynamische Prozesse mit Raumbezug zu analysieren, zu modellieren und zu simulieren: Routenplanung und -optimierung, Stadtentwicklung, Verbreitungs- und Ausbreitungsmodelle in der Biologie, Habitatmodellierung, usw.

In dieser Situation nimmt der Autor eigene frühere Arbeiten [10] wieder auf und untersucht mit dem vorliegenden Artikel zweierlei: Einerseits analysiert er, inwieweit entweder raumbezogene Dynamik in Simulationssysteme zu integrieren ist, oder aber GIS-Software den Aspekt der dynamischen Prozess-Modellierung abzubilden erlaubt.

Andererseits versucht der Artikel eine Reihe prototypischer Anforderungen im Bereich der raumbezogenen Modellierung zu erarbeiten und damit losgelöst von der Debatte um die am besten geeignete Software eine problembezogene Diskussion der aktuellen Problemfelder in diesem Bereich zu ermöglichen.

1 Traditionelle Modellierungsansätze für Prozesse in Raum und Zeit

Am Anfang steht zunächst ein Überblick über traditionelle Modellierungsansätze mit dem Fokus darauf, wie es in diesen möglich ist, raumbezogene Phänomene abzubilden.

1.1 kontinuierlich

In diesem Bereich werden gewöhnliche und partielle Differentialgleichungen behandelt. Bei den gewöhnlichen Differentialgleichungsmodellen ist der Raumbezug eingeschränkt. Zwei Alternativen sind üblich:

Im ersten Fall repräsentiert der Wert einer Zustandsgröße einen Bestand an einem festen, aber in den Gleichungen nicht näher bezeichneten Ort (z.B. die Bevölkerungszahl für Deutschland). Entweder der Ort wird

als punktförmig angenommen und konzentriert den gesamten Bestand dort auf sich, oder aber der Bestand wird als homogen über die Fläche bzw. den Raum der abgebildeten Entität verteilt angenommen.

Im zweiten Fall beschreibt die Differentialgleichung die Bewegung einer (als punktförmig angenommenen) Entität, wobei der umgebende Raum, in dem diese Bewegung stattfindet, beliebige Dimension aufweisen kann. In beiden Fällen muss das Raummodell vom Modellierer spezifiziert werden, das Modellierungsparadigma bietet dabei weder Grenzen noch Unterstützung.

1.2 diskret

Eine gänzliche andere Sicht auf dynamische Systeme bieten die diskreten Modellierungsparadigmen. Plötzliche Zustandsänderungen, die ohne jeden Zeitverzug zu einem Zeitpunkt zu interpretieren sind und durch eine Auslösebedingung aktiviert werden, stellen den einzigen Mechanismus zur Beschreibung von Systemdynamik dar. Mit einem derartigen Mechanismus lassen sich einerseits Zustandsänderungen für Entitäten mit einer festen Position im wie auch immer spezifizierten Raum abbilden, aber auch „ruckartige“ Bewegungen, wenn die Zustandsänderung, die im Ereignisrumpf beschrieben ist, eine Positionsänderung der Entität abbildet. In beiden Fällen gibt das diskrete Modellierungsparadigma die Spezifikation des Raumes nicht vor, sie steht vollständig im Ermessen des Modellierers. Auch hier bietet der Modellierungsansatz zwar vollständige Freiheit für das verwendete Raummodell, andererseits aber auch keinerlei Unterstützung bei dessen Aufbau und Verwaltung.

1.3 Spezialfall Zellularer Automat

Ganz anders sieht die Situation bei einem Sonderfall der diskreten Modelle aus, den Zellularen Automaten. Hier gibt das Modellierungskonzept ein homogenes Raster (in der Regel quadratisch) zur Beschreibung des Raumes vor, was dem Konzept der Rasterdaten im GIS entspricht. In jedem Rasterelement wird ein Zustand angenommen, der sich getaktet und für sämtliche Rasterzellen nach identischen Regeln in Abhängigkeit von den Zuständen der Nachbarzellen ereignishaft im Sinne der diskreten Simulation ändert.

Wegen seiner Einfachheit und seiner direkten Übertragbarkeit auf eine in der Fläche ablaufende Dynamik hat dieser Ansatz eine weite Verbreitung und die ursprünglich sehr strenge Definition des Zellularen Auto-

maten wird durch zahllose Erweiterungen und Anpassungen an die Gegebenheiten des jeweiligen Untersuchungsgegenstandes angepasst (Waben statt quadratische Zellen, mehrdimensionaler Zustandsvektor für jede Zelle, diverse Nachbarschaftsvarianten, usw.). Grundlage ist jedoch immer eine Kachelung der Ebene mit in Form und Verhalten identischen Zellen. Erweiterung auf den 3-dimensionalen Raum mit identischen, würfelförmigen Raumelementen sind denkbar, haben allerdings wohl wegen ihrer Ähnlichkeit zu den Finite-Element-Ansätzen wenig Verbreitung gefunden.

1.4 Objektorientierte und individuenbasierte Ansätze

Anders als bei den vorausgehenden Ansätzen, bietet das objektorientierte oder auch individuenbasierte Modellierungsparadigma (für eine präzise Definition und Unterscheidung sei aus Platzgründen hier nur auf Ortman [6] verwiesen) sämtliche Ausdrucksmöglichkeiten einer allgemeinen (objektorientierten) Programmiersprache in der Regel in Verbindung mit einem Kernel für das Laufzeitsystem, das die Kommunikation und Synchronisation der Modellentitäten regelt (siehe [14], [9]). Derartige Ansätze sichern die Korrektheit bezüglich des zeitlichen Verhaltens, bezüglich des Raumbezugs, in dem die Modellobjekte agieren, bestehen für den Modellierer jedoch weder Einschränkungen noch Hilfestellungen.

1.5 Bewertung der Ansätze

Wie bereits bei der Charakterisierung der Konzepte angedeutet, besteht die Schwierigkeit darin, dass ein Raumbezug zwar möglich, konzeptuell aber wenig unterstützt wird. Einzig die partiellen Differentialgleichungssysteme und die Zellularen Automaten fordern explizit einen Raumbezug, sind aber auch diejenigen Ansätze, die die

„klassisch“ kontinuierlich	Trajektorie für eine Zustandsgröße	1-D	
	Bewegung eines Punktes im Raum	(3-D)	
	Partielle Differential-Gleichungen	3-D	finite element models (Wetter, Grundwasser, Mechanik, ...)
diskret	discrete event simulation	(3-D)	Ereignisse können auch einen Ortswechsel beinhalten
Spezialfall	Zellularer Automat mit beliebigen Erweiterungen	2/3-D	
object-orientiert/ individual-based	freie algorithmische Spezifikation des zeitlichen und räumlichen Verhaltens eines Objektes	1/2/3-D	

Abbildung 1: Traditionelle Modellierungsansätze und ihre Fähigkeit, Raumbezug abzubilden

Freiheit des Modellierers gerade dadurch am weitesten einschränken. Andere Ansätze bieten zwar die Möglichkeit, Raumbezug zu integrieren, allerdings verzichten viele Anwendungen trotzdem auf einen expliziten Raumbezug wohl, so die subjektive Meinung des Autors, wegen der wachsenden Kompliziertheit der entstehenden Modelle. Sicher ist jedenfalls, dass ein echtes geographisches Raumkonzept in keines der Modellierungsparadigma Eingang gefunden hat, obwohl die Arbeit mit Geodaten zunehmend an Bedeutung gewinnt.

2 Die GIS-Sicht auf Prozesse in Raum und Zeit

Im zweiten Abschnitt konzentriert sich die Analyse auf die Sicht der Geographischen Informationssysteme und deren Fähigkeiten, Dynamik darzustellen bzw. dynamische Prozesse zu modellieren und zu simulieren.

2.1 Übersicht nach Yattaw

Eine umfassende Übersicht über die Klassifikation von Entities in Raum und Zeit bietet die folgende Tabelle nach Yattaw [13].

Auf der einen Achse findet man die Grundtypen Punkt,

		zeitliches Abstraktionsniveau			
		statisch	dynamisch		
			kontinuierlich (linear, stetig u.a.)	wiederkehrend (periodisch, unberechenbar u.a.)	sprunghaft (epochal, einmalig u.a.)
geometrisches Abstraktionsniveau	Positionen	Baum Denkmal Festpunkt Funkmast	Aufzeichnung von Fahrzeugbewegung (GPS, Radar)	Hot Spot, Austragungsorte (Formel 1 u.a.)	Erklärung zu Schutz- objekten, Verlegung von Amtssitzen
	linear	Straße Eisen- bahn Akauer Flusslauf	Satellitenbahn, Jetstreams (Hurrikan, Tornado)	Zugvögel, Nomaden	Staatsgrenzen, Flussläufe
	flächig	Wald Siedlung Friedhof See	Desertifikation, Anlandung	landwirtschaftliche Nutzung, Klimazonen	Grundoigentumsüber- gang, Umsiedlung, EU Erweiterung
	räumlich	Hochhaus Geoid	Kontinentaldrift, Meeresströmung	Gezeiten, El Niño	Erdbeben, Explosionswolke (Vulkan, Tschernobyl)

Abbildung 2: Raum- und Zeitbezug von Geobjekten nach Yattaw [13]

Linie, Fläche, Volumen, auf der anderen Achse eine Charakterisierung des dynamischen Verhaltens mit den Klassen statisch, kontinuierlich, wiederkehrend und sprunghaft. Analogien zu den beschriebenen Klassen der Modellierungsparadigmen des ersten Kapitels sind offensichtlich, einen neuen Aspekt bieten die periodischen Phänomene. Bei dieser Zusammenstellung ist eine der objektorientierten Modellierung ähnliche Herangehensweise festzustellen, allerdings geht Yattaw nicht über die

rein deskriptive Ebene der Klassifikation hinaus und bietet keine Vorschläge, wie die klassifizierten Dynamiktypen im Sinne einer Modellbeschreibung zu spezifizieren sind.

2.2 Visualisierung von Dynamik im GIS

Unabhängig von der Art der Spezifikation von raumbezogener Dynamik ergibt sich bei Anwendungen im GIS-Umfeld sich regelmäßig die Schwierigkeit, die dynamischen Veränderungen der dargestellten Objekte (Features) im Sinne einer Ergebnisdarstellung zu visualisieren. Um zu sehen, ob die dabei verwendeten Methoden auch für die Dynamik-Spezifikation verwendbar sind, sollen sie hier kurz andiskutiert werden. Im Wesentlichen kann man auf drei grundsätzlich verschiedene Verfahren zur Dynamik-Visualisierung zurückgreifen:

- Einmal die Darstellung der Dynamik in Form einer Animation (gegebenenfalls mit Überblendtechnik) von Einzelbildern (Karten) mit einem interaktiven Schieberegler für die Zeitachse. [2]
- Oder aber im Sinne eines statischen Kartenbildes die Darstellung von n Einzelkarten für n ausgewählte Zeitpunkte bzw. -perioden, wie man es z.B. aus einem Geschichtsatlas kennt. [3]
- Oder die Übertragung der Zeit auf eine geeignete Symbologie der dargestellten Objekte; z.B. unterschiedliche Farben für unterschiedliche Zeitintervalle, in denen das Objekt aktiv ist, oder aber eine farbliche Codierung der Veränderung für einen Attributwert über die Zeit (sog. Change maps [4])

Unabhängig davon, welche Vor- und Nachteile diese Alternativen besitzen soll an dieser Stelle nur festgestellt werden, dass es sich in allen Fällen um eine reine Visualisierung von Ergebnisdaten einer dynamischen Modellierung bzw. eines Simulationslaufes handelt, nicht aber um die Spezifikation der Dynamik selbst. Diese Verfahren helfen folglich bei der Betrachtung im Sinne einer Modellierungssprache für Prozesse in Raum und Zeit nicht weiter.

2.3 Die Perspektive des Modellierers

Im Gegensatz zur klassifizierenden Sicht soll daher nun ein Blick auf ein Geographisches Informationssystem aus der Sicht eines Modellierers geworfen werden, wie

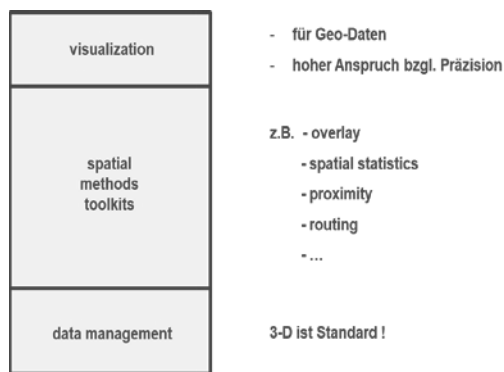


Abbildung 3: GIS-Struktur aus der Perspektive des Modellierers

sie Abbildung 3 zusammenfasst. Aus einer solchen Perspektive sind im Wesentlichen drei Schichten bzw. Funktionalitäten zu unterscheiden, eine Systemarchitektur, wie sie im Übrigen auch die klassische Literatur aus dem Bereich der Softwaretechnik für größere Softwaresysteme im Allgemeinen als sogenannte Three-Tier-Architektur [7] vorschlägt:

- die Datenhaltung, auf der GIS selbstverständlich eine konsistente und effiziente 3D-Datenhaltung anbietet.
- Die Ebene der Tools oder Methoden, auf der eine Vielzahl etablierter und komfortabler Analysemethoden für raumbezogene Daten angeboten wird.
- Die Visualisierung mit präziser und komfortabler Möglichkeit, Kartendarstellungen zu entwickeln.

Aus dieser Perspektive müssten in der Methodenschicht Verfahren zur Spezifikation von Systemdynamik angesiedelt sein. Der Marktführer ESRI bietet allerdings entsprechende Toolboxes nicht an, es steht lediglich eine vollständig freie Programmierschnittstelle auf Basis der Skriptsprache Python zur Verfügung [1]. Im Bereich der OpenGIS [5] ist die Situation ähnlich.

3 Raum und Zeit: GIS versus Simulation

An dieser Stelle soll nun ein Fazit und eine – zugeben - subjektive Beurteilung der Lage versucht werden. Im Wesentlichen lassen sich die beiden Herangehensweisen der vorausgegangenen Abschnitte unter den beiden im Folgenden beschriebenen Aspekten zusammenfassen:

Beide Systeme stellen sich als hochspezialisiert für

ihren Einsatzzweck dar:

GIS konzentrieren sich auf die präzise Abbildung von Raumobjekten in Bezug zur Erdoberfläche und bieten wenig Unterstützung bei der Aufgabe, dynamische Wertänderungen für Attribute von Geo-Objekten zu spezifizieren.

Simulationsparadigmen stellen Konstrukte zur Dynamikbeschreibung zur Verfügung, verfügen aber weder über etablierte Konzepte zur Repräsentation von (geographischem) Raum noch über entsprechende Methoden zur raumbezogenen Analyse.

Daraus ergibt sich der Bedarf einer Zusammenfassung bzw. Integration von Simulationstool und GIS, die wiederum auf zwei verschiedenen Ebenen diskutiert und entsprechend umgesetzt werden kann:

- Funktionale Aspekte einer Integration
Ziel ist eine Vereinigung der funktionalen Möglichkeiten vom GIS und Simulationstool. Damit kommt man schnell auf zwei prinzipiell unterschiedliche Lösungsalternativen, die in der Literatur auch seit langem schon diskutiert sind
 - Modell-Dynamikbeschreibung ins GIS integrieren
 - GIS-Funktionalität ins Simulationsmodell bzw. dessen Spezifikation integrieren
- Softwaretechnische Aspekte der Integration:
Übertragen auf die Ebene der Softwarearchitektur ergeben sich daraus ebenfalls zwei grundsätzlich unterschiedliche Lösungsansätze: Nämlich einerseits,
 - sämtliche Funktionalitäten im Rahmen eines universellen, spatio-geographischen Simulationssystems zusammenzufassen, oder andererseits,
 - sich für den Einzelfall eine problembezogene, individuell angepasste Lösung bzw. Architektur zu implementieren, die dann allerdings den Universalitätsanspruch aufgeben muss.

Darüber hinaus sind in der Praxis eine Vielzahl von Kopplungslösungen zu finden (s.z.B.[10]).

Aus Platzgründen kann auf die unterschiedlichen Lösungen an dieser Stelle nicht detaillierter eingegangen werden. Allerdings fällt auf, dass bei typischen Anwen-

dungen immer wieder Defizite auftreten, die auch mit einer integrierten Lösung nicht anwendergerecht behandelt werden können. Diesen Grundaufgaben will sich der folgende Abschnitt widmen.

4 Ungelöste, problembezogene Anforderungskomplexe

Die folgenden drei Aufgabenbereiche stellen eine aus eigenen Projekterfahrungen und der langjährigen Mitarbeit in der Fachgruppe Simulation in den Umwelt- und Geowissenschaften ([11], [12] und die weiteren Bände der Reihe Umweltinformatik) gewachsene, und damit zwar nur eine persönliche Liste des Autors dar, sie sollen aber dennoch Hinweise geben, auf welchen Themengebieten bei der Arbeit mit Modellen mit Raum- und Zeitbezug noch Defizite auftreten und damit auch Hinweise für zukünftige Forschungs- und Entwicklungsaktivitäten bieten.

4.1 Spezifikation einer "typischen" Trajektorie für eine Entität in Raum und Zeit

Eine immer wiederkehrende Aufgabe für den Modellierer ist es, für ein Objekt, eine Entität, eine "typische" bzw. repräsentative Trajektorie in Raum und Zeit zu beschreiben. Diese Entität kann einerseits ein Individuum sein (z.B. der "typische" Tageslauf eines Menschen mit seiner Mobilität), andererseits können aber auch Gruppen betrachtet werden (z.B. der jährliche Vogelzug).

Eine solche Trajektorie soll entweder aufgrund von punktuell gemessenen (oder aus einer Modellbeschreibung abgeleiteten) Daten durch eine geeignete raumbezogene Interpolation modelliert werden, oder aber sie wird aus einer vorhandenen Verhaltensbeschreibung für die betreffende Entität abgeleitet.

Im ersten Fall benötigen die Modellierer Unterstützung durch geeignete Verfahren der raumbezogenen Statistik, im zweiten Fall muss die Ausdrucksfähigkeit der Modellierungs- bzw. Spezifikationssprache entsprechende Konstrukte zur Verfügung stellen.

In beiden Fällen ist jedoch eine Einbeziehung raumbezogener Rahmenbedingungen notwendig (z.B. die Lage von Gebirgszügen für den Vogelzug, die Topologie von Verkehrssystemen für den Tagesablauf eines Menschen, ...).

4.2 Spezifikation der dynamischen Veränderung der Form einer Linie oder eines Polygons

Der zweite typische Anwendungsfall bezieht sich auf die Änderung der Form von Geo-Objekten mit der Zeit. Die GIS-Grundtypen für Geo-Objekte sind Punkt, Linie, Polygon. Wie kann man die zeitliche Veränderung eines Linienverlaufes spezifizieren (z.B. den exakten Verlauf eines Flusses in einem Tal über einige hundert Jahre) oder aber die Dynamik eines Polygons? Wobei in beiden Fällen wiederum geographische Nebenbedingungen die Dynamik entscheidend beeinflussen und damit auch bei der Dynamikbeschreibung anwenderfreundlich einbezogen werden sollten.

Beispielsweise kann man die Entwicklung einer Stadt an ihren Rändern entlang von vorgegebenen Siedlungsachsen beschreiben, die wiederum durch topographische Nebenbedingungen (Flüsse, Oberflächenrelief, ...) überlagert sind. Eine Verbindung von Zellularem Automaten und der Layerstruktur des GIS für die topographischen Einflussfaktoren liegt in einem derartigen Fall nahe, allerdings wird auf der Ebene der Umsetzung keine generische Lösung auf einer wie auch immer gearteten Metaebene angeboten, den Anwendern bleibt nur die Alternative einer aufwändigen, individuellen Lösung durch eine proprietäre Implementierung.

Ansatz für eine Lösung wären z. B. eine Menge von typischen Mustern, die eine dynamische Veränderung im Sinne eines Vorher-Nachher prototypisch erfassen und beschreiben. Eine gegebene Situation könnte dann zu definierten Zeitpunkten auf Übereinstimmung mit dem Vorher-Muster untersucht und im Falle einer Übereinstimmung eine Transformation gemäß den Vorgaben des Nachher-Musters veranlasst werden, um damit die dynamische Veränderung des Geo-Objektes spezifizieren.

4.3 Spezifikation der dynamischen Veränderung einer Topologie

Der dritte Problembereich erweitert den vorgenannten Punkt um Veränderungen in der zugrundeliegenden Topologie. Wie lässt sich spezifizieren, an welcher Stelle, wann und warum sich eine gegebene Topologie ändert, also neue Punkte, Linien, Flächen entstehen oder bestehende verschwinden. Sämtliche Wachstumsprozesse komplexerer Strukturen führen auf dieses Problemfeld.

Wiederum am Beispiel der Stadtentwicklung: Wann und wo entstehen neue Elemente der Infrastruktur (Geschäfte, Schulen, Straßen, Stadtteile, ...) und wie sind

diese mit den bereits bestehenden Elementen der Topologie verbunden.

Im Bereich der Informatik gibt es beispielsweise Arbeiten zu Graphgrammatiken, die eine Ersetzung einer Teilstruktur eines Graphen (also auch einer Topologie) durch eine neue Struktur formal zu beschreiben erlauben [8]. Inwieweit derartige theoretische Ansätze in eine praxisnahe Spezifikation für topologische Strukturveränderungen zu übertragen sind, verdient sicherlich einer intensiveren Untersuchung.

5 Zusammenfassung

Der vorliegende Artikel bearbeitet das Thema Raumbezug in dynamischen Modellen auf drei unterschiedlichen Ebenen:

Erstens das Problem der Integration von GIS- und Simulationsmethoden auf der Ebene der Softwaretechnik. Hier sind die beiden extremen Alternativen

- a) ein universelles Softwaresystem für Modelle mit Raum- und Zeitbezug oder aber
- b) eine nicht verallgemeinerbare, heuristische, auf das individuelle Problem zugeschnittene Lösung.

Unabhängig von der softwaretechnischen Seite möchte der Artikel die Aufmerksamkeit allerdings verstärkt auf zwei weitere Problemfelder richten:

Das ist einerseits der Bedarf, eine raumbezogene Statistik für die Modellierung aufzubereiten und mit einer anwendungsnahen Bedienoberfläche zur Verfügung zu stellen. Für dieses Problemfeld ist eine Einbeziehung komplexer, raumbezogener Restriktionen und Einflüsse von entscheidender Bedeutung.

Andererseits stellt der Artikel das Fehlen eines adäquaten Formalismus zur Spezifikation von Änderungen in Form und Topologie von raumbezogenen Objekten fest und postuliert gerade auch auf diesem Feld Bedarf für künftige Forschungsaktivitäten.

Literatur

- [1] ArcGIS API for Python, <https://developers.arcgis.com/python/>, Zugriff 25.06.2018, 14:10
- [2] Esri ArcMap Online-Hilfe: Erstellen von Animationen in ArcGIS, <http://desktop.arcgis.com/de/arcmap/10.3/map/animation/about-building-animations-in-arcgis.htm>, Zugriff 25.06.2018, 14:50
- [3] Kimerling, A.J., Buckley, A.R., Muehrcke, P.C., Muehrcke, J.O.: Map Use, Chapter 8, Quantitative Thematic Maps, Esri Press, Redlands California, 8th edition, 2016, p 206
- [4] Kimerling, A.J., Buckley, A.R., Muehrcke, P.C., Muehrcke, J.O.: Map Use, Chapter 8, Quantitative Thematic Maps, Esri Press, Redlands California, 8th edition, 2016, p 209
- [5] OGC® Open Geospatial APIs - White Paper OGC <http://docs.opengeospatial.org/wp/16-019r4/16-019r4.html> Zugriff 25.06.2018, 14:10
- [6] Ortman, Jörg: „Ein allgemeiner individuenorientierter Ansatz zur Modellierung von Populationsdynamiken in Ökosystemen unter Einbeziehung der Mikro- und Makroebene“; Dissertation am Fachbereich Informatik, Universität Rostock, 1999
- [7] Rechenberg, P., Pomberger, G.: Informatik Handbuch, Stichwort Architektur von SW-Systemen, 4.Auflage, Hanser München 2006, S.812
- [8] Schneider, H.J.: Graph Transformations - An Introduction to the Categorical Approach, Vorlesungsunterlagen, Universität Erlangen-Nürnberg, Department Informatik, Lehrstuhl für Programmiersysteme, <https://www2.informatik.uni-erlangen.de/teaching/SS2006/GraTra/slides/index.html> Zugriff 25.06.2018 14:50
- [9] Uhrmacher A.M.: Dynamic Structures in Modeling and Simulation - A Reflective Approach. ACM Transactions on Modeling and Simulation, Vol.11. No.2:206-232, 2001
- [10] Simulationsmodell und Geographisches Informationssystem Kopplungsalternativen am praktischen Beispiel in: Cremers, A.B.; Greve, K. (Hrsg.): Umweltinformatik 2000, 12.Internationales Symposium, Bonn, 2000, Metropolis-Verlag, Marburg 2000, S.45-58
- [11] Wittmann, Jochen (Hg.): Simulation in Umwelt- und Geowissenschaften: Workshop Berlin 2017, S. 1-177, Shaker Verlag, Aachen, 2017, ISBN 978-3-8440-5492-7
- [12] Wittmann, Jochen; Thiel-Clemen, Thomas (Hg.): Simulation in Umwelt- und Geowissenschaften - Workshop Hamburg 2016, S. 1-154, Shaker, Aachen, 2016, ISBN 978-3-8440-4770-7
- [13] Yattaw Nancy J. (2013) Conceptualizing Space and Time: A Classification of Geographic Movement, Cartography and Geographic Information Science, 26:2, 85-98, DOI: 10.1559/152304099782330734
- [14] Zeigler, B.P.: Object-Oriented Simulation with Hierarchical, Modular Models. Academic Press, London, 1990

Proposing Multiple-Criteria Ranking to Simulate Building Renovation in Cities

Arjun Jamil^{1*}, Nils Weiss¹, Thomas Preisler¹, Wolfgang Renz^{1*}

¹Faculty of Engineering and Computer Science, Hamburg University of Applied Sciences, Berliner Tor 7, 20099 Hamburg, Germany *{arjun.jamil, wolfgang.renz}@haw-hamburg.de

Abstract. Building renovation plays a huge role in reducing CO₂ emissions and energy demand. To understand the effects of different renovation strategies, expert users and decision makers need to explore renovation scenarios both spatially and temporally. The ongoing GEWISS project aims to do this by providing a GIS (Geographical Information System) to visualize energy demand per building as well as an interface to simulate possible renovation paths to its users.

This white paper presents a novel proposal to simulate future renovation paths by utilizing different user scenarios containing multiple selection criteria to rank current building stock and consequently use optimization to pass them through the user selected renovation strategy in the most optimal way.

Keywords. Urban planning; Building renovation; Multiple criteria decision making;

Introduction

Existing building stock contributes to up to 40% of the EU's total CO₂ emissions [1]. Renovation of building stock to national standards such as the German energy saving ordinance (EnEV 2014) is critical to reducing Germany's carbon footprint and energy usage. The GEWISS¹ (Geographical Heat Information and Simulation System, 2014 - 2019) project aims to analyse and visualise heating demand and CO₂ emissions based on the existing building stock of cities and to construct an expert tool that would help decision makers to simulate different scenarios and find the ones that would have the maximum cost effectiveness in reducing emissions and heating demand [2]. As can be seen from its contribution to the Hamburg Wärmekataster², the GEWISS project approaches the problem in a spatial context, effectively making it a GIS tool with additional simulation support. Another question that the decision makers in Hamburg would like to explore is the organic expansion of district

heating grids in Hamburg; the GEWISS project is expected to provide insights on which areas are worth expanding into with the district heating grid. Hence the simulation ideally should also consider the district heating parameters. By knowing what future grid expansions might provide the best reduction in emissions and heating demand, the city can expand the grid in the most cost-efficient way.

Using classifications provided by third party project partners to the project, an underlying deterministic data model is created, one which contains select information of all current buildings in the city of Hamburg (Figure 1).



Figure 1 Map extract containing building geometry for each unique building (taken from [2])

This data is obtained on the basis of the Hamburg ALKIS database³ with an estimation of IWU building type [3] also for those 50% of houses where the building year is unknown [4]. Using building geometry as a metric, all the existing buildings are classified into different building types. These building types have standardised heating demand and other identifying parameters. Therefore, with this standardised building typology, a data model

¹ Project page (German): <http://gewiss.haw-hamburg.de/>

² <http://www.hamburg.de/energiewende/waermekataster/>

³ <http://suche.transparenz.hamburg.de/dataset/alkis-liegenschaftskarte-ausgewahlte-daten-hamburg>

can be constructed, with standardised heating demand and emissions for every building unit. This paper uses this underlying data model to simulate building renovation. The data model is further explained in Section 1.

The GEWISS simulation team followed the rapid prototyping methodology and started with mock-ups based on multi-agent simulation tools [2]. The current approach is to rank all buildings in the data model and stochastically renovate ranked buildings in order according to a global renovation rate, for every year of the simulation run. The building's ranking score is affected by the following *impact factors*-

- Building type factor, maps the impact factor for different building types (IWU for residential and an Ecofys⁴ typology for non-residential buildings) for each year of the simulation.
- Location factor, mapping of the building's spatial allocation for each year of the simulation. Either on the level of districts, quarters or statistical areas in Hamburg to map socio-economic factors.
- Property factor, maps different impact factors for private, cooperative, company-owned and public owned property for each year of the simulation.
- Heritage protection factor, maps factors for buildings under monument protection for each year of the simulation.

To calculate scoring value, above parameters are factored only globally for every building in the current year of the simulation and are multiplied with some inherent factors, to give every building a rank. After the ranking process, a predefined percentage of the ranked buildings are stochastically renovated to a higher building standard.

The current approach is limited in its flexibility, as the impact factors affecting the ranking score are based on single criteria. Therefore, only making it possible to affect the ranking of e.g. one-family houses and houses in the quarter Altona but not just 'single family houses in Altona'. Also, the current approach does not allow for temporal restrictions on selection of buildings, or the input of possible investment scenarios.

Based on the observation that the scenario modelling abilities are crucial for the applicability in studying what-if questions under various assumptions, this paper proposes a novel solution to the current limitations. First, the simulation would be user manipulatable, i.e. the user would be able to temporally and spatially identify a subset of buildings. Then, using a multi-criteria selection algorithm, the ranking of this building subset would be

modified, within the entire set of buildings. This approach to referred to as the "Modifier Approach", further explained in Section 2.1. Finally, an outlook towards optimization of renovation to meet user criteria is provided in Section 3, along with suggestions for future research.

1 Initial Parameters

To understand the problem statement the GEWISS simulation aims to solve, a look at the available data model must be taken. The following subsection explores the foundational data model our simulation uses. Further on, the simulation's inherent properties are also explained with an outro towards the renovation thresholds in the later subsections.

1.1 Available and Assumed Data

The data model in consideration contains all current buildings in Hamburg as a unique entry. Their data is classified with the following attributes-

1. B_{ID} -Building Identifier (unique ID)
2. B_{ort} -Location (District, Borough)
3. B_{jahr} -Construction year (estimated IWU epoch)
4. B_{typ} -Building type (residential/non-residential)
5. B_{ss} -Current renovation level (standard of renovation)
6. B_{own} -Ownership of Building (Private, Public)
7. B_{hs} -Current heating system (Gas, District heating, etc)
8. B_{fw} -Distance to the nearest district heating connection
9. B_{DMS} -Heritage protection status

An important parameter is current renovation level B_{ss} of a building. In our model, the renovation level is classified into 3 stages-

- 0: Unrenovated state
- 1: Renovated to EnEV 2014 standards
- 2: Renovated to Passive house standards

Thereby, a renovation level of 0 means that the building has never been renovated since it was built, a renovation level of 1 means that the building has been renovated according to the German Passive House Standard which is also defined in the latest German energy saving ordinance (EnEV 2014)⁵.

The simulation would operate on the selection of buildings using the above written parameters, as well as *inherent factors* that the simulation operates upon. These factors are explained in the following subsection.

1.2 Inherent Simulation Operators

The simulation operates upon two assumptions that are

⁴ Ecofys.com is GEWISS project partner

⁵ (German) http://www.enev-online.com/enev_2014_volltext/

made of the simulated environment [2]-

1. Age incentive I_{rp} – As age since last renovation increases, renovation chance of a building increases. Conversely; newly renovated buildings will not be considered for renovation again immediately.
2. Neighborhood Effect I_n - A building’s renovation affects the renovation chance of its neighbors. If a building is neighbored by renovated buildings, its renovation chance also increases.

These inherent operators define the environment of the simulation. They provide a base ranking to each building that is then modified by user selected *modifiers*. Finally, depending on user prioritised modifiers, the selected buildings would then be passed through the selected renovation threshold. These are explained in the following subsection.

1.3 Renovation Thresholds

The simulation also relies on the following rates to change the state of considered buildings-

1. Global renovation rate R_g - Percentage of the buildings that would be renovated.
2. Passive house rate R_p - Percentage of the renovated houses that will be renovated to the passive house standard ($B_{ss} = 2$)
3. Rate of change of heating system R_h - Percentage chance of the change of heating system.

Using the criteria and factors explained above, the next section delves further into the proposed simulation logic.

2 Proposed approach

The simulation engine is designed as a combination of multiple components. The simulation engine accepts data from the user through the “scenario editor” GUI, which supplies the simulator with the user scenarios. The simulator then outputs one list containing ranked renovation candidates. These buildings are then passed through the optimizer, that decides the optimal renovation levels for each building according to the renovation strategy. This process is pictured in figure 2.

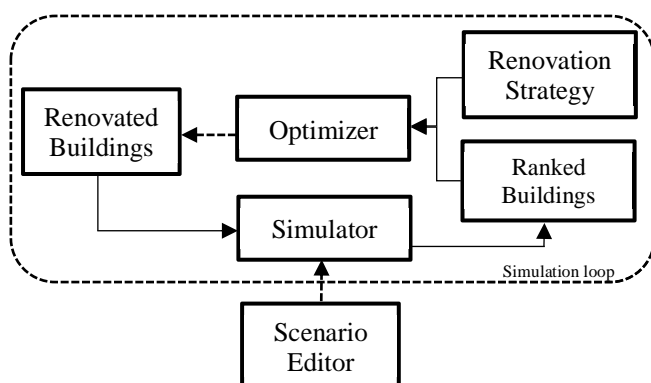


Figure 2 Proposed simulation logic. The simulation runs the ranking

and renovation process for each simulation year.

Every year of the simulation run, every unique building is assigned a renovation score (S_i). The simulation ranks the buildings using the formula outlined here-

$$S_i = I_{rp} * I_n \tag{1}$$

The simulation and optimizer logic are explored in detail below, with the help of an illustrative example.

2.1 Modifier Approach

The scenario editor uses GUI elements we call modifiers. Modifiers are criteria-based selectors, selecting a set of buildings according to user-specified criteria and modifying the rank of these buildings within the simulation. They are active for a specified finite period within the simulation. When a building satisfies the modifier criteria, the building rank of a building is altered by a user assigned *modifier impact factor* (I_m), in addition to the inherent operators in the simulation. Equation (2) demonstrates renovation score S_i , for a building that matches criteria that was selected for modifiers $m_1, m_2, m_3 \dots m_n$, during set time frame.

$$S_i = I_{rp} * I_n * I_{m_1} * I_{m_2} * I_{m_3} \dots I_{m_n} \tag{2}$$

Where I_m refers to the respective modifier impact factor set by the user. The primary objective of this approach is to allow the user to manipulate the simulation, to model stories that can represent complex real-world events. These stories can represent financial promotion, legal incentives, or target emissions to be achieved. The following example is considered-

“All unrenovated public buildings built before 1960 are renovated from gas to district heating (FW) in 2025”

This user story uses the following parameters- renovation level (B_{ss}), ownership of Building (B_{own}), construction year (B_{jahr}), heating system implemented (B_{hs}) and the time period of incentive. These parameters are clustered together and fed into the simulation engine in the form of a modifier. Assuming the *modifier impact factor* (I_{m_1}) to be 2, the renovation score (S_i) of buildings selected by the modifier can be represented by-

$$S_i = I_{rp} * I_n * 2 \tag{3}$$

To illustrate the renovation score with the example given above, table 2 is input to the simulator as building data.

Build- ing ID	Renova- tion level	Ownership	Construction Year	Heating system
B1	0	Private	1960	District
B2	0	Public	1920	Gas
B3	1	Public	1970	Gas
B4	2	Private	2000	District
B5	0	Public	1940	Gas

Table 2: Illustrative data input into simulation.

When the simulation is in the year 2025, the renovation ranking of selected buildings would be modified,

which in this case are B2 and B5. Assuming values of Inherent operators, table 3 demonstrates renovation score for the above buildings for simulation year 2025.

Build- ing ID	Renovation score	Neighborhood effect	Modi- fier 1	Renovation score
B1	1.2	1	NA	1.2
B2	1.3	1	2	2.6
B3	0.8	2	NA	1.6
B4	0.5	1	NA	0.5
B5	1.9	1	2	3.8

Table 3: Illustrative ranking from simulation. Inherent factors are assumed values. B2 and B5 are ranked highest; modifiers are able to prioritise user selected buildings.

As can be seen in the above example, modifiers are able to prioritize buildings that match user specified criteria. One simulation run can take multiple modifiers into consideration, allowing the user high levels of control over the way the simulation ranks houses. Taking the inherent operators into account, the simulation allows for differing values renovation rank every simulation year.

Modifiers are a form of *reduced processing strategy*, implemented to allow control over the decision overhead of the simulation, by allowing for multiple criteria ranking. The following section explains the steps the optimization engine takes to renovate buildings against a user set renovation strategy and concludes with an outlook.

3 Conclusion & Outlook

3.1 Renovation Strategies & Optimization

The user might also like to explore financial investment scenarios, or even heating demand/emission reduction scenarios. To perform these simulations, the ranked buildings must be evaluated against the renovation strategy, and then different renovation paths must be compared. This task is assigned to the optimizer. The optimizer uses global renovation rate R_g , passive house rate R_h and heating system change rate R_h as *renovation thresholds*. As seen in figure 2, the optimizer takes in the ranked building list from the simulation, and then passes them through the renovation thresholds optimally to match the user set renovation strategy. Given below is a list of renovation strategies considered-

1. Money invested annually to incentivize renovation.
2. Heating demand reduction
3. CO₂ emission reduction

In the case of financial incentive, the user would specify a lump sum amount of money that would be invested into the city to renovate buildings through the simulation run. Using modifiers, the user can ensure that selected buildings are likelier to receive financial support. The end result in this case would show reduction in heating demand and CO₂ emissions. When the strategy is to reduce heating demand and CO₂ emissions, the user could

specify targets that the simulation must achieve. The end result in this case would show money invested over the simulation run.

Depending on the renovation state of the building, there could be up to 2 state changes (0→1, 1→2) and/or a change in the heating system of the house. All these state changes result in differing levels of cost, and demand/emissions reduction. Furthermore, these state changes are controlled with the renovation threshold described earlier. The optimizer must then, considering the set renovation strategy, find the best candidates in the ranking list to pass through the renovation thresholds.

3.2 Outlook to Future Development

Considering the optimization search space outlined in the previous subsection, the authors suggest the use of *metaheuristics*, with a preference towards ant colony optimisation (ACO). This would allow the optimizer to roam the search space for optimal solutions parallelly, reducing processing time and simulation overhead.

Future development might consider the implementation of renovation thresholds and strategies at a local scale, as the current implementation defines renovation thresholds and strategies globally. Using *exclusion sets*, the user would be able to restrict different renovation strategies spatially and temporally, allowing full control over the simulation results.

Finally, making the data model format open source might aid other researchers in the development of expert simulation and visualisation tools for other German cities.

References

- [1] Balaras, Constantinos A., et al. "European residential buildings and empirical assessment of the Hellenic building stock, energy consumption, emissions and potential energy savings." *Building and environment* 42.3 (2007): 1298-1314.
- [2] Preisler, Thomas, et al. "Towards an agent-based simulation of building stock development for the city of hamburg." *Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on. IEEE, 2017.*
- [3] Diefenbach, Nikolaus, et al. "Deutsche Wohngebäudetypologie - Beispielhafte Maßnahmen zur Verbesserung der Energieeffizienz von typischen Wohngebäuden," IWU - Institut Wohnen und Umwelt, Tech. Rep., 2015. [Online]. Available: http://www.building-typology.eu/downloads/public/docs/brochure/DE_TAB-ULA_TypologyBrochure_IWU.pdf
- [4] Dochev, Ivan, et al. "Assigning IWU building types to buildings in the Hamburg ALKIS," 2017.

A Difference Equation Approach for Modeling Arterial Wave Reflection

Laura Lotteraner^{1*2}, Bernhard Hametner², Siegfried Wassertheurer², Felix Breitenecker¹

¹Institute for Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *laura.lotteraner@tuwien.ac.at

²Center for Health & Bioresources, AIT Austrian Institute of Technology, Giefinggasse 4, 1210 Vienna, Austria

Abstract. A simple difference equation model is presented that describes the development of blood pressure in the ascending aorta over time. Reflections both at one and more sites in the arterial system as well as re-reflections of reflected waves at the heart are considered. Model parameters include reflection magnitude, pulse wave velocity, reflection site positions and heart rate. The model is discussed both analytically and numerically by implementation in MATLAB. Comparison of blood pressure curves generated by the model and blood pressure curves measured in patients demonstrates that the model is able to represent reality to some extent. Results can be improved by adding reflection sites which motivates further research.

Introduction

Elevated blood pressure is a major cause of death mainly but not only in developed countries. In order to improve diagnosis and prevention it is crucial to understand the underlying dynamics. Mathematical models can help with this. While many aspects of blood pressure have been considered by previous researchers, the concept of re-reflections of blood pressure waves at the heart has been neglected in most models and is only poorly understood so far. In this paper a simple difference equation model for aortic blood pressure is presented that focuses on the characterization of pulse wave reflections and re-reflections.

1 The Model

Model Design. Even though blood pressure dynamics are time-continuous, time-discrete difference equations are a valuable tool to describe reflection effects. The arterial system is considered a single uni-

form tube with the heart, connected to the system via the aortic valve (AV), on one end and an elastic closed boundary, referred to as distal reflection site (RS), on the other. This idea has been applied in many different models and has been proved to be a reasonable approach by different authors [1]. A measuring site (MS) is fixed in the ascending aorta, close to the aortic valve. This so-called aortic root is of particular interest because it allows to assess the interaction of reflections in the vascular system and the heart [2]. The distance between measuring site and distal reflection site will be named effective length in this paper, a terminology that expresses the reduction of multiple reflection effects to one. Analogously, the distal reflection site is also named effective reflection site. The model describes the development of blood pressure at this fixed measuring site over time. Therefore all spatial information has to be transformed into temporal information.

Model Parameters. The parameters of the model describe both the heart and the arterial system. Reflections are described by real-valued reflection coefficients, $0 < R_D < 1$ for the distal reflection site and $0 \leq R^{av} \leq 1$ for the aortic valve. The reflection coefficients include information on the elasticity properties of the arterial bed and the damping of pulse waves as they propagate through the blood. While R_D is assumed to be constant, reflection at the aortic valve can be chosen to be time-dependent, i.e. $R^{av} = (R_k^{av})_{k \in \mathbb{N}}$ with the time index k where the length of a time step will be given by the step size τ . Typically, different values are chosen during systole, when the valve is open, and diastole, when the valve is closed. The number of time steps it takes a pressure impulse to travel from the measuring site to the distal reflection site and back will be denoted by the time constant $t_b \in \mathbb{N}$. It includes information on both the distance between measuring site and reflection site

and pulse wave velocity. In a more complex description of reflection phenomena the time of return of the reflected wave is also affected by the phase shift that occurs during reflection [3]. Due to the simplification of the reflection coefficient in this paper this time delay must be included in the time constant. Analogously, the number of time steps it takes a pressure impulse to travel from the measuring site to the aortic valve will be denoted by $t_f \in \mathbb{N}$. Since the measuring site is very close to the valve there holds $t_f \ll t_b$. Not only does the heart serve as a reflection site, it also is the driving force for pressure dynamics. During systole blood is pumped into the system, increasing the pressure. This phenomenon is represented by the discrete function $(P_k^{in})_{k \in \mathbb{N}}$.

Model Equations and Solutions. In order to formulate the model equations total pressure P_k is split into its forward and backward components P_k^f and P_k^b , i.e. $P_k = P_k^f + P_k^b$. Using all the aforementioned parameters the difference equation system

$$P_k^f = P_k^{in} + P_{k-2t_f}^b R_{k-t_f}^{av}, \quad k \in \mathbb{N}_{>2t_f}, \quad (1)$$

$$P_k^b = P_{k-t_b}^f R_D, \quad k \in \mathbb{N}_{>t_b} \quad (2)$$

can be obtained. Figure 1 illustrates the idea behind equation (1).

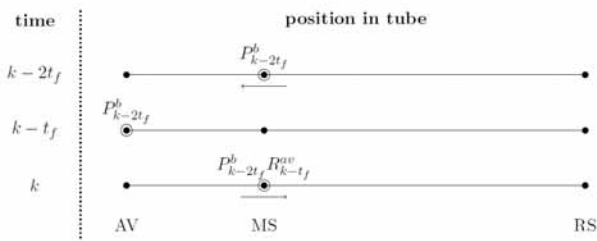


Figure 1: Backward pressure is reflected at the time-dependent aortic valve (AV) as described in equation (1) and returns to measuring site (MS).

Inserting (1) into (2) and vice versa yields the two independent equations

$$P_k^f = P_k^{in} + P_{k-2t_f-t_b}^f R_D R_{k-t_f}^{av}, \quad k \in \mathbb{N}_{>2t_f+t_b},$$

$$P_k^b = P_{k-t_b}^{in} R_D + P_{k-2t_f-t_b}^b R_D R_{k-t_b-t_f}^{av}, \quad k \in \mathbb{N}_{>2t_f+t_b}.$$

Obviously both equations are of order N where $N = 2t_f + t_b$. Since they only define a recurrence relation for $k > N$, initial values P_k^f and P_k^b , $k = 1, \dots, N$ are requi-

red. Then standard literature [4] guarantees the existence of unique solutions for the resulting initial value problems. The initial values are chosen as $P_k^f = P_k^{in}$, $k = 1, \dots, N$, $P_k^b = 0$, $k = 1, \dots, t_b$ and $P_k^b = P_{k-t_b}^{in} R_D$, $k = t_b + 1, \dots, N$. Using a solution method for linear difference equations of fixed but unknown order [5] yields the solutions

$$P_k^f = P_k^{in} + \sum_{j=1}^{\lfloor \frac{k-1}{2t_f+t_b} \rfloor} P_{k-j(2t_f+t_b)}^{in} R_D^j \prod_{m=1}^j R_{k-(m-1)(2t_f+t_b)-t_f}^{av}$$

and

$$P_k^b = P_{k-2t_b}^{in} R_D + \sum_{j=1}^{\lfloor \frac{k-t_b-1}{2t_f+t_b} \rfloor} \left(P_{k-j(2t_f+t_b)-t_b}^{in} R_D^{j+1} \prod_{m=1}^j R_{k-(m-1)(2t_f+t_b)-t_f-t_b}^{av} \right)$$

for $k \in \mathbb{N}_{>2t_f+t_b}$.

Model Properties. Using stability theory for linear difference equation systems [4] it can be shown that all the solutions are stable and bounded under the assumption $R^{av} \equiv 1$. For this purpose both independent equations of order N have to be transformed to an equivalent difference equation system of order one with system matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & \dots & 0 \\ \vdots & 0 & 1 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & 0 \\ 0 & \vdots & & & \ddots & 1 \\ R_D & 0 & \dots & \dots & \dots & 0 \end{pmatrix} \in \mathbb{R}^{N \times N}.$$

Then, for any matrix norm $\|\cdot\|$ there holds

$$\lim_{k \rightarrow \infty} \|A^k\| = 0.$$

Let \tilde{A} be the system matrix of the equivalent difference equation system of order one with variable R^{av} . Because of

$$\|\tilde{A}^k\| \leq \|A^k\|.$$

for any matrix norm $\|\cdot\|$ the same result is valid for this system.

Both functions $(R_k^{av})_{k \in \mathbb{N}}$ and $(P_k^{in})_{k \in \mathbb{N}}$ are periodic with the same period length K that equals the duration of one cardiac cycle, i.e. the sum of systole and diastole duration. Therefore, also the solutions of the model become periodic with period length K after a certain setting time that depends on the choice of the reflection coefficient R_D . Under the assumptions made for the model it is possible to set $t_f = 0$, i.e. shift the measuring site towards the heart, without major changes in the results. In order to simplify analysis this will be done in the rest of the paper.

2 Implementation and Results

After theoretical discussion the model is implemented in MATLAB. Numerical analysis is conducted by comparing plots of generated curves where one parameter is varied while the others are held constant. Both the influence of parameters on the shape of the curve and on the minimum (diastolic pressure) and maximum (systolic pressure) values are analyzed.

Parameter Values and Implementation. Parameter ranges are determined according to previous studies. Systole duration typically is 0.3 seconds, and diastole duration typically is 0.7 [6]. They are included in the model via the choice of the input function and the reflection at the heart. The distal reflection coefficient should be chosen around $R_D = 0.5$. Different authors give different values that are computed using different methods [3, 7, 8]. The same goes for the return time from the distal reflection site where both effective length and pulse wave velocity can be determined in various ways [3, 8, 9]. A typical value is 145 milliseconds which is much shorter than systole duration [10]. This time constant can be transformed to a discrete number of time steps using the number of time points per second $1/\tau$. Implementation of the model using the previously defined parameters and initial values in MATLAB is straightforward.

Input Function and Reflection at the Heart. Two different functions are compared for P^{in} . Both functions are zero during diastole when the valve is closed and the heart produces no output. During systole a half sine that is motivated by other authors' results [11, 12] is compared to the much simpler constant one function. The half sine (see figure 2) that considers the opening

and closing time of the aortic valve yields more realistic results than the strongly simplified constant function (see figure3).

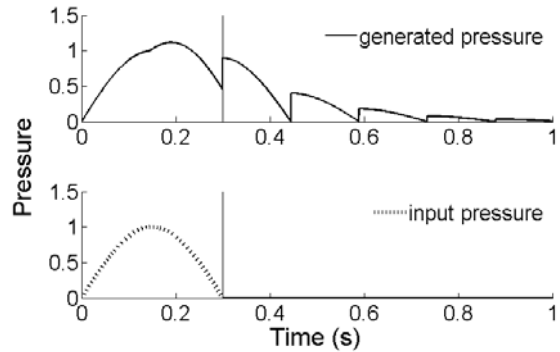


Figure 2: Generated pressure curve (solid) with corresponding input function (dotted). A half sine is chosen during systole. Reflection at the heart is zero during systole and one during diastole. End of systole is indicated by a vertical line. One cardiac cycle.

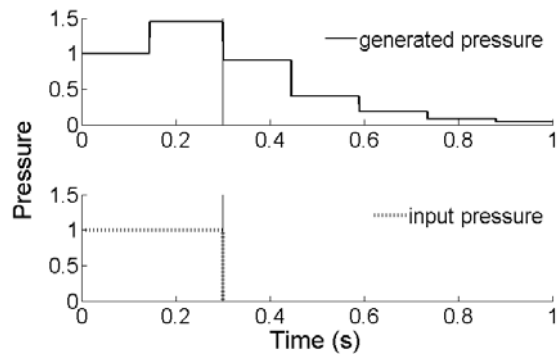


Figure 3: Generated pressure curve (solid) with corresponding input function (dotted). The constant one function is chosen during systole. Reflection at the heart is zero during systole and one during diastole. End of systole is indicated by a vertical line. One cardiac cycle.

For modeling reflection at the heart three different functions can be compared. A step function assumes zero during systole when the valve is open and all waves are absorbed and one during diastole when the valve is closed and all waves are reflected. As before, motivation for this function is given by other authors [12]. Instead, the simpler constant functions zero and one can be used. They produce smoother curves with less distinct spikes, see figure 4. In the following analysis the

reflection is assumed to be a step function.

Results. Figure 5 compares a curve generated by the model (solid) with a measured blood pressure curve (dotted) taken from a data set that was collected for a previous study [13]. Both curves are normalized to the same pulse pressure to increase comparability. The focus of this paper is the shape of blood pressure curves. For the generated curve the input function P^{in} is chosen as a half sine during systole and zero during diastole. Reflection at the heart, R^{av} , is chosen as the constant one function during systole and zero during diastole. A vertical line indicates the end of systole.

Even though the modeled curve shows more spikes and is generally less smooth, the main characteristics of an aortic blood pressure curve [14] are represented. There is a (relatively) fast systolic upstroke, a notch at the end of systole and a slow diastolic decay.

Increasing the reflection coefficient R_D significantly increases systolic pressure and the prominence of spikes while diastolic pressure remains close to zero, yielding an increase in pulse pressure, i.e. the amplitude of the wave. The curves look realistic only for a sufficiently small value of R_D where the threshold depends on the choice of other parameters.

Return time t_b negatively influences systolic and pulse pressure while diastolic pressure remains zero. All spikes occurring during diastole become wider and more prominent with increasing return time, see figures 6 and 7. The width of spikes during diastole equals the return time, therefore representing the single reflections.

Choosing a return time that is larger than systole duration does not lead to realistic results.

3 Model Refinement

The curves' shapes can be improved by adding more reflection sites. It has been argued before that at least two peripheral reflection sites are necessary to model the entire arterial system - one for the upper and one for the lower body half [15, 16].

The Model. While modeling two reflection sites is often accomplished with an asymmetric T-tube model [17], the reflection sites in the present paper are assumed to be completely independent from each other. The

equations are

$$\begin{aligned} P_k^f &= P_k^{in} + P_{k-2t_f}^b R_{k-t_f}^{av}, & k \in \mathbb{N}_{>2t_f}, \\ P_k^b &= P_{k-t_b^1}^f R_D^1 + P_{k-t_b^2}^f R_D^2, & k \in \mathbb{N}_{>t_b^2} \end{aligned}$$

where R_D^1 and R_D^2 are the reflection coefficients of the two reflection sites. Their respective return times are t_b^1 and t_b^2 with $t_b^1 < t_b^2$.

Inserting each of the equations into the other one and prescribing initial values P_k^f and P_k^b for $k = 1, \dots, N = 2t_f + t_b^2$ yields an initial value problem. Standard literature [4] guarantees the existence of unique solutions. They cannot be expressed explicitly. It is possible, however, to prove analytically that the solutions of the homogeneous problems are bounded as long as $R_D^1 + R_D^2 < 1$. In the following, $t_f = 0$ will be assumed.

Results. Comparison with the original model reveals that the prominence of spikes during diastole is decreased significantly, see figure 8. In particular, blood pressure does not decrease to zero after each spike, presumably a result of the superposition of reflected waves. The main characteristics of a blood pressure curve are represented.

It can be observed that the generated curves are both bounded and periodic after a certain setting time, even for $R_D^1 + R_D^2 \geq 1$.

Again, the influence of different parameters can be assessed. While the general effects are the same as in the model with only one reflection site, the different influences of the two reflection sites can be compared. The reflection coefficients have similar influences on systolic and diastolic pressure as long as both are small. When they exceed a certain threshold that depends on the choice of the other parameters, R_D^1 increases systolic blood pressure and R_D^2 increases diastolic blood pressure more significantly. Physiologically meaningful results can only be obtained when the sum $R_D^1 + R_D^2$ does not exceed some threshold value that depends on the return times. Each of the reflection coefficients has to be chosen smaller than the reflection coefficient in the original model.

The influence of systole and diastole duration is the same as can be observed in the original model with one reflection site. The same is valid for the influence of both return times.

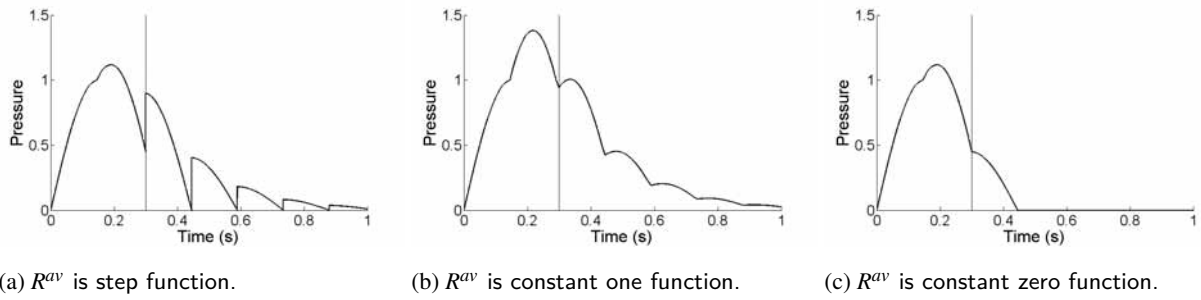


Figure 4: Comparison of different shapes of R^{av} . Input function is a half sine during systole and zero during diastole. End of systole is indicated by a vertical line. One cardiac cycle.

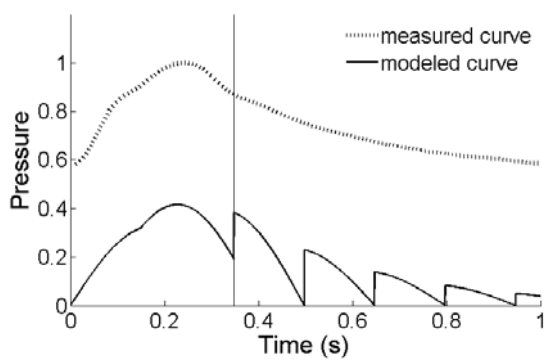


Figure 5: Comparison of a modeled (solid) and a measured (dotted) blood pressure curve. End of systole is indicated by a vertical line. One cardiac cycle.

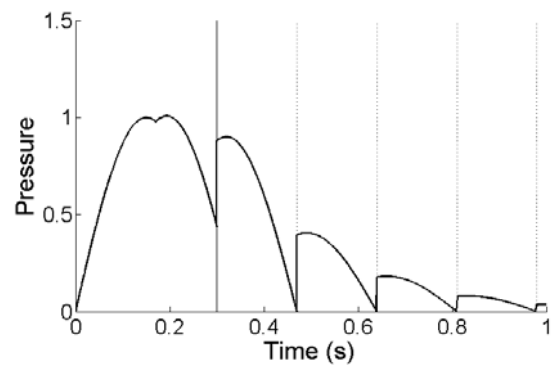


Figure 7: Return time 170ms. End of systole is indicated by a solid vertical line. Return times are indicated by dotted vertical lines. One cardiac cycle.

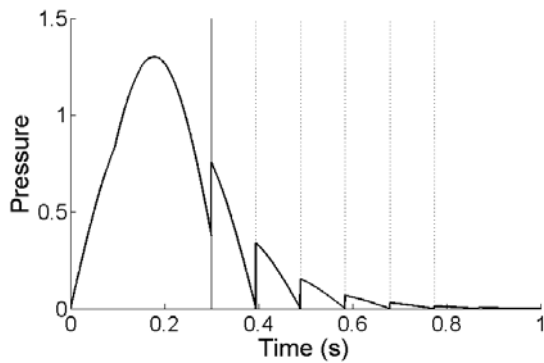


Figure 6: Return time 95ms. End of systole is indicated by a solid vertical line. Return times are indicated by dotted vertical lines. One cardiac cycle.

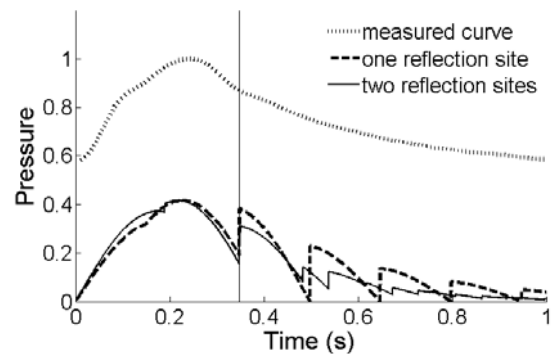


Figure 8: Comparison of a measured curve (dotted), a curve modeled with one reflection site (dashed) and a curve modeled with two reflection sites (solid). End of systole is indicated by a vertical line. One cardiac cycle.

4 Conclusion and Outlook

All the results obtained by theoretical and numerical analysis correspond to effects observed in reality

and therefore indicate that modeling aortic blood pres-

sure with difference equations is a reasonable approach. Instead of a variable reflection at the heart constant functions can be chosen. If the heart is considered to be a total reflector during the entire cardiac cycle, or equivalently, $R^{av} \equiv 1$, estimating formulas for systolic and diastolic pressure can be found, simplifying analysis.

The model can be further refined by adding more distal reflection sites. Also, more complex functions to model reflections at the heart that consider opening and closing of the valve can be chosen. Then, the model can be fit to measured curves by determining the correct parameter combination. From this point, a variety of applications can be considered, such as implementing a wave separation [18] or wave tracking [19] algorithm.

References

- [1] Westerhof, N., Stergiopoulos, N. and Noble, M.I.M. *Snapshots of Hemodynamics*. Boston: Springer; 2005. 192 p.
- [2] Burattini, R., Knowlen, G.G. and Campbell, K.B. Two arterial effective reflecting sites may appear as one to the heart. *Circulation Research*. 1991; 68: 85–99, doi: 10.1161/01.RES.68.1.85.
- [3] Westerhof, B.E., van den Wijngaard, J.P., Murgo, J.P. and Westerhof, N. Location of a reflection site is elusive. *Hypertension*. 2008; 52:478–483. doi: 10.1161/HYPERTENSIONAHA.108.116525.
- [4] Agarwal, R.P. *Difference Equations and Inequalities*. New York: Marcel Dekker Inc; 1992.
- [5] Mallik, R.K. Solutions of linear difference equations with variable coefficients. *Journal of Mathematical Analysis and Applications*. 1998; 222:79–91. doi: 10.1006/jmaa.1997.5903.
- [6] Thiriet, M. and Parker, K.H. Physiology and pathology of the cardiovascular system: a physical perspective. In: Formaggia, L., Quarteroni, A. and Veneziani, A., editors. *Cardiovascular Mathematics*. Milano: Springer; 2009. p 1–45.
- [7] Latham, R.D., Westerhof, N., Giolma, J.P. and Altobelli, S.A. Regional wave travel and reflections along the human aorta: a study with six simultaneous micromanometric pressures. *Circulation*. 1985; 72:1257–1269. doi: 10.1161/01.CIR.72.6.1257.
- [8] Segers, P., De Backer, J., Devos, D. et al. Aortic reflection coefficients and their association with global indexes of wave reflection in healthy controls and patients with Marfan's syndrome. *American Journal of Physiology - Heart and Circulatory Physiology*. 2006; 290(6):H2385–H2392. doi: 10.1152/ajpheart.01207.2005.
- [9] Murgo, J.P., Westerhof, N., Giolma, J.P. et al. Aortic input impedance in normal man: Relationship to pressure wave forms. *Circulation*. 1980; 62: 105–116. doi: 10.1161/01.CIR.62.1.105.
- [10] Sherwin, S.J., Franke, V., Peiro, J. and Parker, K.H. One-dimensional modelling of a vascular network in space-time variables. *Journal of Engineering Mathematics*. 2003; 47:217–250. doi: 10.1023/B:ENGI.0000007979.32871.e2.
- [11] Formaggia, L., Lamponi, D. and Quarteroni, A. One-dimensional models for blood flow in arteries. *Journal of Engineering Mathematics*. 2003; 47:251–276. doi:10.1023/B:ENGI.0000007980.01347.29.
- [12] Wang, J.J. and Parker, K.H. Wave propagation in a model of the arterial circulation. *Journal of Biomechanics*. 2004; 37:457–470. doi: 10.1016/j.jbiomech.2003.09.007.
- [13] Weber, T., Auer, J., O'Rourke, M.F. et al. Arterial Stiffness, wave reflections, and the risk of coronary artery disease. *Circulation*. 2004; 109:184–189. doi: 10.1161/01.CIR.0000105767.94169.E3.
- [14] Stoner, L., Young, J.M. and Fryer, S. Assessments of arterial stiffness and endothelial function using pulse wave analysis. *International Journal of Vascular Medicine*. 2012; Article ID 903107, 9 pages. doi: 10.1155/2012/903107.
- [15] O'Rourke, M.F. and Taylor, M.G. Input Impedance of the System Circulation. *Circulation Research*. 1967; 20:365–380. doi: 10.1161/01.RES.20.4.365.
- [16] Sipkema, P., Westerhof, N. and Randall, O.S. The arterial system characterised in the time domain. *Cardiovascular Research*. 1980; 14:270–279. doi: 10.1093/cvr/14.5.270.
- [17] O'Rourke, M.F. The arterial pulse in health and disease. *American Heart Journal*. 1971; 82:687–702. doi:10.1016/0002-8703(71)90340-1.
- [18] Westerhof, N., Sipkema, P., van den Bos, G.C. and Elzinga, G. Forward and backward waves in the arterial system. *Cardiovascular Research*. 1972; 6:648–656. doi: 10.1093/cvr/6.6.648.
- [19] Parker, K.H. The reservoir-wave model. *Artery Research*. 2017; 18:87–101. doi:10.1016/j.artres.2017.04.003.

Development of a simulation model to analyze the performance of decentral rescheduling algorithms in production systems

Julian Sundermeier*, Felix Gehlhoff*, Alexander Fay

Helmut-Schmidt-University / University of the Federal Armed Forces Hamburg, Institute of Automation Technology, Holstenhofweg 85, 22043 Hamburg, Germany; *Sundermeier.julian@hsu-hh.de, Felix.gehlhoff@hsu-hh.de*

Abstract. Real production systems that rely on manual production and transportation processes are especially prone to disturbances. Thus, schedules often have to be revised on the spot by the operators using simple heuristics. These methods do not generate optimal solutions. Other approaches use agent-based control systems where the agents enable decentralized rescheduling based on local information. The following paper describes a simulation model of a production system that allows a quantitative comparison of the alternative control and planning logics by using the simulation software *FlexSim*[®].

Introduction

In today's competitive environment, manufacturing companies are confronted with the need of increasing flexibility [1]. This can be attributed to e.g. shortened product life cycles, increasing variety, and competitive pressure, which requires short and reliable delivery times [2]. Therefore, the production system must not only be able to produce efficiently, but moreover it needs to be resilient to disturbances and adaptable to changing demands. In addition, a modularized system architecture is advisable to enable system reconfigurations during ongoing production without having to adjust the control code.

Classic optimization or scheduling methods often struggle with these requirements [3]. Although these techniques can calculate an optimal solution, they require a considerable amount of time for complex problems. Furthermore, when using e.g. linear programming, it is neglected that significant and frequent changes to the production schedule create considerable noise and thus problems within the factory.

Therefore, in many real production systems, classic optimization processes are usually not suitable for dynamic rescheduling. Instead, there are mostly two approaches to deal with rescheduling. One is that the operators make the dynamic control decisions on their own. Often these decisions consider only a fraction of the

available information, e.g., what the operator assumes to be the most important job to fulfil next. Some companies employ software with simple algorithms which mimic such heuristics, e.g. the right shift (i.e. the postponement of all follow-up orders corresponding to the delayed order). This type of control results in production schedules that lag (far) behind an optimal solution, which could have been attained by using a more sophisticated algorithm.

In order to meet the stated requirements for flexibility and quality of the solution, several researchers propose agent-based control approaches. The agents enable decentralized decision-making and rescheduling based on local information. Usually, this interaction is much faster than classic optimization methods and allows a focused adaptation of the plan. However, the results achieved with decentral decision-making are usually not globally optimal.

The distribution of control intelligence and the information linked to it makes production systems more robust and represents a major goal in Industry 4.0 [4]. Despite an abundant amount of available literature on the development of multi-agent systems [1], there is still a lack of comparison – qualitatively and quantitatively – of agent-based control in case of disturbances with ‘traditional’ solutions [5]. Benchmark systems are one way to cope with this problem [6], but they mostly do not consider manual and thus less predictable processes. They also do not provide much help in terms of analysing the effects of rescheduling instead of just applying decentral control algorithms that take local decisions on the spot. Recent approaches, however, already give advice on how to measure qualitative factors such as robustness of a system that reacts to dynamic scenarios [7].

This paper describes a simulation model that enables the quantitative and qualitative comparison of different rescheduling algorithms – e.g. simulated operator deci-

sion-making, right-shift algorithms and agent-based dynamic rescheduling – to analyse the aforementioned effects. The context of the simulation model and the agent-system is outlined in Section 1. Section 2 gives a brief introduction into the simulation software that was used to build the model. The communication scheme between the simulation and the decision maker (e.g. an MAS (multi-agent system)) is explained in Section 3. This is followed by a description of the simulation model itself in Section 4. Section 5 briefly outlines the functionalities of the MAS and examines the interactions between simulation and the agents. The paper closes with a short validation of the developed simulation concepts in Section 6 and concludes in Section 7 with a brief summary and outlook.

1 Application context

The simulation model resembles a factory where heavy single workpieces have to be moved between subsequent production steps. The factory comprises different production stages in two hall bays. In at least on process step, alternative workstations allow parallel processing of workpieces. Manually operated cranes in each hall bay connect the workstations. A limited number of operator teams operates the cranes. It is one goal of the simulation to replicate the estimated durations of these transport operations, as they have a significant impact on the overall productivity of the factory. These estimated durations stem from the ERP system of the considered factory and look similar to the following example in Table 1.

Start	Destination	Time needed [min]
Workstation 1	Workstation 2	30
Workstation 2	Workstation 3	45
...		

Table 1: Excerpt of the ERP system transport data

These times include the time that is necessary to load and unload the workpieces but do not include the time needed to reach the workpiece, i.e. the time to move the crane from its previous position to the start position of this transport.

Shuttle cars connect the two hall bays. Two product variants are produced that are made of one main body and one additional part. These additional parts require different work- and transportation processes (e.g. only one crane for transportation instead of two for the main

body). One major problem is the coordination of the crane system. Due to various restrictions (e.g. the weight distribution on the ceiling construction), this task becomes particularly challenging.

If disturbances occur during production, today’s human endeavours mainly constitute the right-shift after a delayed order and on the spot decisions by e.g. crane operators. This has to be replicated within the simulation model. Furthermore, it is necessary that the simulation is able to follow a specified production schedule, which is stored in a database.

In addition, there must be the possibility that an agent-based decentralized planning algorithm (or any other algorithm) can dynamically adjust or replace the production schedule when disturbances occur and thus control the simulation. If the simulation is controlled by an external planning algorithm it strictly follows the production schedule that is provided by the algorithm via the database. It does not execute any planning or rescheduling functions itself (e.g. right-shift).

2 FlexSim® simulation software

FlexSim® is an object oriented, discrete-event simulation software distributed by FlexSim Software Products Inc. (Orem, Utah, USA). Also declared as application-oriented simulation package [8], it is especially common for the simulation of production and intra-logistic processes. *FlexSim*® offers a GUI with a 2D or 3D view of the model and several standard library objects that can be included into the model via drag & drop.

Generally, *FlexSim*® can process numerical and string values which can be stored in global variables or tables or locally in labels on an object. For further information on the capabilities of *FlexSim*® see [9].

FlexSim® has been chosen due to two advantages compared to other simulation software [10]: on the one hand, all library objects can be edited and complex logic can be integrated into their event-triggers by using *FlexScript*, an embedded programming language. Logic can also be added by use of C++, but this code needs to be compiled before the model can be run. On the other hand, *FlexSim*® provides a standard interface to a MySQL database, which can be used for the necessary access to the production schedule. The use of a standard interface facilitates the data exchange, as no new interface has to be configured.

3 Connecting to the simulation via MySQL

To ensure the required data exchange, the communication scheme depicted in Figure 1 was used.

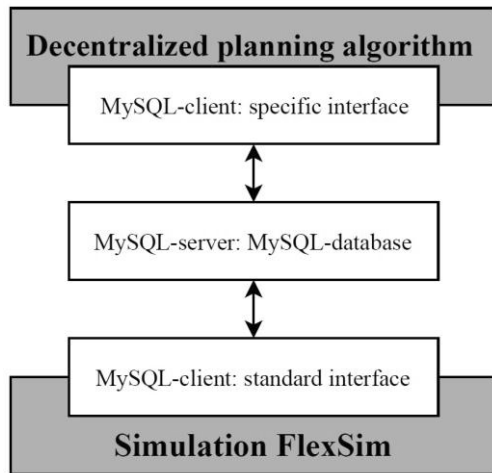


Figure 1: Communication scheme between simulation and agent-system

Both, the decentralized planning algorithm and the simulation model in *FlexSim*® act as MySQL-clients via appropriate interfaces to a MySQL-server with the MySQL-database. The database itself contains the three tables ‘productionplan’, ‘resources’ and ‘monitoring’. In the table ‘productionplan’ (see Table 3 in the appendix), all orders are listed according to their unique value of the label ‘ID’, which is used inside the simulation model. Starting at column two, every seventh column contains the title of a production step. The value of the element that is at the intersection of a row and such a column indicates the assignment of a workstation for an explicit order. The columns in between are used for data collection of the target start/finish and actual start/finish date of a process and whether the processing of an order has already been started or completed.

In the table ‘resources’ (see Table 4 in the appendix), all workstations are listed according to their unique value of the label ‘Object_ID’ which is used in the simulation model. In addition, the table provides information on the respective operating status of a workstation and contains information on when and where a disturbance has occurred.

For the decentralized planning algorithm only those entries are relevant which concern the occurrence of a

disturbance. For this reason, a third table called “monitoring” has been created, consisting of only one element. This element serves as a binary indicator of whether rescheduling is required or not. Thus, the decentralized planning algorithm only has to monitor this value and only become active when there is a change. This value also serves as an auxiliary value for the algorithm to detect whether it has already responded to a disturbance or not.

The database that serves as an interface between the simulation and a planning algorithm is integrated within the process depicted in Figure 2.

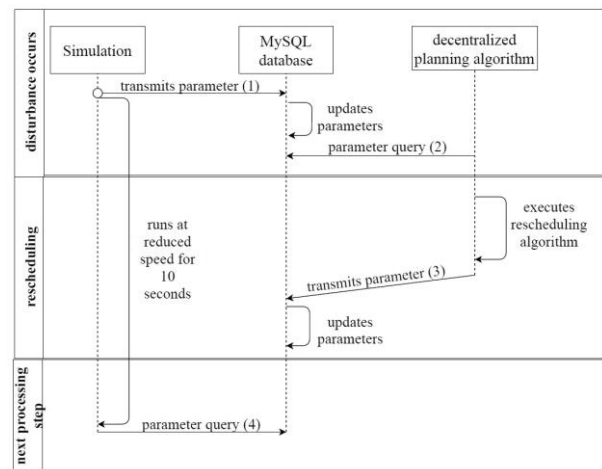


Figure 2: Sequence diagram of disturbance handling between simulation and MAS

This process shows how the simulation and the decentralized planning algorithm interact in case of disturbances. Section 5.2 explains the details of this interaction. Due to the intended application of the decentralized planning algorithm within a real factory (see section 5.1) there should be no more than ten seconds between the occurrence of a disturbance and further processing after the rescheduling algorithm has been executed.

For the execution of the rescheduling algorithm, the decentralized planning algorithm might need up to nine of the ten seconds available. In the remaining second, there are four accesses to the database (see numbers one to four in Figure 2). First, a disturbance occurs in the simulation and the simulation sets corresponding parameters in the database. Second, the decentralized planning algorithm reads these parameters and executes the rescheduling algorithm. Third, after the execution is completed, the production schedule is updated in the database. Finally,

the simulation queries the database to receive the updated production schedule and runs the model accordingly.

To validate if this concept is feasible it should be determined whether the access time to the database can fulfil a real-time criterion of a maximum of 250 ms (four accesses in one second). Within that access time, the connection to the database must be established and all query parameters exchanged. To analyse the access times a proof of concept model has been created (see Figure 3).

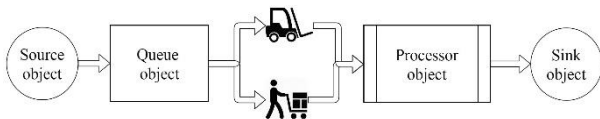


Figure 3: Structure of the proof of concept model

A source object creates ‘flowitems’ that are buffered in a queue object. The transport from the queue object to the processor object can be done either by an operator or by a transporter. The respective assignment is stored in a MySQL database. After processing on the processor object, the ‘flowitems’ leave the system via a sink object. If either the operator or the transporter fails during a simulation run, an agent system is activated which adjusts the assignments in the database to transport all ‘flowitems’ from the remaining resource.

In the proof of concept model, parameters from the database were queried and / or manipulated by firing individual event triggers. To determine the required access time for such an operation, the *FlexSim*[®] internal *FlexScript Code Profiler* tool was used (see Figure 4)

Self Time (ms)	Total Time (ms)	Hit Count	Path
10674.21	10683.79	1192	/Monitoring_3>variables/processfinishtrigger
29.65	29.65	4	/Tools/MTBFMTTR/G1>variables/downtrigger
26.01	26.01	3	/Tools/MTBFMTTR/G1>variables/downtrigger
24.49	24.49	3	/Tools/MTBFMTTR/G1>variables/uptrigger
17.16	17.16	5	/Queue10>variables/transportdispatcher

Figure 4: Excerpt from the *FlexScript Code Profiler* tool

Lines two through four include function calls that both query and manipulate parameters from the database. In the simulated time, the functions were called four times (see second row, third column) or three times (see third and fourth row, third column). Overall, the function calls in total according to the second column have required between 29.65 ms and 24.49 ms. This results in an average access time of 5.666 ms per function call for

reading and writing a parameter into the database. The fifth row contains a trigger in which only one value is entered (not read) into the database. This results in an access time of 3.432 ms per function call. In summary, access times are short enough to guarantee the real-time capability of 250 ms.

4 Simulation model

This section first discusses the conception of the simulation model. For this purpose, the basic requirements and the particular restrictions are emphasized. Subsequently, the respective implementation will be briefly discussed.

4.1 Conception

A basic requirement for the control of the simulation model is that the orders run through the system according to the given production schedule. It must be ensured that the sequence of the production processes is adhered to and the actual start and finish dates are entered into the database.

Beside the basic material flow, particular restrictions must be respected during implementation. This includes e.g. compliance with follow-up constraints arising from production requirements.

The crane system is a key component of the production system, because it is indispensable for the material flow. In this context, a transport can be understood as a process characterized by a movement to the pick-up location and a transport time. These times correspond to the set-up and processing time in a production process. For this reason, a crane in the simulation model can also be represented by the standard object 'processor'. Although this abstraction restricts the visualization of the material flow in the model, it also offers an advantage. Namely, the routing of the ‘flowitems’ can be more easily controlled because the behavior of the input and output ports of a processor object can be specified via appropriate event triggers. Both, the time for the movement to the pick-up location as well as the transport time are stored in the database. If the production schedule was created by a decentralized planning algorithm, the time required for the movement to the pick-up location is eliminated since this movement is anticipated in advance.

In order to avoid coordination problems due to overlapping work areas of the cranes in a hall bay, capability profiles are created which define the executable transport

processes of a crane. As these profiles are mutually exclusive, there are only very few possible collision points. These can be disregarded due to the very low probability and the capability of human operators in the real application to avoid the actual collision (e.g. by waiting until the other crane has been removed).

A last challenging problem occurs from the restriction that *FlexSim*[®] does not provide an option within its standard commands to cancel the processing of a ‘flowitem’ on a processor object. However, this is exactly what should happen when a material defect is detected so that the ‘flowitem’ can be transported to another workstation for a post-processing.

4.2 Implementation

To facilitate later adaptation or reconfiguration of the model, only standard library objects have been used for the implementation.

The routing of the ‘flowitems’ and the associated adherence to the process sequence is ensured by the two labels ‘ID’ and ‘NextStep’, which are stored on each ‘flowitem’. While the value of the label ‘ID’ is constant, the value of the label ‘NextStep’ is adapted dynamically after the successful completion of a production step. This is done via the ‘OnProcessFinish’ trigger of each processor object that represents a workstation in the simulation model. The value of the label ‘NextStep’ starts at 2, so it points to the second column of the table ‘production-plan’, where the assignment of the workstation for production step 1 takes place (compare Table 3). After the production step finished, the value is increased by the value 7 so it points to the ninth column etc.

For the modeling of the cranes, a pull system was implemented, which ensures that a ‘flowitem’ is not transported until its processing is completed and the following workstation is idle. If both conditions are fulfilled, the ‘flowitem’ gets pulled and the value of the label ‘NextStep’ is read out. Thereafter, the ‘flowitem’ is processed (this represents the ongoing transport) and finally send to the target workstation. The processing times correspond to the estimated times from the ERP and are chosen according to the destination. Each crane is connected to other workstations through its output ports, which reflects the implementation of its capability profile.

The actual start and finish date of a production step is set by the ‘OnEntry’ and ‘OnExit’ trigger of a processor object which represents a workstation.

Compliance with the follow-up constraints can easily

be achieved due to the routing concept used.

In order to simulate the detection of a material defect, including all subsequent repair processes, a slightly more cumbersome solution was implemented. For this, a queue object was created for each workstation and then connected via the center port. After that, a separate mean time between failure / mean time to repair (MTBF / MTTR) object was created for each workstation, which is not intended to control the failure behavior of a workstation, but instead simulates the detection of a material defect. For this reason, the MTTR has been set to a very small value. The ‘OnBreakdown’ trigger contains the functions that cause the canceling of the current production step.

The ‘flowitem’, which is located on the workstation whose ‘OnBreakdown’ trigger was fired, is duplicated and the duplication is send to the connected queue object. Then the original ‘flowitem’ is destroyed, which represents the cancellation of the processing. When executing this command while processing a ‘flowitem’ on a processor object, the *FlexSim*[®] engine can no longer execute the other scheduled events of the processor object (e.g. completion of processing). For this reason, another command must destroy all scheduled events. At the same time, this command causes the affected processor object to never receive ‘flowitems’ again, even if it is intended in the production plan. However, this can be removed by executing a command that resets the properties of the processor object.

After all commands of the ‘OnBreakdown’ trigger have been executed, post-processing of the duplicated ‘flowitem’ must be scheduled by the decentralized planning algorithm. This process – among others – is explained in the next section.

5 Interaction between simulation and decentralized planning algorithm

This section examines the interactions that take place between the decentralized planning algorithm (henceforth, an MAS is assumed to fulfil this role) and the simulation in case of disturbances as well as the functionalities of the MAS itself. The detailed design of the MAS is not a part of this paper, however, it is briefly outlined which functionalities are provided by the MAS.

5.1 Functionalities of the MAS

Starting point for the interaction between the MAS and the simulation is the creation of an initial schedule. The MAS in place is designed similar to approaches like [11] or [12]. Workpiece agents are responsible for the fulfilment of all necessary production steps for the workpiece which they represent. Requirements regarding the number and type of workpieces to be produced as well as the production plan for each type of workpiece are stored in the database. A production plan looks similar to the following example.

ID	Product Name	Step	Operation (Op)	First Operation	Last Operation
1	A	1	Op_1.1	1	0
1	A	2	Op_1.2	0	0
...					

Table 2: Exemplary excerpt of a production plan

All information regarding the resources is also stored in the database. The agent system uses the same table for this information as the simulation. Each resource has a certain capability. These capabilities are mapped to operations that can be accomplished by the resource.

This design, which uses a database to store all necessary information for products and resources, makes the MAS easily adjustable to changing needs and production technologies as there are no changes necessary within the code of the agents itself. When all agents are created, which happens automatically based on the information in the database, the workpieces use an auction-based process based on the Contract Net [13], which is the most common coordination mechanism for MAS [14]. As the MAS is implemented in JADE [15], the workpiece agents can search for resources that provide certain operations by using the so-called Directory Facilitator of the JADE platform.

The main goal of the agent system is to reschedule in case of disturbances. Thus, the scheduling process is not focussed on creating the optimal solution for the factory for all orders at the same time but instead schedules one job after another and tries to keep as close to already scheduled operations as possible. This results in a completely decentralized approach in which no supervisory component or agent knows all information and each agent manages its own schedule.

Each workpiece schedule is created including the transport processes, as these might be a bottleneck within the production system. To achieve this, the operations at workstations (WS) are scheduled as depicted in Figure 5, similar to [12].

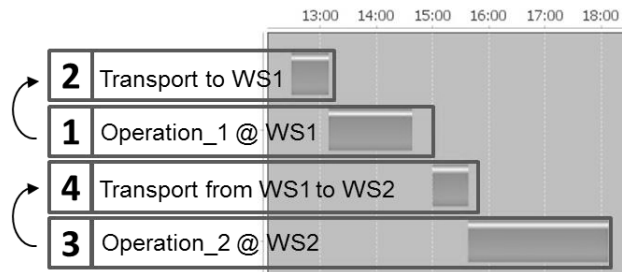


Figure 5: Exemplary scheduling in a regular case

The workpiece agent uses an estimation of the transport time as the earliest possible start date for the initial production operation request or CFP (call for proposal). After the transport operations have been scheduled, the agent sends messages to the production resource agents that inform them about the actual arrival and departure times. In case the transport process is the bottleneck, the production step is cancelled and rebooked at the earliest possible arrival time. This time is derived from the best proposal of all transport resources. If a disturbance has happened that demands a deviation from the initial production plan – here a repair process at another workstation – the scheduling process follows a similar approach. The difference is that not only the production operation has to be scheduled but also the buffering and repair process. The production operation has to be scheduled first because the necessary duration, i.e. the end of the buffering process, depends on the earliest start of the production process.

5.2 Interaction in case of disturbances

Different situations can trigger a rescheduling process by the MAS. These vary from equipment breakdowns to process delays. The focus of this paper is on defects that are detected in the workpiece itself. When combining the MAS with the simulation, the simulation has to set the triggers for the MAS in the database that will start a rescheduling process (as shown in Figure 2).

The information provided include the error type and occurrence time in the resource table at the applicable resource. In addition, the parameter in the table ‘monitoring’ is set to true. The simulation also goes into a mode

of slower simulation speed. This is necessary because the *FlexSim*[®] engine does not offer an opportunity to completely stop a simulation run for a certain period of time. The dynamic adaptation of the simulation speed uses the object orientation of *FlexSim*[®]. Thus, the ‘OnBreakdown’ trigger, which is fired anyway in the detection of a material defect, spawns a new ‘flowitem’ in a specially created Queue object. This ‘flowitem’ fires the ‘OnEntry’ trigger of the Queue object, which reduces the simulation speed to one hundredth of the original simulation speed. In addition, the ‘OnEntry’ trigger sends a message to a second Queue object with a delay of 10 real-time seconds. Upon receiving the message, the second Queue object increases the simulation speed to its original value. This procedure does not prevent the case that, during the time when the simulation is running at a reduced speed, another event occurs which relies on information from the database. However, this case is considered very unlikely and thus acceptable.

The MAS monitors a parameter that triggers its activities. If this parameter is set, the agent detects this change and reads all necessary disturbance information from the database. Afterwards the parameter in table ‘monitoring’ is set back to false. The information at which resource and what time the disturbance occurred enables the agent to conclude which workpiece agent is involved and needs to be contacted. The disturbance information is sent to the corresponding agent. The workpiece agent in turn updates its own schedule with the process finished values from the database that the simulation has set for each operation. The workpiece agent then determines which actions are needed for the disturbance that occurred. In this case, all operations that are not finished yet have to be cancelled and the already explained error handling with booking of a buffer place is started.

After the new schedule is complete, the agent sends this schedule to an agent that has the capability to insert this schedule correctly into the database (DB Connector). A so-called Sniffer Agent that comes with the JADE platform can automatically monitor and visualize the exchange of messages. An abbreviated (as can be seen from the number of messages on the left) and commented example is shown in Figure 6.

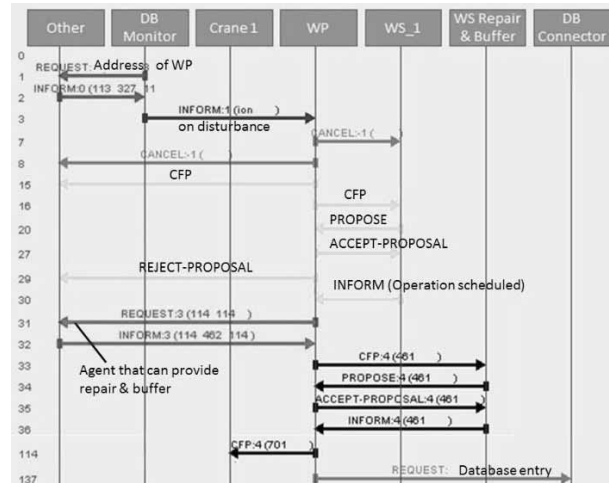


Figure 6: Sequence diagram of disturbance handling within the MAS

The simulation now uses these actualized values to determine the correct destinations of the objects within the simulation.

6 Validation and analysis

It is necessary to validate whether the simulation model does correctly implement the production schedule that the agent system has provided and show the effects of default solutions in case of incorrect schedule data. In addition, it must be analyzed whether the production schedule is correctly implemented after a rescheduling by the agent system.

In order to validate that a provided production schedule is implemented correctly, two labels were stored on each flowitem. The production schedule as provided in advance is stored on one label. On the other label, the actual process sequence that a ‘flowitem’ passes through is plotted during a simulation run. When a ‘flowitem’ reaches the sink and both labels match, the production schedule was implemented correctly. This result was achieved in the evaluation of several simulation runs. If the rescheduling algorithm was executed during a simulation run, probably the labels no longer match. In this case, it is currently necessary to manually check whether the updated production schedule has been adhered to.

On the other hand, it could be shown that the simulation model cannot implement incorrect production schedules. If the simulation implements such a production schedule, the material flow stagnates and the simulation run is aborted.

The provided target times could be adhered to with minor deviations. This results from the comparison of the corresponding values in the database with the actual times entered by the simulation.

7 Conclusion and Outlook

This paper presents an approach how control algorithms, such as a decentralized planning algorithm, can be tested

in combination with a simulation model in *FlexSim*[®], linked via a production schedule in a database. Since it is possible to link arbitrary algorithms to the database, the simulation model can be used to compare algorithms which implement different production schedules. Possible metrics that can be examined are e.g. the throughput time of the orders, the makespan or the impact of the occurrence of a disturbance.

8 Appendix

ID	Step1	Target start time 1	Target finish time 1	Processing started 1	Actual start time 1	Actual finish time 1	Processing finished 1	Step 2	...
1	Value of the label "Object_ID" indicates on which workstation from the table "resources" the order should be processed	Elapsed simulation time at which process step 1 should start	Elapsed simulation time at which the processing of process step 1 has actually started	= 1, if processing of process step 1 has started; = 0, otherwise	Elapsed simulation time at which process step 1 should end	Elapsed simulation time at which the processing of process step 1 has actually ended	= 1, if processing of process step 1 has ended; = 0, otherwise	Value of the label "Object_ID" indicates on which workstation from the table "resources" the order should be processed	...
...

Table 3: Schematic structure of the table 'productionplan'

Object ID	designation	WIP/BU	On/Off	Error Type	Error Occure Time
1	Designation of the workstation where the label 'Object_ID' has the value 1 in the simulation	= 1 if an order is being processed in the workstation / buffered in a buffer; = 0, otherwise	= 1 if the workstation is functional; = 0, if there is a disturbance and the workstation is not available	= 1 if a defect in the workpiece itself has been detected; = ..., is not considered in this paper	Past simulation time at which the disturbance occurred
...

Table 4: Schematic structure of the table 'resources'

References

- [1] Leitão, P.: Agent-based distributed manufacturing control. A state-of-the-art survey. In: Engineering Applications of Artificial Intelligence 22 (7), pp. 979–991.
- [2] Wiendahl, H.-P.; ElMaraghy, H. A.; Nyhuis, P.; Zäh, M. F.; Wiendahl, H.-H.; Duffie, N.; Brieke, M. (2007): Changeable Manufacturing - Classification, Design and Operation. In: CIRP Annals 56 (2), pp. 783–809.
- [3] Paolucci, M.; Sacile, R. (2005): Agent-based manufacturing and control systems. New agile manufacturing solutions for achieving peak performance. Boca Raton Fla.: CRC Press
- [4] Spath, D. (Hrsg.), Ganschar, O., Gerlach, S., Hämmerle, M., Krause, T., Schlund, S.: Studie des Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO: Produktionsarbeit der Zukunft – Industrie 4.0, Stuttgart 2013.
- [5] Wior, I.; Jerez, S.; Fay, A.: Automated transportation systems subject to interruptions in production and intralogistics - a survey and evaluation. In: International Journal of Logistics Systems and Management (IJLSM), Vol. 30, No. 4, 2018. DOI: <http://dx.doi.org/10.1504/IJLSM.2018.10011675>
- [6] Schreiber, S.; Fay, A.: ARGESIM Benchmark C20 'Complex Production System' – Definition and Call. In: SNE Simulation Notes Europe, vol. 21 (3-4), 2011.
- [7] Trentesaux, D.; Pach, C.; Bekrar, A. ; Sallez, Y. ; Berger, T.; Bonte, T.; Leitão, P.; Barbosa, J. (2013): Benchmarking flexible job-shop scheduling and control systems. In:

- Control Engineering Practice 21 (9), pp. 1204–1225.
- [8] Law, A M.: Simulation Modelling & Analysis. New York: McGraw-Hill, 2007, pp. 182-185
- [9] Beaverstock M., Greenwood A., Lavery E., Nordgren W.: Applied Simulation: Modelling and Analysis Using FlexSim. Orem, Utah, USA: Flexsim Simulation Software, 2011.
- [10] Swain, J. J. Simulation software survey: Simulated worlds. In: OR/MS Today, Vol. 42, No. 5, October 2015, pp. 36-49.
- [11] Heinze, M.; Lüder, A.; Gantner, W.; Kühnle, H.; Peschke, J. (2008): Structure and Functionality of a PABADIS PROMISE Agent System. In: Klaus-Dieter Thoben (Hg.): ICE2008. The 14th International Conference on Concurrent Enterprising, a new wave of innovation in collaborative networks, Lisbon, Portugal, 23 - 25 June 2008.
- [12] Badr, I.; Schmitt, F.; Göhner, P. (2010): Integrating Transportation Scheduling with Production Scheduling for FMS: An Agent-Based Approach. In: IEEE International Symposium on Industrial Electronics (ISIE), 2010.
- [13] FIPA Contract Net Interaction Protocol Specification, Geneva 2002. URL: www.fipa.org/specs/fipa00029/SC00029H.pdf (Stand 26.07.2018).
- [14] Caridi, M.; Cavalieri, S. (2007): Multi-agent systems in production planning and control. An overview. In: Production Planning & Control 15 (2), S. 106–118.
- [15] Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE, Chichester 2007.

Simulationsbasierte Optimierung des Umgangs mit Chef-Aufträgen im Produktionsbetrieb

Max Melter¹, Jens Heger^{1*}, Carsten Wagner¹

¹Institut für Produkt- und Prozessinnovation, Leuphana Universität Lüneburg, Universitätsallee 1, 21335 Lüneburg, Germany; *jens.heger@leuphana.de

Abstract. Im Produktionsablauf werden Unternehmen in Engpasssituationen vor die Frage einer Bevorzugung bestimmter Aufträge gestellt. Vor diesem Hintergrund wird in dieser Arbeit diskutiert, wie ein Umgang mit bevorzugten Chef-Aufträgen, in Abwägung mit der Termintreue normaler Aufträge, gelingen kann. Dafür wird erläutert und diskutiert, inwieweit die Fertigungssteuerung lieferzeitvariabel und engpassorientiert gestaltet werden kann. Abgeleitet aus theoretischen und praktischen Herangehensweisen werden drei Ansätze entwickelt und simulativ verglichen. Dabei zeigt sich, dass eine absolute Priorisierung von Eilaufträgen zwar eine Bevorzugung sicherstellt, die Termintreue von normalen Aufträgen aber deutlich abnimmt. Dem entwickelten Ansatz der variablen Plan-DLZ gelingt dies situationsabhängig besser. In bestimmten Unterversuchen zeigt sich eine höhere Termintreue bei Eilaufträgen als auch bei normalen Aufträgen.

Einleitung

Ein häufig auftretendes Problem industrieller Lieferketten ist, dass produzierende Unternehmen über gut ausgelastete Ressourcen verfügen und keine ausreichende Flexibilität aufweisen. Die Erweiterung der Ressourcen bei steigender Nachfrage ist aufgrund der erforderlichen Reaktionszeit oder hoher Investitionsanforderungen oft keine verfügbare Handlungsoption. Die Aufträge der Kunden konkurrieren folglich um die vorhandenen Ressourcen, sodass viele Unternehmen Probleme haben, die logistischen Anforderungen ihrer Kunden an Lieferzeiten und Termintreue zu erfüllen. Dabei sind sowohl eine hohe Termintreue als auch geringe Durchlaufzeiten ein gewichtiges Verkaufsargument gegenüber dem Kunden. Folglich müssen Unternehmen entscheiden, ob sie einzelne profitable Aufträge anderen gegenüber bevorzugen, wie eine Bevorzugung innerhalb des Produktionsablaufs gelingen und wie gleichzeitig eine hohe Termin-

treue bei normalen, nicht priorisierten Aufträgen gehalten werden kann. Dazu werden im folgenden Artikel verschiedene Ansätze untersucht und aufgezeigt, in welchen Situationen diese effektiv eingesetzt werden können.

1 Motivation, Problemstellung

Die in dieser Studie betrachtete Problemstellung ist an den Produktionsprozess eines mittelständischen Unternehmens angelehnt, kann aber auch auf andere produzierende Unternehmen übertragen werden und behandelt den Umgang mit sogenannten „Chef-Aufträgen“. Darunter werden spontan auftretende ungeplante Aufträge verstanden, die gegenüber normalen Aufträgen priorisiert behandelt werden sollen. Die Chef-Aufträge konkurrieren folglich mit bereits eingeplanten Aufträgen um die vorhandenen Ressourcen. Im Fall ausgelasteter Ressourcen entstehen Engpässe, sodass das Unternehmen Probleme hat, die logistischen Anforderungen seiner Kunden hinsichtlich Lieferzeiten und Termintreue zu erfüllen. Darüber hinaus wird in der Arbeit die Frage gestellt, inwieweit das Auftreten solcher Aufträge durch zu erwartende Mehreinnahmen positiv zu betrachten ist und wie eine Fertigungssteuerung im Vorhinein auf die Bearbeitung dieser Aufträge eingestellt werden kann. In der Praxis ist die Fertigungssteuerung derzeit nicht auf die effektive Einplanung von spontan auftretenden Chef-Aufträgen ausgerichtet und entsprechende negative Folgen für die Termineinhaltung werden hingenommen. Daher besteht das Ziel darin, die Fertigungssteuerung dahingehend anzupassen, dass Chef-Aufträge zumindest in einem bestimmten Umfang am Produktionsprogramm ohne Terminkonsequenzen für bereits eingeplante Aufträge angenommen und bearbeitet werden können. Dies kann in einem Wettbewerbsvorteil für Unternehmen in unsicherem Umfeld resultieren.

2 Stand der Technik

Die Auswahl des richtigen Fertigungssteuerungsverfahrens für die jeweiligen unternehmensspezifischen Anforderungen wird schon seit langer Zeit aufgrund seiner hohen Relevanz in der Praxis diskutiert. Lödding unterteilt die Fertigungssteuerung in die Auftragsproduktion, die Auftragsfreigabe, die Kapazitätssteuerung und die Reihenfolgebildung und gibt eine Übersicht über verschiedene Methoden für die jeweiligen Abschnitte [1]. Je nach praktischem Anwendungsfeld erzielt die Kombination verschiedener Verfahren gute Ergebnisse. Eine Priorisierung einzelner Aufträge erfolgt im Produktionsablauf in der Reihenfolgebildung. Ansätze dafür lassen sich in optimale Verfahren und in Heuristiken unterteilen. Heuristiken werden je nach Planungshorizont in zentrale und dezentrale Heuristiken eingeteilt [2]; Jain und Meeran [3] geben eine Übersicht über verschiedenen Verfahren.

Die Behandlung von Chef-Aufträgen in der Fertigungssteuerung wird hingegen kaum diskutiert und Methoden zum Umgang mit diesen sind zu ergründen. Unter dem Stichwort „Eilaufträge“ gehen Günther und Tempelmeier [4], sowie Drexl und Haase [5] auf die Notwendigkeit ein, diese in die Fertigungssteuerung mit einzubeziehen. Letztere entwerfen als ersten Ansatz die Idee, gewisse Kapazitäten zum Abarbeiten dieser Aufträge freizuhalten. Tyrzna [6] diskutiert die Modellierung von Durchlaufzeiten für priorisierte und für normale Aufträge und einer Neugestaltung der Auftragsannahme und -terminierung innerhalb der Fertigungssteuerung. Es zeigt sich aber, dass praxisorientierte Ansätze in der Literatur bisher nur ansatzweise beschrieben werden.

3 Versuchsaufbau

Zur Diskussion der Anpassungen der Fertigungssteuerung wird ein Produktionsablauf, angelehnt an das betrachtete mittelständische Unternehmen, simuliert und die Auswirkungen einer Priorisierung auf diesen untersucht. Dieser besteht aus einer Fertigung, einer Vormontage, einer Montage, einer Qualitätskontrolle und einer Kommissionierung (siehe Abbildung 1).

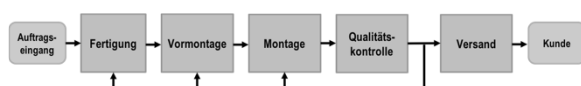


Abbildung 1: Der Produktionsprozess als Simulationsmodell

Wenn bei einem Auftrag ein Fehler festgestellt wird, der jeweils mit einer Eintrittswahrscheinlichkeit von 0,05 (Fertigung und Vormontage) und 0,01 (Montage) angenommen wird, muss dieser in die entsprechende Vorstufe zurückgeleitet und erneut bearbeitet werden. Die Auftragsgänge werden nach Eley [7] mithilfe einer Poisson-Verteilung modelliert. Zur weiteren Untersuchung werden Annahmen zur Darstellung eines praxisnahen Modells getroffen:

- Für die Datenerhebung werden 5.000 Aufträge betrachtet. Für einen standardisierten Versuchsaufbau findet eine Ein- und Ausschwingphase statt. Dadurch wird sichergestellt, dass das System während der Untersuchung ausbalanciert ist [8].
- Je nach Versuch wird ein Auftragsgang von 40 oder 45 Aufträgen je Monat angenommen. Dadurch können die Auswirkungen der Variationen unter verschiedenen Auslastungen untersucht werden.
- Von diesen Aufträgen werden zehn Prozent der Aufträge als Eilaufträge angenommen.
- Die Bearbeitungszeit für die einzelnen Aufträge beträgt 2,8 Tage. Bevorzugte und normale Aufträge benötigen die gleiche Bearbeitungszeit. Es fällt keine Transportzeit zwischen den Arbeitsstationen an und Rohstoffe sind immer lieferbar [9].
- Die Plan-Durchlaufzeit beträgt für normale Aufträge neunmal die Bearbeitungszeit, für Chef-Aufträge, um ein markattraktives Angebot zu schaffen, dreimal die Bearbeitungszeit. Aufträge, die nach der Plan-Durchlaufzeit fertiggestellt werden, sind als verspätet anzunehmen.

Basierend auf diesen Annahmen werden im Folgenden drei Untersuchungsansätze erläutert und die Auswirkungen diskutiert. Beim ersten Ansatz wird gar keine Priorisierung vorgenommen, sodass alle Aufträge gleich wichtig sind, beim zweiten Ansatz werden Chef-Aufträge an jeder Station des Produktionsablaufs immer bevorzugt. Beim dritten Ansatz erfolgt eine Priorisierung anhand des zugesagten Fertigstellungstermins, der bei Chef-Aufträgen kürzer als bei Normal-Aufträgen angenommen wird. Das betrachtete Zielkriterium ist die Anzahl der termingerechten Aufträge.

4 Untersuchungsansätze

4.1 Steuerung nach FIFO

Im ersten Versuch werden die Auswirkungen einer klassischen FIFO-Steuerung untersucht. Dafür wird das im vorherigen Abschnitt entwickelte Simulationsmodell genutzt und eine FIFO-Priorisierung als Steuerungsverfahren in den Warteschlangen der einzelnen Arbeitssysteme verwendet. Da das Steuerungsverfahren keine Priorisierung von Chef-Aufträgen vornimmt, ist ein hoher Verspätungsanteil bei diesen zu erwarten.

4.2 Steuerung mit maximaler Priorisierung

Um eine Priorisierung der Chef-Aufträge im Simulationsmodell zu gewährleisten, wird im zweiten Versuch eine maximale Priorisierung der Chef-Aufträge innerhalb der Warteschlangen der einzelnen Arbeitssysteme angenommen. Dafür wird der Anteil der Chef-Aufträge in der Quelle markiert und entsprechend in den einzelnen Produktionsschritten normalen Aufträgen gegenüber prioritär behandelt. Deswegen wird in den Warteschlangen der einzelnen Arbeitssysteme die Prioritätsregelung vom FIFO-Verfahren auf eine auf der Priorität basierende Regel umgestellt.

4.3 Einführung einer variablen Plan-Durchlaufzeit

Voraussichtlich ist es für die Termintreue nachteilig, wenn bestimmte Aufträge entsprechend ihrer Markierung prioritär behandelt werden, dabei aber normale Aufträge in der Warteschlange verbleiben. Daraus resultiert ein längeres Verweilen der normalen Aufträge in der Warteschlange, was unweigerlich zu einer stark verspäteten Fertigstellung führt. Sinnvoll wäre daher eine Abwägung innerhalb der Warteschlange, ob die Priorität des Auftrags oder die aktuelle Wartezeit wichtiger zu bewerten ist.

Dies gelingt durch eine Anpassung der Plan-Durchlaufzeiten in Kombination mit einer Änderung der Reihenfolgeregelung. In den Warteschlangen wird anstelle des FIFO-Verfahrens ein Priorisierungsverfahren gewählt, das denjenigen Auftrag mit dem nächsten Fertigstellungstermin freigibt. Ein geeignetes Verfahren hierfür ist das „*earliest due-date*“ Verfahren [10]. Es sorgt in der Anwendung mit einer großen Anzahl an Aufträgen

mit ähnlichen Bearbeitungszeiten für eine hohe Termintreue. Allerdings muss das Verfahren noch ergänzt werden. Ansonsten würde dieses Verfahren zwar zu einer termingerechten Bearbeitung der Aufträge führen, Chef-Aufträge aber nicht bevorzugen. Entsprechend muss das terminierte Datum der Fertigstellung, gesteuert über die Plan-Durchlaufzeiten, angepasst werden. Innerhalb der Simulationsstudie wird als Plan-Durchlaufzeit äquivalent zu der im dritten Abschnitt definierten Termintreue-Regel mit dreimal der Bearbeitungszeit bei Chef-Aufträgen und neunmal der Bearbeitungszeit bei normalen Aufträgen gearbeitet.

5 Simulationsstudie

Für die Simulationsstudie werden die drei Ansätze, wie im Abschnitt „Versuchsaufbau“ beschrieben, mit unterschiedlichen Parametern durchgeführt. Die Ergebnisse sind in den nachfolgenden Tabellen dargestellt. Dabei wird nach der Anzahl und Art der Aufträge und dem simulierten Ansatz unterschieden.

Ansatz	Auslastung am Engpass	normale Aufträge	Chef-Aufträge	Gesamt
1. Ansatz FIFO	83 %	0,0	11,2	1,2
	95 %	3,3	53,6	8,1
2. Ansatz max Prio	83 %	0,0	0,2	0,1
	95 %	4,2	0,8	3,9
3. Ansatz variable Plan-DLZ	83 %	0,0	0,2	0,1
	95 %	1,3	2,3	1,4

Tabelle 1: Darstellung der Anteil verspäteter Aufträge nach Auftragsart in Prozent

In den Versuchen werden Chef-Aufträge mit einer Fertigstellung innerhalb von drei Bearbeitungszeiten, normale Aufträge hingegen mit einer Fertigstellung innerhalb von neun Bearbeitungszeiten als pünktlich angenommen ($DueDate = \text{aktuelle Zeit} + 3 \text{ bzw. } 9 * \text{Bearbeitungszeit}$). Bei dieser Betrachtung zeigt sich, dass eine Priorisierung der Chef-Aufträge innerhalb der Warteschlangen erfolgen muss. Nur so lässt sich die Termintreue dieser bevorzugten Aufträge, abgeleitet aus dem Anteil verspäteter Aufträge, niedrig halten. Darüber hinaus lässt sich die anfangs aufgestellte These nachweisen, dass eine maximale Priorisierung von Chef-Aufträgen

zwar für eine hohe Termintreue dieser sorgt, gleichzeitig aber die Termintreue der normalen Aufträge sinkt. Vor allem bei einer höheren Auslastung gelingt dies bei der Annahme einer variablen Plan-DLZ in Kombination mit einer EDD-Steuerung besser. Dies geschieht allerdings im Kompromiss zu einer leicht geringeren Termintreue der Chef-Aufträge. Bei der Entscheidung für ein Steuerungsverfahren, das die Bevorzugung von Aufträgen berücksichtigt, muss in der Praxis entsprechend eine Abwägung der Wichtigkeit dieser im Vergleich zu normalen Aufträgen erfolgen (siehe Tabelle 1). Bei der Betrachtung der durchschnittlichen Durchlaufzeit zeigt sich ebenfalls, dass wie zu erwarten, eine Priorisierung innerhalb der einzelnen Warteschlangen erfolgen muss, um eine schnellere Bearbeitung der Chef-Aufträge zu gewährleisten.

Ansatz	Auslastung Engpass	normale Aufträge [2 x Std. Err]	Chef- Aufträge [2 x Std. Err]	Gesamt
1. FIFO	83 %	5,05 [+0,044]	5,33 [+0,046]	5,08
	95 %	9,74 [+0,072]	10,81 [+0,076]	9,84
2. max. Prio	83 %	5,28 [+0,046]	3,79 [+0,030]	5,08
	95 %	10,46 [+0,076]	3,93 [+0,030]	9,84
3. vari- able Plan- DLZ	83 %	5,22 [+0,042]	3,79 [+0,030]	5,08
	95 %	10,81 [+0,086]	4,10 [+0,032]	10,17

Tabelle 2: Darstellung der gemessenen durchschnittlichen Durchlaufzeiten nach Auftragsart in Tagen

Dies gelingt sowohl bei einer maximalen Priorisierung der Chef-Aufträge innerhalb der einzelnen Warteschlangen, als auch bei der Annahme einer variablen Plan-DLZ je nach Auftragsart (siehe Tabelle 2). Zu betrachten ist auch der Standardfehler der Durchlaufzeit, dieser ergibt sich nach Koschack [11] aus der Varianz der Durchlaufzeit der einzelnen Aufträge geteilt durch die Wurzel der Anzahl der Aufträge des Versuchs. Aus diesem Wert lässt sich die Streuung der Durchlaufzeiten ableiten, die für einen gut geplanten Produktionsablauf niedrig zu hal-

ten ist. Es zeigt sich, dass bei einer niedrigeren Auslastung die Nutzung einer variablen Plan-DLZ unter diesem Gesichtspunkt zwar etwas besser, aber nicht signifikant besser ist, bei einer hohen Auslastung sich eine maximale Priorisierung anbietet.

6 Fazit

Abschließend kann festgehalten werden, dass die Diskussion der Berücksichtigung von Chef-Aufträgen innerhalb der Fertigungssteuerung sinnvoll ist. Es konnte festgestellt werden, dass klassische Fertigungssteuerungsansätze keine lieferzeitvariable und rückstandsorientierte Bearbeitung von Aufträgen zulässt. Aufgrund der Nachfrage von Kundenseite ist solch eine Möglichkeit zur Steigerung der Profitabilität allerdings notwendig. Wie aufgezeigt wurde, bieten sich abhängig von den in der Praxis bestehenden Gegebenheiten unterschiedliche Variationen der Fertigungssteuerung an. So sollte im einfachsten Fall eine maximale Priorisierung der Chef-Aufträge erfolgen. Weitere Forschung sollte sich auf den in der Arbeit entwickelten Ansatz der variablen Plan-DLZ fokussieren, da es Potenzial zeigt, die Termintreue weiter zu verbessern. Darüber hinaus bedarf es der Diskussion weiterer Konzepte. Denkbar wäre etwa eine Reihenfolgevertauschung mit Rabattangeboten. Bei gezielter Nutzung dieses Ansatzes erfolgt eine Profitabilitätssteigerung, da eine Priorisierung von Chef-Aufträgen bei gleichbleibender Termintreue normaler Aufträge gelingt, ohne dass Kapazitäten zurückgehalten werden müssen.

Literatur

- [1] Lödning, H. (2016). *Verfahren der Fertigungssteuerung*. Berlin: Springer-Verlag.
- [2] Heger, J. (2014). *Dynamische Regelselektion in der Reihenfolgeplanung: Prognose von Steuerungsparametern mit Gaußschen Prozessen*. Springer-Verlag.
- [3] Jain, A. and Meeran, S. (1998). *A state-of-the-art review of job-shop scheduling techniques*. Technical report, University of Dundee.
- [4] Tempelmeier, H., & Günther, H. O. (2009). *Produktion und Logistik*. Berlin, Heidelberg, New York: Springer-Verlag.
- [5] Drexel A., Haase K. (2000) *Vorbeugende Instandhaltung und Eilaufträge in der Produktionsplanung*. Physica, Heidelberg
- [6] Trzyna, D. (2015). *Modellierung und Steuerung von Eilaufträgen in der Produktion* (Doctoral dissertation, Technische Universität Hamburg).
- [7] Eley, M.: Simulation in der Logistik: Einführung in die

Erstellung ereignisdiskreter Modelle unter Verwendung des Werkzeuges Plant Simulation. Berlin: Springer Gabler 2012.

- [8] Zheng, N., & Lu, X. (2009). *Comparative study on push and pull production system based on Anylogic*. In Electronic Commerce and Business Intelligence, 2009. ECBI 2009. International Conference on (pp. 455-458).
- [9] Kibira, D., Shao, G., & Johansson, B. (2016). *Framework for standardization of simulation integrated production planning*. In Proceedings of the 2016 Winter Simulation Conference (pp. 2970-2981). IEEE Press.
- [10] Rajendran, C., & Holthaus, O. (1999). *A comparative study of dispatching rules in dynamic flowshops and jobshops*. European journal of operational research, 116(1), 156-170.
- [11] Koschack, J. (2008). *Standardabweichung und Standardfehler: der kleine, aber feine Unterschied*. Zeitschrift für Allgemeinmedizin, 84(12), 258-260.

Effiziente Logistikplanung mittels angewandter Simulation

Anne-Sophie Sabrowski^{1*}, Dr.-Ing. Knut Borrmann¹

¹MVI PROPLANT Nord GmbH, Dieselstr. 33, 38446 Wolfsburg; *anne-sophie.sabrowski@proplant-mvi.com

Abstract. Die Beschreibung von Zuständen, die sich aus dem Zusammenspiel verschiedener Objekte in einer Struktur ergeben, ist selten ein leichtes. Insbesondere wenn Exaktheit und Determinismus gefordert sind, aber die Komplexität der Realität dem entgegensteht. So ist es nicht weiter verwunderlich, wenn für umfangreiche Planungsvorhaben die Realität soweit vereinfacht wird, bis sie im statischen Gerüst des ruhigen Gewissens eingezwängt brauchbare Resultate liefert. In Folge dessen entsteht allzu häufig eine überdimensionierte und ineffiziente Lösung, die einen bedeutenden Teil der „dynamischen Realität“ ignoriert. Für die logistische Planungsaufgabe, bei der es grundsätzlich gilt, dass Struktur, Objekt und Dynamik aufeinander abzustimmen sind, erscheint es unabdingbar, wenn sie um den Aspekt der Simulation erweitert wird. Schlussendlich ist nur mittels angewandter Simulation, so die feste Überzeugung, eine effiziente Logistikplanung erst möglich.

Einführung

Allzu häufig findet die Simulation bei der Planung innerbetrieblicher Abläufe eine zu geringe Berücksichtigung. Obwohl ihr Stellenwert hinlänglich bekannt ist, kommt es auf unternehmerischer Seite nicht selten zu einer strikten Trennung zwischen Planungs- und Simulationsaufgabe. Begründbar durch den jahrelangen Einsatz statischer Berechnungsverfahren, die sich als ein bewährtes und gültiges Werkzeug erwiesen haben, mit denen sich Planungsherausforderungen kundengerecht und erfolgreich bewältigen lassen. Die Frage nach der Sinnhaftigkeit, neben schon eingesetzten Planungsressourcen noch zusätzlich Simulationskapazitäten zu investieren, darf durchaus gestellt sein. Deshalb sollen im Folgenden die Vorteile einer in dem Planungsprozess integrierten Simulation näher beleuchtet werden.

1 Problemstellung

Die Potenziale der Materialflusssimulation liegen im Vergleich zu einer statischen Systembetrachtung in der Realisierung einer erhöhten Planungsqualität, in der Steigerung der Planungsgeschwindigkeit sowie in der Reduzierung der Planungs- und Investitionskosten [1].

In der Wissenschaft hat sich die Simulation daher als essentielles Planungswerkzeug während sämtlicher Planungsphasen bewährt [2].

Dahingegen können Unternehmen die aufgezeigten Potenziale der Materialflusssimulation häufig nicht nutzen. Die Gründe hierzu sind vielfältig und beruhen u. a. in einem in der Praxis wahrgenommenen unausgewogenen Aufwand-Nutzen-Verhältnis [3]. Dieser Eindruck verstärkt sich zwangsweise, wenn es während des Simulationseinsatzes an einer systematischen Abstimmung und Kommunikation sowie an einer klaren Aufgabenteilung zwischen den verantwortlichen Planungs- und Simulationsexperten mangelt, sodass der Erfolg einer planungsbegleitenden Simulation maßgeblich beeinträchtigt wird [4].

Daneben sind es aber auch die Entscheidungsträger selbst, denen sich die Potenziale und Möglichkeiten eines Simulationseinsatzes u. a. aufgrund unzureichender Simulationskenntnisse und -erfahrungen nicht erschließen. Deshalb ist es nicht weiter verwunderlich, wenn in herkömmlichen Logistikplanungsaufgaben kein konkreter Simulationsbedarf gesehen wird. Ebenso wenig dienlich ist die in der Praxis bestehende Vorstellung, was „Planungswelt“ und was „Simulationswelt“ zu sein hat.

Vor diesem Hintergrund ist die Verwendung der Simulationstechnik meist nur bei spezifischen Fragestellungen, in denen vielzählige dynamische Zusammenhänge bestehen, gefragt. Um jedoch in der Praxis die mit der Materialflusssimulation verbundenen Vorteile erzielen zu können, wird letztlich die Forderung nach einer ganzheitlichen Integration der Simulation in den Logistikplanungsprozess gestellt.

Obwohl diesem Ansinnen in verschiedenen Ansätzen nachgegangen wird und sich wahrlich keine Neuartigkeit daraus ableiten lässt, ist in dem hier vorgeschlagenen Vorgehensmodell der integrative Aspekt umfassender und praktikabler formuliert, als es die bisherigen Ansätze vermögen. Besonders die detaillierte Prüfung der Logistikplanungsaufgabe auf Simulationswürdig-

keit, die Bestimmung des Simulationseinsatzzeitpunktes sowie die Klärung und Definition der interdisziplinären-schnittstellenübergreifenden Zusammenarbeit geht über das bisherige Maß bekannter Ansätze hinaus. [5]

2 Vorgehensweise

Zur effizienten Gestaltung der Logistikplanung ist daher ein Vorgehensmodell zu entwickeln, welches die kontinuierliche Einbindung der Simulation in den Planungsprozess fokussiert und mit dem der gesamte Planungshorizont abgedeckt ist. Das Hauptaugenmerk des Modells liegt jedoch auf der Schnittstellengestaltung zwischen den Planungs- und Simulationsexperten. Indem ebenso deren Aufgabenverteilung vor, während und im Anschluss an die Simulationsstudie eindeutig definiert wird, kann den betrachteten Experten ein Leitfaden für eine verbesserte Zusammenarbeit zur Verfügung gestellt werden. Auf diese Weise wird die Nutzung der aufgezeigten Potenziale der Simulationstechnik auch in der Praxis erreicht.

Beruhend auf der Untersuchung bestehender Konzepte (Tabelle 1) sowie der Durchführung qualitativer Experteninterviews wurden zunächst in Relation zur aufgezeigten Situation wesentliche Anforderungen erarbeitet.

Kriterien	Ansätze						
	Reinhart 2000	Skudelny 1994	Ten Hompel et al. 2007	Gunthner et al. 1998	Jensen 2007	Malaga 2017	Dieser Ansatz
Möglichkeit der Integration der Simulation in den Planungsprozess	●	●	●	●	●	●	●
Eindeutige Vorgehensweise zur Simulationsintegration	◐	○	○	○	○	◐	●
Berücksichtigung der Planungs- und Simulationsexperten (Rollen)	○	○	○	○	◐	○	●
Bestimmung des Simulationseinsatzpunktes	○	◐	◐	○	○	○	●
Zentrale Datenverwaltung	○	●	○	●	●	●	●
Sukzessive Detaillierung der Planung im Simulationsmodell	○	●	○	◐	◐	◐	○
Assistenzsysteme & automatisierte Modellerstellung	○	○	○	●	●	●	○

Legende:

● wird umfassend thematisiert ◐ wird zum Teil thematisiert ○ wird nicht thematisiert

Tabelle 1: Erfüllung der Kriterien durch die verschiedenen Ansätze zur Integration der Simulation in den Planungsprozess [5]

Obwohl bereits vielfältige methodische und rechnergestützte Ansätze zur Integration der Simulation in den Planungsprozess existieren, zeigt sich in der Praxis, dass die angestrebten Potenziale einer Integration oft nicht erreicht werden. Häufig mangelt es den Ansätzen u. a.

- an einer eindeutigen Vorgehensweise zur Simulationsintegration,
- der Berücksichtigung von Planungs- und Simulationsexperten,
- der fehlenden Definition von Rollen der einzelnen Akteure im Planungsprozess und
- der Bestimmung sinnvoller Simulationseinsatzpunkte.

Ähnliche Aspekte äußern Simulations- und Planungsexperten, die für eine verbesserte Zusammenarbeit:

- einen standardisierten Ablauf,
- regelmäßige und systematische Abstimmungen,
- eine transparente Verteilung der Verantwortlichkeiten und Aufgaben sowie
- ein beiderseitiges Verständnis einfordern.

Vor diesem hier in der Kürze beschriebenen Hintergrund lassen sich aus den weitergehenden Untersuchungen und Interviews konkrete Ziele für das zu entwickelnde Vorgehensmodell ableiten. Demnach soll das Modell:

- die Bestimmung des korrekten, projektspezifischen Simulationseinsatzpunktes ermöglichen,
- eine zentrale Datenverwaltung als Voraussetzung zur effizienten Zusammenarbeit berücksichtigen,
- keine sukzessive Detaillierung der Planung am Simulationsmodell fördern (aufgrund des individuellen Simulationseinsatzes einzelner Planungsaspekte),
- keine rechnergestützte Methode zur automatisierten Modellerstellung (aufgrund nicht ausgereifter technologischer Schnittstellen) verwenden und
- ein Leitfaden für die Zusammenarbeit sein.

Darauf aufbauend lassen sich nun die wesentlichen Erfolgsfaktoren benennen (Abbildung 1). Bei fehlender Berücksichtigung dieser Faktoren ist der Erfolg des Vorgehensmodells grundlegend gefährdet. In der Quintessenz sind die tragenden Säulen eines integrativen Einsatzes der Simulation in den Logistikplanungsprozessen in der Harmonisierung des Wissens, einer einheitlichen Datenbasis und zweckmäßigen Dokumentation des Geschehens sowie in der Methodik, also das

Vorgehensmodell an sich, zu sehen (Abbildung 2).



Abbildung 2: Erfolgsfaktoren

Wissensharmonisierung:

Die Wissensharmonisierung bildet die Grundlage für die gemeinsame Zusammenarbeit der Experten sowie für das Durchlaufen des Vorgehensmodells. Es wird als wesentlich angesehen, sowohl die Planungsexperten über die Simulation als auch die Simulationsexperten über die Planungsgrundlagen vor der Kooperation aufzuklären. Durch die Stärkung des Bewusstseins der Experten für das jeweilig andere Fachgebiet wird das Vorgehensmodell eher akzeptiert und im Alltag erfolgreich eingesetzt. Die Wissensvermittlung kann dabei auf verschiedene Weisen erfolgen, die je nach Unternehmensstrategie variieren. Eine Möglichkeit besteht in der Ausbildung ausgewählter Multiplikatoren, die zunächst von Spezialisten geschult werden [6]. Diese können daraufhin die gewonnenen Kenntnisse an ihre Kollegen weitergeben und stehen diesen bei Herausforderungen im Alltag beratend zur Verfügung. Dadurch kann einerseits einer großen Anzahl an Planungs- und Simulationsexperten das notwendige Wissen vermittelt werden. Andererseits halten sich der Aufwand und die Kosten dieser Wissensweitergabe in Grenzen.

Um die Planungsexperten für den Simulationseinsatz zu sensibilisieren, müssen diese im Wesentlichen über folgende Inhalte informiert werden. Zunächst muss grundlegend aufgezeigt werden, was unter dem Begriff der Simulation zu verstehen ist und warum der Einsatz der Simulation im Planungsprozess von hoher Bedeutung ist. Des Weiteren ist es erforderlich, sowohl die Möglichkeiten als auch die Grenzen der Simulation darzulegen [8]. Auch die Erläuterung, wann die Simulation eingesetzt wird, bildet einen wichtigen Bestandteil

der Wissensvermittlung. Damit die Planungsexperten die Sinnhaftigkeit der Simulation erkennen, sollten zudem deren Ergebnismöglichkeiten beschrieben werden. Dabei ist im Besonderen auch die Abhängigkeit der Ergebnisqualität von den Eingangsdaten hervorzuheben. Zur Vereinfachung der Zusammenarbeit während der Simulationdurchführung sollte außerdem die hierzu angewandte Vorgehensweise vorgestellt werden. Um das allgemeine Verständnis der Planungsexperten für die Simulation zu fördern, ist es zweckmäßig, exemplarische Anwendungsbeispiele zu bearbeiten. Auf diese Weise erhalten diese ein Gefühl für den Ablauf von Simulationsprojekten sowie für den notwendigen Aufwand. Indem die Planungsexperten über die Simulationsgrundlagen und deren Vorteile aufgeklärt werden, können zudem die Hemmnisse des Simulationseinsatzes vermindert werden. Diese liegen oftmals vor, wenn die betroffenen Experten die Simulationstechnik selbst nicht beherrschen und diese als Kontrolle ihrer Arbeit wahrnehmen [7].

Zugleich ist es erforderlich, die Simulationsexperten in die Logistikplanung einzuführen. Dabei sollte diesen zunächst der Logistikplanungsprozess und die einzelnen Planungsaspekte vorgestellt werden. Indem die darin enthaltenen Meilensteine aufgezeigt werden, wird das Verständnis der Simulationsexperten für die Zeitknappheit während der Planung und für die Dringlichkeit der Simulationsergebnisse gefördert. Zudem ist das Verhältnis zum Auftraggeber während der Planung zu charakterisieren. Darüber hinaus ist es notwendig, mögliche simulationsrelevante Planungsinhalte zu verdeutlichen, welche mit Hilfe statischer Berechnungsmethoden oftmals nicht ausreichend dimensioniert werden können.

Dokumentation und Datenbasis:

Wie schon erwähnt stellen eine einheitliche Datenbasis und eine durchgängige Dokumentation wesentliche Erfolgsfaktoren des Vorgehensmodells dar. Auch Wenzel et al. [8] verdeutlichen die Wichtigkeit einer konsequenten Dokumentation bei der Durchführung von Simulationsstudien. Daher soll diese Säule die Simulationsintegration in den Planungsprozess unterstützen sowie als Basis für das methodische Vorgehen dienen. Indem während der Durchführung des Vorgehensmodells eine übersichtliche Datenablage und -verwaltung bereitsteht, können Ergebnisse einfach gespeichert und auf vorhandene Daten zugegriffen werden. Dabei ist eine zentrale Datenbasis anzustreben, die

sowohl von den Planungs- als auch von den Simulationsexperten verwendet und regelmäßig aktualisiert wird. Auf diese Weise können sowohl die Vorteile der Redundanzfreiheit als auch der Konsistenz und Übersichtlichkeit genutzt werden [9]. Zudem stehen den Experten bei Datenänderungen sofort die aktuellen Informationen bereit. Im Folgenden wird beispielhaft aufgezeigt, welche Daten und Dokumente im Rahmen des Vorgehensmodells erforderlich sind.

Zum einen gilt es, Daten zu sammeln und Dokumente zu erstellen, die während der Projektdurchführung und im Besonderen zur Erarbeitung des Simulationsmodells von Bedeutung sind [7]. Dazu zählen Planungsdaten [10], welche von den Logistikplanern zu Beginn des Vorgehensmodells bereitgestellt werden. Des Weiteren gilt es, simulationsspezifische Daten wie Bearbeitungs- und Störzeiten sowie Geschwindigkeiten zu beschaffen und zur späteren Aufbereitung abzulegen [8]. Zudem wird eine durchgängige Dokumentation aller Phasenergebnisse und Entscheidungsprozesse gefordert [11]. Dazu zählen u. a. die Ergebnisse der Prüfmethodik, die Zielbeschreibung, die Aufgabenspezifikation, das Simulationsmodell sowie Besprechungsprotokolle und Statusberichte. Diese Dokumente sind über den gesamten Projektverlauf anzufertigen, sodass Entwicklungsschritte nachvollziehbar und Vorgängerversionen wiederverwendbar sind. Im Wesentlichen dient diese Dokumentation zur Anwendung des Simulationsmodells sowie zur richtigen Interpretation der Simulationsergebnisse [7]. Auch für eine durchgängige Verifikation und Validierung (V&V) ist eine konsequente Dokumentation aller Daten und Modelle unverzichtbar [8]. Eine ausführliche Darstellung der Dokumentstrukturen zur Durchführung von Simulationsprojekten zeigen Rabe et al. [10] auf. Zum anderen enthält diese Säule Dokumente, die für eine erfolgreiche Projektorganisation erforderlich sind. Dazu zählt das an das jeweilige Projekt angepasste Vorgehensmodell. Des Weiteren wird empfohlen eine Verantwortlichkeitsmatrix, zum Beispiel ein individualisierter RACI-Chart sowie der damit verknüpfte Terminplan für alle Projektbeteiligten sichtbar abzulegen. Die systematische Dokumentation zur verwendeten Vorgehensweise und zur Absprache mit den Projektbeteiligten wird von Neumann [12] als wesentlicher Aspekt zur Weitergabe des Simulationswissens angesehen, der bisher meist keinen Eingang in die Dokumentation findet.

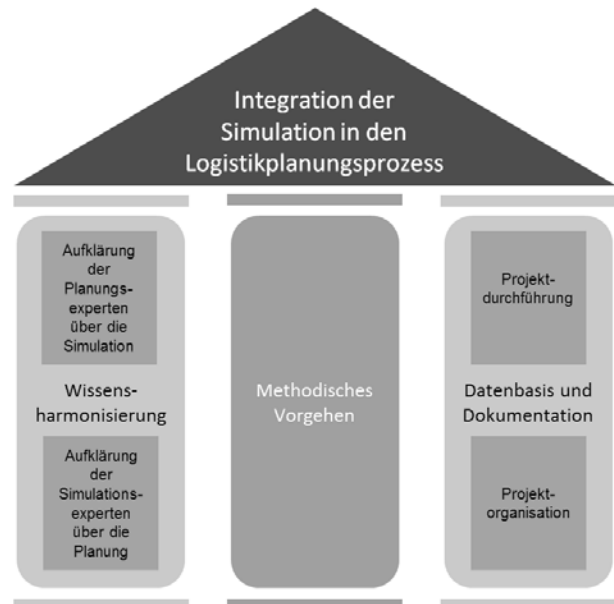


Abbildung 3: Voraussetzungen zur Integration der Simulation in den Logistikplanungsprozess [5]

Methodisches Vorgehen:

Flankiert von der Harmonisierung des Wissens sowie einer einheitlichen Datenbasis und einer zweckmäßigen Dokumentation ist das Vorgehensmodell zu entwickeln (Abbildung 4). Dieses berücksichtigt sowohl die Wahl des für die Simulation richtigen Planungsproblems als auch die Gestaltung der gemeinsamen Expertenarbeit. Vom übergeordneten Logistikplanungsprozess ausgehend erfolgt die Durchführung der Prüfmethodik durch den Planungsexperten. Zur Präzisierung des korrekten Simulationseinsatzpunktes, welcher von der individuellen Projektsituation abhängig ist, soll jeder Planungsaspekt auf einen Simulationsgrund sowie auf die Simulationswürdigkeit überprüft werden. Auf diese Weise wird die Bestimmung der Simulationsnotwendigkeit ermöglicht. Dabei dient die Klärung des Simulationsgrundes der

- Planungsabsicherung,
- Planungsverbesserung und
- Akzeptanzförderung,

während bei der Bestimmung der Simulationswürdigkeit festgestellt wird, ob

- statische Methoden,
- analytische Methoden,
- Experimente und / oder
- Erkenntnisse vorheriger Simulationsstudien

zur Problemlösung heranzuziehen sind.

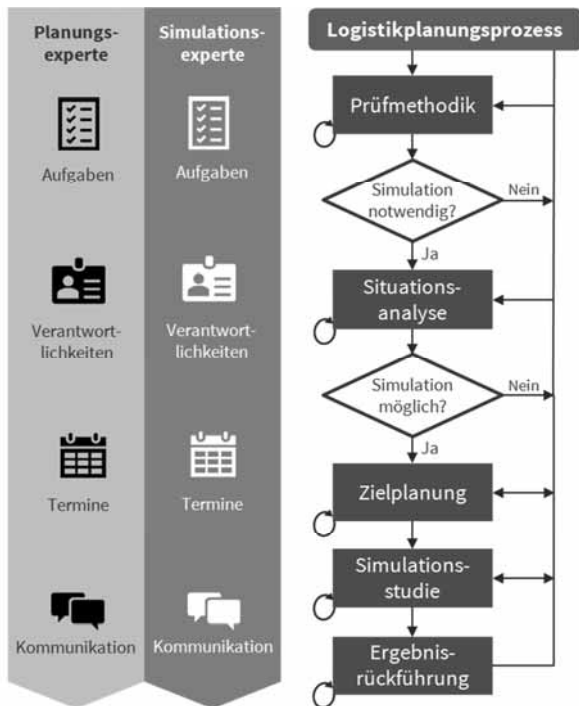


Abbildung 4: Vorgehensmodell zur Integration der Simulation in den Planungsprozess

Daraufhin wird im Rahmen der Situationsanalyse hinterfragt, ob die möglichen Simulationsergebnisse mit dem Erkenntnisinteresse seitens der Planung übereinstimmen. In dieser Phase steht insbesondere die Abstimmung zwischen den Planungs- und Simulationsexperten im Fokus, sodass eine Entscheidung über das weitere Vorgehen ermöglicht wird.

Zu klärende Punkte der Situationsanalyse:

- Expertenintegration
- Problemstellung
- Erkenntnisinteresse
- Ergebnismöglichkeit
- Nutzung bestehender Ergebnisse
- Individuelle Anpassung
- Aufwandsabschätzung und Freigabe.

Falls im Resultat die Simulation als gewinnbringend und durchführbar eingestuft wird, erfolgt der Übergang in die Zielplanung. Während dieser Phase gilt es, eine detaillierte Dokumentation der Fragestellung, der angestrebten Simulationsziele sowie der gegebenen Prämissen und Restriktionen als Ausgangsbasis für die Simulationsstudie zu erarbeiten.

Schritte der Zielplanung:

- Fragestellung
- Zielbeschreibung
- Projektumfang & -abgrenzung
- Prämissen und Restriktionen
- Datenabstimmung
- Terminplanung.

Daraufhin wird zur Durchführung der Simulationsstudie die bereits etablierte Vorgehensweise nach Rabe et al. [11] in das hier aufgezeigte Vorgehensmodell integriert.

Vorgehensweise der Simulationsstudie:

- Aufgabendefinition
- Systemanalyse
- Datenbeschaffung
- Modellformalisierung
- Datenaufbereitung
- Implementierung
- Experimente & Analyse.

Abschließend erfolgt die Phase der Ergebnisrückführung, in der es gilt, die generierten Simulationsergebnisse aufzubereiten und systematisch in den Planungsprozess zu integrieren. Zudem dient diese Phase der Reflexion sowie der kontinuierlichen Verbesserung des Vorgehensmodells. Als elementarer und abschließender Schritt soll der Simulationsexperte über die tatsächliche Ergebnisverwendung in der Planung informiert werden.

Schritte der Ergebnisrückführung:

- Ergebnisaufbereitung
- Integration der Simulationsergebnisse
- Präsentation & Dokumentation
- Lessons Learned
- Rückmeldung an den Simulationsexperten.

Zur Unterstützung der entwickelten Vorgehensweise wurden im Rahmen einer Verantwortlichkeitsmatrix die Rollen der Experten beschrieben. So sind zu jeder Phase die jeweiligen Aufgaben, deren Verantwortlichkeiten und Termine definiert (Abbildung 4). Zudem wird auf diese Weise die einzuhaltende Kommunikation und Abstimmung zwischen den Experten charakterisiert. Da die Planungs- und Simulationsexperten im Rahmen dieses Vorgehensmodells eine bedeutende Rolle spielen, werden diese in erster Linie in der Verantwortlichkeitsmatrix berücksichtigt. Der Simulationsexperte nimmt sowohl die Rolle des Simulationsfachmanns als auch des V&V-Experten ein. Des Weiteren gilt es, die Aufgaben des Auftraggebers, der zugleich den Projektleiter des Planungsexperten darstellen kann, zu definieren. Zudem ist

auch die Rolle des Auftragnehmers auf Seiten des Simulationsexperten in das RACI-Chart aufzunehmen, der als dessen Vorgesetzter die Verantwortung trägt. Außerdem wird eine Rolle für die weiteren Spezialisten berücksichtigt, wobei hierzu die IT-Verantwortlichen und die Softwareexperten sowie die Beauftragten zur Modellabnahme zählen.

	Planungs- experte	Simulations- experte	Auftraggeber (zgh. Planungs- experte)	Auftragnehmer (zgh. Simulations- experte)	Spezia- listen	Termin plan
Prüfmethodik						
Simulationsgrund	R		A			
Simulationswürdigkeit	R		A			
Situationsanalyse						
Expertenintegration	R	I	A			
Problemstellung	R	I	A			
Erkenntnisinteresse	R	I	A			
Ergebnismöglichkeit	C	R		A		
Nachnutzung bestehender Ergebnisse	R	C	A			
Individuelle Anpassung	R	R	A	I		
Aufwandsabschätzung & Freigabe	R	R	C, A	C		
Zielplanung						
Fragestellung	R	I	A			
Zielbeschreibung	R					

Legende:

R = Responsible A = Accountable C = Consulted I = Informed

Abbildung 5: Auszug aus dem RACI-Chart zur Expertenabstimmung [5]

3 Anwendung des Vorgehensmodells am Beispiel der FTS-Routenplanung

Die praktische Validierung sowie die Prüfung der logischen Phasenordnung des erarbeiteten Vorgehensmodells erfolgt anhand einer experimentellen Anwendung zur Routengestaltung von Fahrerlosen Transportsystemen (FTS). Der Gebrauch der Simulationstechnik spielt während dieses Planungsaspekts eine wesentliche Rolle, da die Anzahl der benötigten Fahrzeuge sowie Aussagen über den zu erreichenden Durchsatz mit Hilfe analytischer Methoden nur näherungsweise bestimmt werden können [13]. Das herangezogene Beispielprojekt orientiert sich an der innerbetrieblichen Logistikplanung eines neuen Fahrzeugprojekts aus der Automobilindustrie. Den Planungsexperten obliegt die Logistikverantwortung bis zum ersten Anstellort für die Bereiche Karosseriebau, Montagelinie und Vormontage. Im Rahmen der Logistikplanung ist als wesentliches Prinzip die staplerfreie Versorgung der Materialbedarfsorte zu berücksichtigen. Daher wird eine effiziente Nutzung von Kapazitäten und Ressourcen durch den Einsatz von FTS anstatt Staplern angestrebt.

Im Zuge der betrachteten Projektphase, der Technik- und Routenplanung, sind die Fahrzeugwahl für den innerbetrieblichen Behältertransport im Bereich Fahrzeugunterbau zu definieren und die Routen für die Materialanstellung festzulegen. Zum aktuellen Planungszeitpunkt ist der Transport der Spezialladungsträger (SLT) mit Hilfe eines FTS zu untersuchen. Dabei sollen die Fahrerlosen Transportfahrzeuge (FTF) als automatisierte Unterfahrschlepper dienen, sodass die SLT von diesen am Behälterbahnhof aufgenommen und an den jeweiligen Bedarfsort geliefert werden.

Aufgrund der Betrachtung eines neuen Planungsaspekts wird ausgehend von der aufgezeigten Fragestellung die Prüfmethodik angestoßen. Dabei können diesbezüglich sowohl mehrere Simulationsgründe genannt (s.o.) als auch die Simulationswürdigkeit dieses Planungsaspekts bestätigt werden. Der Einsatz der Simulationstechnik wird daher als notwendiges und geeignetes Planungshilfsmittel erachtet.

Im Rahmen der anschließenden Situationsanalyse wird ersichtlich, dass das Erkenntnisinteresse seitens der Planung neben der Ermittlung der Funktionsfähigkeit, die mit Hilfe konventioneller Planungsmethoden nicht nachweisbar ist, auch die Kalkulation der zu tätigen Investitionen für den FTS-Einsatz umfasst. Darüber hinaus wird die übersichtliche Abbildung der dynamischen Wirkungszusammenhänge angestrebt. Zudem sind die Fahrerlosen Transportfahrzeuge sowie die Batterieladestationen zu dimensionieren. Die Ergebnismöglichkeit der Materialflusssimulation deckt hierbei nahezu vollständig das angestrebte Erkenntnisinteresse ab. Ob das FTS jedoch unter Berücksichtigung der zu tätigen Investitionen tatsächlich einzuführen ist, obliegt der Planung sowie letztendlich dem Auftraggeber.

Mit Hilfe der Zielplanung werden die zuvor grob aufgezeigten Inhalte der Situationsanalyse dokumentiert sowie spezifiziert. Als wesentliche Simulationsziele werden der Nachweis der Funktionsfähigkeit, die Ermittlung der Fahrzeuganzahl und der Anzahl der Ladestationen sowie der FTS-Auslastung festgehalten. Dabei sind die Verfügbarkeiten sowie die Reparaturzeiten der Anlagen und die Ladezeiten der FTF als wesentliche Einflussgrößen zu berücksichtigen.

Im Anschluss erfolgt die Durchführung der Simulationsstudie nach der etablierten Vorgehensweise, sodass neben der Bearbeitung der definierten Phasen auch geeignete V&V-Techniken angewendet werden.

Als abschließende Phase dient die Ergebnissrückführung

der Übertragung der Simulationsergebnisse in den übergeordneten Planungsprozess. Neben den mit Hilfe der Experimente gewonnenen Ergebnisdaten sind dabei auch die Systemzusammenhänge sowie relevante Erkenntnisdetails zu besprechen. Im Rahmen der Experimente konnte zum einen die Funktionsfähigkeit des betrachteten Systems nachgewiesen werden. Zum anderen wurde sowohl das FTS dimensioniert als auch dessen Auslastungsgrad bestimmt. Bei genauerer Betrachtung der Ergebnisse konnte zudem festgestellt werden, dass insbesondere der Transport der Bauteile mit einer hohen Umschlaghäufigkeit die ermittelte Fahrzeuganzahl bedingen. Zur Reduzierung der FTF-Anzahl sowie zur Erhöhung der FTS-Auslastung sind mit dem Auftraggeber daher prozessorganisatorische sowie strategische Anpassungen zu diskutieren. Mit Hilfe der Animation wurde das System zudem auf Blockierungseffekte untersucht. Im eingeschwungenen Zustand kann es aufgrund der nicht prognostizierbaren Auftragslage zu konzentrierten Stauungen am Behälterbahnhof kommen. Hier müssen genügend Auswegmöglichkeiten der FTF bereitgestellt werden, sodass eine Behinderung weiterer Fahrzeuge vermieden wird.

Zusätzlich erfolgt in dieser letzten Phase auch die Reflexion des Vorgehensmodells im Rahmen einer Lessons Learned. Als wesentliches Ergebnis dieser ist festzuhalten, dass vor allem innerhalb der Situationsanalyse ein angeregter Wissensaustausch zwischen den Experten anzustreben ist. Nur so können alle wesentlichen Einflussgrößen des Systems berücksichtigt und die Möglichkeiten der Simulation passend zur Fragestellung ausgeschöpft werden. Des Weiteren ist zur Vermeidung von Unstimmigkeiten die Datenabstimmung bereits frühzeitig zu detaillieren. Insbesondere ist es wichtig, die diesbezüglichen Erwartungen der Experten zu hinterfragen. Im Rahmen der Experimentdurchführung wurde zudem deutlich, dass eine kontinuierliche Abstimmung mit dem Planungsexperten zur bestmöglichen Lösungsermittlung unumgänglich ist. Insgesamt konnte mit Hilfe der systematischen Vorgehensweise die Routenplanung des FTS durch den Simulationseinsatz effizient und strukturiert gestaltet werden.

4 Resultat

Die Materialflusssimulation bietet als dynamische Planungsmethode die Möglichkeit, Auswirkungen stochastischer Einflussgrößen zu berücksichtigen und so auch

komplexe Wirkungszusammenhänge eines Logistiksystems abzubilden [14]. Im Gegensatz zu statischen Berechnungen können auf diese Weise zahlreiche Vorteile hinsichtlich der Planungsqualität und -genauigkeit generiert werden [15]. Der Forderung nach einer durchgängigen Integration der Simulation in den Planungsprozess kann allerdings in der Praxis oftmals nicht nachgegangen werden. Ein wesentlicher Grund dafür besteht in der unzureichenden und ineffizienten Abstimmung der Planungs- und Simulationsexperten, sodass die Nutzenaspekte der Simulation nicht ausgeschöpft werden [11]. Daher zielt dieser erweiterte Ansatz darauf ab, mit Hilfe des dargestellten Vorgehensmodells die Experten Zusammenarbeit im Rahmen der simulationsbasierten Planung durch eine eindeutige Aufgabenverteilung zu verbessern. Zusätzlich zur praktischen Validierung am Beispiel der FTS-Routenplanung wurde dieses im Rahmen einer Gruppendiskussion von den zukünftig anwendenden Experten als gewinnbringende Unterstützung während der gemeinsamen Zusammenarbeit eingeordnet. Insgesamt wurde so erstmals ein allgemeingültiger Leitfaden zur Integration der Simulation in den Planungsprozess erarbeitet, der eine sorgfältige Vorbereitung der Simulationsstudie, eine systematische Projektdurchführung über alle Phasen sowie einen strukturierten Abschluss enthält. Darüber hinaus wird durch dessen Einsatz eine Effizienz- und Qualitätssteigerung des Planungsprozesses begünstigt.

Kritische Auseinandersetzung:

Kritisch hinsichtlich des Vorgehensmodells ist anzumerken, dass dessen Anwendbarkeit im Unternehmen aufgrund verschiedener Aspekte nicht umfassend sichergestellt werden kann. Einerseits konnte die Zusammenarbeit der Planungs- und Simulationsexperten aufgrund der beispielhaften Anwendung nicht hinreichend untersucht werden. So ist die Eignung des Vorgehensmodells zur tatsächlichen Verbesserung dieser zu hinterfragen. Auch die Auswirkungen der Realisierung der Wissensharmonisierung sowie der einheitlichen Datenbasis wurden im Beispielprojekt nicht weiterführend evaluiert. Obwohl deren Notwendigkeit mit Hilfe der wissenschaftlichen Ansätze sowie der Experteninterviews herausgestellt wurde, sind deren erforderliche Struktur und der damit verfolgte Nutzen differenziert herauszuarbeiten. Andererseits ist die Motivation der Experten zur Verwendung der erarbeiteten Vorgehensweise nicht hinreichend abgesichert. Zwar sind die

Experten mit Hilfe der heranführenden Interviews sowie der abschließenden Gruppendiskussion in die Modellerarbeitung und -verbesserung integriert worden. Allerdings wurden die Potenziale aufgrund der fehlenden realen Projektanwendung nicht anschaulich verdeutlicht.

Ausblick:

Zukünftig ist die Anwendbarkeit des Vorgehensmodells am realen Projekt zu validieren, sodass die Expertenzusammenarbeit näher untersucht und damit die tatsächliche Verbesserung der Abstimmung evaluiert werden kann. Anschließend sind die schrittweise Einführung des Modells in der Praxis sowie dessen Einbindung in den Planungsalltag zu fokussieren. Dabei ist auch die Durchführung von zielgerichteten Mitarbeiterschulungen notwendig. Das detailliert erarbeitete Vorgehensmodell bietet damit einen erheblichen Mehrwert und vielseitige Einsatzmöglichkeiten in zukünftigen Planungsprojekten aus dem Bereich der Produktion und Logistik. Denn nur mittels angewandter Simulation sind die aufgezeigten Effizienzpotenziale im Rahmen der Logistikplanung tatsächlich zu realisieren.

References

- [1] Martin, H. *Transport- und Lagerlogistik. Planung, Struktur, Steuerung und Kosten von Systemen der Intralogistik*. 8., überarb. und erw. Aufl. Wiesbaden: Vieweg+Teubner Verlag; 2011. 534 p.
- [2] Eley, M. *Simulation in der Logistik. Einführung in die Erstellung ereignisdiskreter Modelle unter Verwendung des Werkzeuges „Plant Simulation“*. Berlin Heidelberg: Springer-Verlag; 2012. 327 p.
- [3] Kumpf, A. *Anforderungsgerechte Modellierung von Materialflusssystemen zur planungsbegleitenden Simulation* [Dissertation]. [Maschinenwesen, (GER)]. Technische Universität München; 2001.
- [4] Kudlich, T. *Optimierung von Materialflusssystemen mit Hilfe der Ablaufsimulation* [Dissertation]. [Maschinenwesen, (GER)]. Technische Universität München; 2000.
- [5] Sabrowski, A.-S. *Möglichkeit der Integration der Materialflusssimulation in den Logistikplanungsprozess - Effizienzsteigerung der Routengestaltung von Fahrerlosen Transportsystemen*. [Masterarbeit]. [Produktionsorganisation und Fabrikplanung, (GER)]. Universität Kassel; 2018.
- [6] Unger, K. Mitarbeiterschulung im großen Stil. Unternehmensinterne Trainer von entscheidendem Vorteil / Steigerung der gesamten Firmenkompetenz. *Die Tabak Zeitung*. 2011; (41).
- [7] Košturiak, J; Gregor, M. *Simulation von Produktionssystemen*. Vienna: Springer-Verlag; 1995. 198 p.
- [8] Wenzel, S; Collisi-Böhmer, S; Pitsch, H; Rose, O; Weiß, M. *Qualitätskriterien für die Simulation in Produktion und Logistik. Planung und Durchführung von Simulationsstudien*. Berlin Heidelberg: Springer-Verlag; 2008. 220 p.
- [9] Feuerabend, T. *Planungs- und Projektsteuerung auf Basis relationaler Datenmodelle* [Dissertation]. [Bauwesen, (GER)]. Universität Dortmund; 2005.
- [10] Rabe, M; Spieckermann, S; Wenzel, S. *Verifikation und Validierung für die Simulation in Produktion und Logistik. Vorgehensmodelle und Techniken*. Berlin Heidelberg: Springer-Verlag; 2008. 237 p.
- [11] Rooks, T. *Rechnergestützte Simulationsmodellgenerierung zur dynamischen Absicherung der Montagelogistikplanung bei der Fahrzeugneutypplanung im Rahmen der Digitalen Fabrik* [Dissertation]. [Mathematik/Informatik, (GER)]. Technische Universität Clausthal; 2009.
- [12] Neumann, G. Wissensmanagementkonzepte für die Validierung und Qualitätssicherung in der prozessorientierten Logistiksimulation. In: Engelhardt-Nowitzki, C; Krenn, B; Nowitzki, O. *Management komplexer Materialflüsse mittels Simulation*. 1. Aufl. s.l.: Gabler Verlag (Leobener Logistik Cases); 2008. p 183-204.
- [13] Lienert, T; Fottner, J. Entwicklung einer generischen Simulationsmethode für das zeitensterbasierte Routing Fahrerloser Transportfahrzeuge. *Logistics Journal: Proceedings*. 2017; (10): p 1-10. Doi: 10.2195/lj_Proc_lienert_de_201710_01
- [14] Bracht, U; Rooks, T; Adrian, R. Virtuelle Logistikplanung für die Montage im Rahmen der Digitalen Fabrik. *Advances in Simulation for Production and Logistics Applications*. 2008: p 439-447.
- [15] Klug, F. *Logistikmanagement in der Automobilindustrie. Grundlagen der Logistik im Automobilbau*. Berlin Heidelberg: Springer-Verlag; 2010. 479 p.

An approach to a self-organizing production in comparison to a centrally planned production

Torsten Munkelt^{1*}, Martin Krockert^{1#}

¹Dresden University of Applied Sciences Faculty of Computer Science, Friedrich-List-Platz 1, Dresden, 01069, GERMANY *torsten.munkelt@htw-dresden.de / #martin.krockert@htw-dresden.de

Abstract. Industry 4.0, a part of the German high tech strategy, prefers self-organization in production over central production planning for the sake of greater flexibility, faster response to disruptions and to deviations, and less effort. Current planning systems usually plan centrally. We developed a universal self-organizing production and empirically compared its performance to a centrally planned production. Compared to most other approaches, the self-organizing production does not just schedule already existing operations, but starts with material requirements, explodes the bills of materials, creates production orders, and schedules backwards. The self-organizing production is based upon agents. For better comprehensibility, we additionally implemented central planning. The results of self-organization in production are promising in relation to central planning; especially when disruptions and deviations occur. Furthermore, the results suggest as best solution a combination of rough central planning and self-organization.

Introduction

Current ERP, APS, and ME systems plan, control, and optimize production rather centrally. The systems provide “naive” resources with the following targets: when, what, where, how much, and by what means to purchase, to produce, to store, etc. Thereupon, the resources give feedback regarding time, quantity, and completion of production. According to the feedback, the above systems plan, control, and optimize anew after every shift or just every night; some are even able to react event-based but still plan centrally. Industry 4.0 turns away from central planning, control, and optimization and promotes self-organization in production [1] which is expected to result in greater flexibility, in faster response to disruptions of the production and deviations in processing times, and in less computational effort. In this contribution, we confirm empirically that pure self-organization in production is possible. We compare a centrally planned and a self-organizing production empirically. Our experiments confirm that a self-organizing production performs as good as a centrally planned production [2] in most cases and often even better; especially when disruptions or deviations of processing times occur.

1 Self-organization in our production

Since self-organization is always an ability of a system, we initially define a system: A system is a set of components, which interact with each other and differ from the environment of the system. [2] There are many different definitions of self-organizing systems, e.g. “A self-organizing system is a system that changes its basic structure as a function of its experience and environment.” [3] and “[A self-organizing system is a system in which] pattern formation occurs through interactions internal to the system, without intervention by external directing influences.” [4]

In literature, the following properties are attributed to self-organizing systems [5, 4, 6]. Those properties may not be mutually exclusive and collectively exhaustive:

- 1 Openness: The system exchanges matter, energy, or information with its environment, and the exchange is the cause of the development of the system.
- 2 Adaptiveness: The systems adapts to changes in its environment.
- 3 Autonomy: The system and its components process all (external) stimuli from the environment according to internal mechanisms.
- 4 Nonlinearity: The output of the system does not always change proportionally to its input.
- 5 Indeterminism: The system develops in a way, which is hard to predict because it depends on random events, or almost equal inputs lead to completely different paths of development.
- 6 Attraction: Over time, the system converges asymptotically to a region (a so-called attractor) in the state space of the system, and if the environment disturbs the system only slightly, the system will stay near the region.
- 7 Path dependence/operational closedness: The history (path of development) of the system influences the further development of the system much more than the

environment of the system does.

- 8 Emergence: The system develops new spatio-temporal characteristics or structures as result of the interaction of its components. Those new characteristics and structures cannot be explained by the rather simple behavior of the single components.
- 9 Autopoiesis: The system or its components can autonomously create the components of the system. Hence the systems preserves itself.

According to the above definitions, properties 1, 2, 3, 7 and 8 are essential for a self-organizing system. Thus, if a system exhibits these properties, we will call it a self-organizing system. A production is a system, which produces goods [7]. If the production is also a self-organizing system, then it is a self-organizing production (system). Productions often also exhibit property 5. Because of property 5, analytical methods often fail to analyze productions successfully. Hence, we analyze our production by means of discrete-event simulation.

Self-organization in production is a central topic of Industry 4.0. The German government introduced Industry 4.0 as part of its high-tech strategy in 2011. [8] Industry 4.0 describes the tight interlinkage between industrial production and modern information and communication technology. [10] Industry 4.0 is supposed to enable self-organizing production. Industry 4.0 comprises “intelligent” resources and “intelligent” materials. Resources are humans, machines, means of transportation, and storage systems. Materials are raw materials, purchase parts, assembly groups as well as final products. Resources and materials communicate and cooperate directly and locally and should thereby self-organize the complete value-added network. Communication and cooperation span all phases of the product lifecycle. Concepts, programs, or platforms similar to Industry 4.0 are US-American “Smart Manufacturing”, “Advanced Manufacturing”, or “Industrial Value-Chain Initiative” [9], or the Chinese “Made in China 2015”.

Production planning, control, and optimization already successfully applied self-organizing systems like self-organizing maps, artificial ant colonies [10], artificial neural networks [11], and genetic algorithms. However the combination of self-organizing production and Industry 4.0 leads to intelligent autonomous behaviour of machines and material, their communication, distributed computation, and in the end to multi-agent systems (MAS). For a definition of MAS and its agents, we refer

to the Foundation for Intelligent Physical Agents [14]. There are three possible relationships between the agents on one side and the “intelligent” resources and materials on the other: Resources and materials are (two types of) agents, or agents connect to resources and materials physically, or agents are digital twins [12] of resources and materials. In the latter two cases, the agents make resources and materials “intelligent”. In either case, the agents communicate and collaborate. Thus, the production organizes itself. Additionally, our MAS exhibits property 9: The MAS and its agents can autonomously create its agents.

2 A Giffler-Thompson-based resource planning to challenge the self-organized productions

We implemented our own central production planning in order to compare a self-organizing production and centrally planned production fairly and comprehensibly. We adopted and stripped data structures and algorithms from current ERP systems, like SAP, as far as possible.

Our algorithm for central planning adapts well known scheduling and planning algorithms. It works as follows: In the first phase, the algorithm creates recursively the network of requirements and satisfiers via bill of materials explosion. Requirements are positions of customer orders, positions of production orders, or stock requirements. Satisfiers are positions of purchase orders, positions of production orders, or stock. They satisfy the requirements. The algorithm starts with requirements, which are not yet satisfied. During the same phase, the algorithm merges and splits lots according to fixed lot sizes and it schedules the network backwards, starting at their due time. If at least one satisfier starts in the past, the algorithm will execute phase two and phase three: In the second phase, the algorithm schedules the network forwards. It begins with the satisfiers, which start in the past and have no predecessors. In phase three, the algorithm schedules the network backwards again. It starts at the maximum of two points in time: the due time or the end time calculated by forwards scheduling. Phase three of the algorithm guarantees that the satisfiers start as late as possible. Phase four of the central planning executes finite capacity planning. It adapts the algorithm of Giffler and Thompson [13], considers earliest start times additionally, and schedules first the operation of that production order with shortest remaining slack time. Our central

planning deliberately omits the phases of infinite capacity demand planning and capacity levelling because existing planning systems also do not automate these steps. Instead, we adjusted the parameters manually to guarantee a high and steady capacity utilization. Our central production planning operates periodically. When it starts anew, it changes remaining processing times and quantities according to the feedback from the production. Moreover, every new planning run firstly discards all connections between requirements and their satisfiers and secondly creates new connections with respect to the current situation of the production.

3 Our production created for the comparison of self-organization and central planning

In order to compare self-organization and central planning, we created a production (see Figure 1). It produces two products. The products are wooden toy trucks. The bill of material of each product is three levels deep and contains 30 materials. Each material is produced in up to three operations. The operations of one material can seize a machine more than once. The operations sum up to 20 per product. Transportation times are not yet considered, but can be represented by an additional operation assigned to the material.

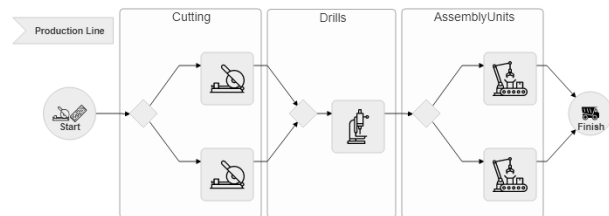


Figure 1: The main production flow involving two cutting machines, one drilling machine, and two assembly units. The production allows different routings for one product. The drilling machine is the bottleneck.

During the simulation of the production, new customer orders arrive at the production. The inter-arrival time of the orders is exponentially distributed as suggested in [16, 17]. We choose an inter-arrival time, which leads to a well-utilized production but does not cause an overload. The production runs for two weeks, 24 hours a day. The production will produce approximately 35 products per day if the processing times do not deviate. To examine the flexibility of the production, we vary the

processing times of the operations. According to [16] processing times are distributed log-normally. The inter-arrival times, the processing times, the capacity of the machines as well as the duration of the simulation can be configured separately for each simulation.

4 Multi-agent-based self-organization

4.1 Concept of a self-organizing production

As defined in Chapter 2, a self-organizing production adapts to environmental changes. Although its components are able to build hierarchical structures, they interact as equals. Simulations of self-organizing systems commonly apply multi-agent systems [14]. The “intelligent” resources and material of Industry 4.0 also suggest a multi-agent-based approach to a self-organizing production. Hence, we also applied a multi-agent-based approach.

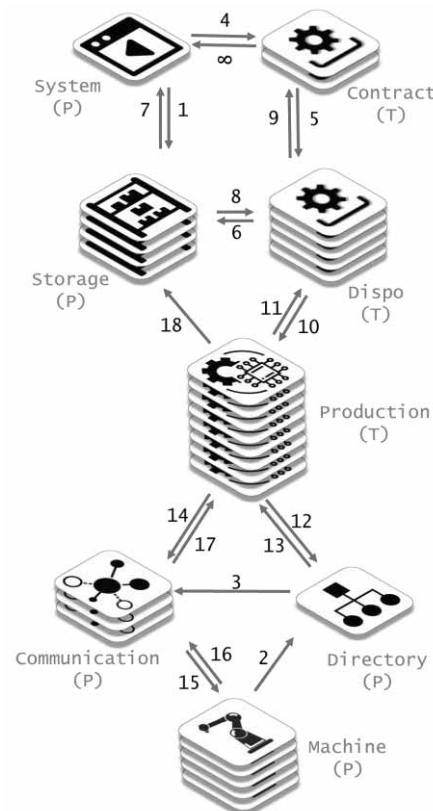


Figure 2: Concept of a multi-agent-based self-organizing production. Each symbol represents one type of agent. Each arrow is a communication path between two types of agents. In the text below, the numbers (#number) refer to the communications paths. The letters in brackets describe the type of the agents according to their lifetime: (P) .. persistent and (T) .. transient.

We defined eight different types of agents, which can be further divided into transient and persistent agents. The lifetime of transient agents starts and ends with their task. The lifetime of persistent agents starts and ends with the lifetime of the production. Figure 2 shows that all agents only know their direct neighbors. They are not aware of any other agent within the production.

Next, we explain the functionality of the multi-agent-based self-organizing production as well as the interactions between the agents. Therefore, we follow the processing of a single sales order throughout the production step by step.

4.2 System initialization

The MAS initializes the *system*, *machine* and *directory* agents during start-up and provides all required data, i.e. stock quantities to the *storage* agents (#1), machine capabilities to the *machine* agents, etc. The agent initialized first during start-up, is the *system* agent, followed by the *storage* agents and the *directory* agent. The *system* agent is responsible for all in- and outbound communication of the MAS with its environment. Second, the MAS creates all *machine* agents. The *machine* agents register at the *directory* agent and provide their capabilities (#2). The *directory* agent creates and assigns a *communication* agent to each machine group (#3) to provide a simple way of service discovery and to mediate between homogenous machines. Afterwards the production is ready to take orders and to start.

4.3 Order processing

If a new sales order arrives, the *system* agent will create a *contract* agent for this sales order (#4). The *contract* agent supervises the order fulfillment. Therefore, the *contract* agent creates one *dispo* agent (#5) for each product required to fulfill the order. The *dispo* agent represents a material in the real world: a digital twin. The *dispo* agent will ask the *storage* agent (#6) if there is an equivalent product in stock at the time the sales order requires it. The *storage* agent will check if there is a product in stock or needs to be purchased by the *system* agent. (#7). If the *storage* agent's response (#8) to the request is positive, the *dispo* agent's task will be done and it will signal its parent agent, in this case the *contract* agent, that the task is finished (#9). The *contract* agent signals the *system* agent that the *contract* is fulfilled (# ∞). If the *storage* agent's response is negative, the *dispo* agent will create a *production* agent for the required material (#10). Then,

the *production* agent instantiates new *dispo* agents for each of its components (#11). The cycle will start again until the bill of material of the ordered product is fully processed. The *contract* agent does not ask the *storage* agent for available material directly but the newly created *dispo* agent does because the *dispo* agent has to ask for available material anyway when asked by a *production* agent. Thus, only one agent keeps the same functionality.

4.4 Self-organizing scheduling with restricted horizon

The task of each *production* agent is not only to order its components but also to organize all required operations to assemble its assembly group. All *production* agents are competing for the earliest production timeslot on a machine. Therefore, the *production* agent requests one *communication* agent for each of its operations from the *directory* agent (#12). The *directory* agent returns the *communication* agent, which is responsible for the operation specified at the request (#13).

Afterwards, the *production* agent sends a "request for proposal" to the *communication* agent assigned to the operation (#14), similar to the Contract Net Protocol [20]. The *communication* agent forwards each incoming request to all of its *machine* agents (#15). Each *machine* agent calculates a possible start time based on the slack time of the operation and the current queue of the *machine* agent. That start time is returned as proposal by the *machine* agents (#15) to the *communication* agent (#16). After each *machine* agent returned a proposal to the *communication* agent, it decides for the proposal with the earliest start time and sends an acknowledgement to the assigned *machine* agent (#15). Furthermore, the *communication* agent sends the assigned scheduling information to the requesting *production* agent (#17).

The *machine* agent organizes the acknowledged proposals as operations in its own queue. Operations, which would not be processed for more than 60 minutes, are rejected with the prompt to try again after 45 minutes. Further analyses may address other time limits. If the *machine* agent receives an acknowledgement of an operation, this operation will be enqueued. Operations with a longer slack time than the previously enqueued operation are dropped from the queue, and the *machine* agent informs their *communication* agents to request new proposals for them (#16). This way, another *machine* agent could respond with a better proposal.

After the machine processes a material, its *machine*

agent sends a completion message to the *communication* agent (#16). The *communication* agent forwards the message to the related *production* agent (#17). The *production* agent receives the message and sets the status of the subsequent operation to ready, for machines are only allowed to process ready operations. If there is no subsequent operation, the *production* agent will assume that the material is fully processed and forward the completion message to the related *dispo* agent (#11) which terminates itself because its task is done. The *production* agent also sends a message to the *storage* agent (#18) and terminates itself because its task is also done. The *storage* agent provides the currently produced material to the requesting *dispo* agent (#6) with the shortest slack time. After the *dispo* agent receives the required material, the production cycle starts over until the product is fully assembled.

5 Results of the empirical comparison of a self-organizing and a centrally planned production

5.1 Parameters to vary

The simulation applied the parameters specified in Table 1. The production reached its steady state after approximately 24 hours. We added another 24 hours before we started the measurement of the KPIs. First, we chose a set of parameters both productions behaved well with: deviation of processing times $\pm 20\%$, lot size 1, planning scope 32 h, planning horizon 24 h, planning reset after 24 h, and average time to delivery 72 h. Applying these parameters, both productions had an average machine workload of 80% and delivered 100% of the sales orders in time. A major difference between the centrally planned and the self-organizing production was a significant deviation of lead times for both products. The centrally planned production had an average lead time of 4 h in contrast to the self-organizing production with an average lead time of 8 h. This results from longer lay time between the operations which cause higher stock for all components (see Table 2).

The difference between the lay times and lead times of the centrally planned production and the self-organizing production leads back to two reasons: The self-organizing production starts as early as possible, and the Giffler-Thompson algorithm of the centrally planned

production packs the operations tightly.

	Description	Type	Values
Deviation	Time duration deviation for operations	C/S	0/20/40 %
Lot-size	Lot size for assembly groups	C	1/5/10 pcs.
Planning scope	Limit time up which orders are taken into account	C	8/24/32 h
Planning reset	Time for rescheduling of all remaining and new orders	C	8/24 h
Order due	Average time from order placement to delivery	C/S	48/72 h

Table 1: Parameters, their descriptions, and their values for the initial simulation run, [C]entrally planned and [S]elf-organizing production.

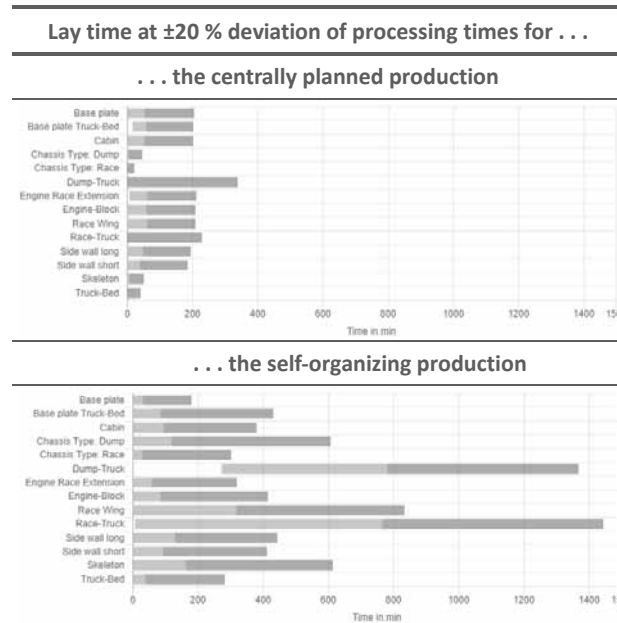


Table 2: Lay time range for each component. Green equals the lower half and red equals the upper half of the average lay time. The median lies at the transition from green to red.

5.2 Shortening the time to delivery

We shortened the time to delivery, to examine the performance of both productions under stress. Therefore, we ran the simulation with $\pm 20\%$ deviation, lot size 1, planning scope and reset after 24 h, and in average 48 h

to delivery. Again, both productions reached a similar average machine workload of 80 %. The lead time for the products increased moderately by about $\pm 20\%$. But the centrally planned production only delivered 14 % of the products in time. In contrast, the self-organizing production was able to adapt and delivered 100 % of the products in time. Table 3 shows the results of the simulation and visualizes that the self-organizing production was not only able to produce everything in time, but also produced 41 more end products during the analyzed period of time.



Parameter	Values	
	centrally planned	self-organizing
Simulation Type	centrally planned	self-organizing
total timeliness in [%]		
Quantity in [pcs.]	385	423
Max/Min deviation from delivery time in [min]	-320/385	-1431/-7

Table 3: Product timeliness with $\pm 20\%$ deviation and an average of 48 hours to delivery for all orders.

We tried to increase the timeliness of the centrally planned production. For this purpose, we tested fixed lot sizes with 5 and 10 pieces; but the timeliness increased only marginally. In further tests, we allowed the centrally planned production to plan anew every 8 hours but the timeliness did not improve either, because new customer orders enter the system more often and processing times deviate all the time. This leads us to the conclusion that our central planning is not able to cope with shorter times to delivery. That is why we will apply professional planning systems for further comparison.

5.3 Varying the deviation of processing times of operations

Many disruptions and deviations may occur during the manufacturing of a product. Machines may break down, products may be rejected because of quality issues, personnel may drop out, operations may take more or less time than originally planned, etc. All these disruptions and deviations lead to deviating processing times of operations. We varied the duration of processing times of operations randomly by $\pm 20\%$ and $\pm 40\%$ to examine the influence of the variations on the KPIs. The self-organizing production was able to handle both. It handled $\pm 20\%$

deviation with ease, and even at $\pm 40\%$ deviation, it delivered 94 % of all incoming sales orders in time. The centrally planned production performed even worse than expected: At $\pm 20\%$ deviation, the centrally planned production reached 14 % timeliness only, and at $\pm 40\%$ deviation, it was not able to finish any product in time. Due to the high deviation of processing times and the rather long time between two consecutive central planning runs, the average lead times grew from 200 min (at $\pm 20\%$ deviation) up to 3000 min (at $\pm 40\%$ deviation). In contrast, the self-organizing production had a maximum of average lead time of 1100 min (see Table 4).

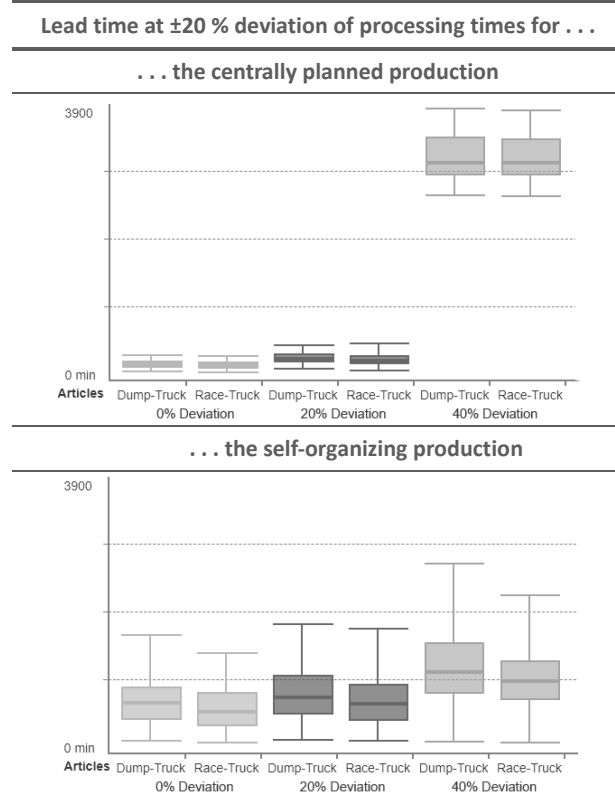


Table 4: Lead time comparison from left to right with 0 %, $\pm 20\%$ %, and $\pm 40\%$ % deviation of processing times.

To reduce the lead times in the self-organized production, we delayed the production start like it is common [23]. Only when the time remaining until due time was equal to the upper fence [24] of the previous lead times, the production started. The deviation of the processing times of the operations stayed at $\pm 20\%$. The delay led to lead times almost equal to the lead times achieved by the centrally planned production (see Table 5). Furthermore, the delayed start of the production reduced overall stock

in the self-organizing production to values even lower than in the centrally planned production (see Table 6). Despite the delayed start of the production, the self-organizing production still completed all orders in time.

Table 5 also shows quartiles of the lead times of the self-organizing production. These quartiles are more than twice as wide as the quartiles of the lead times of the centrally planned production. To narrow the quartiles of the lead times of the self-organizing production, we will investigate how different priority rules and load-dependent start times influence lay times, lead times, and stock. In order to achieve even better results, we will combine rough central planning and our self-organizing production.

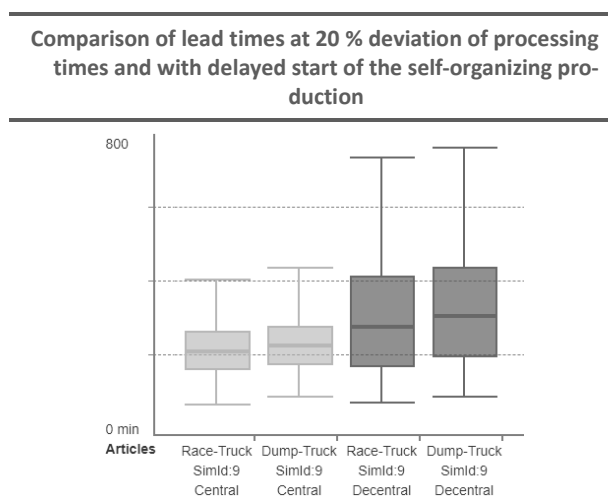


Table 5: Lead time comparison ([1, 2] centrally planned production; [3, 4] self-organizing production)

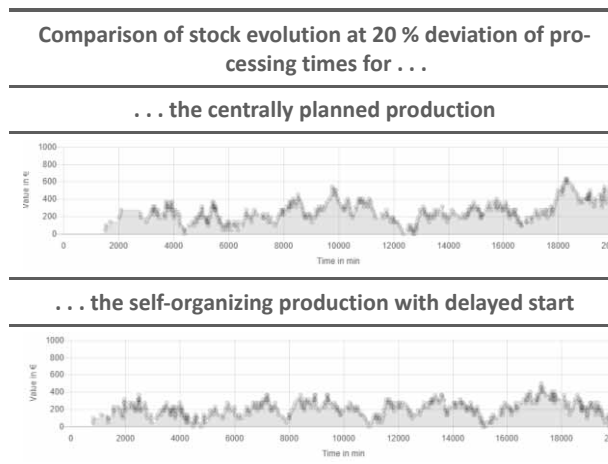


Table 6: Stock values over time in comparison

6 Conclusion and outlook

The target of our research was to develop a concept for a self-organizing production and prove the concept empirically. We developed the concept and a prototype of a self-organizing production. We let the prototype compete with a widely used and well-established algorithm for central production planning. We simulated a self-organizing and a centrally planned production with the same presets of random data and were able to prove the viability of the self-organizing production under the given circumstances. The results show that a self-organizing production is extremely robust against disruptions and deviations during physical production. While the centrally planned production stuck with its originally created plan, the self-organizing production timely adapted to deviations in processing times. The tradeoff were longer lay times, longer lead times, and higher stock.

Next, we will incorporate set-up times and dynamic lot-sizing in our production.

References

- [1] H. Lasi, „Industrie 4.0,“ *Wirtschaftsinformatik*, pp. 261-264, 04 04 2014.
- [2] B. Scholz-Reiter, M. Görges und T. Philip, „Autonomously controlled production systems—Influence of autonomous,“ in *CIRP Annals - Manufacturing Technology*, Bremen, 2009.
- [3] . W. F. Buckley, *Society - a Complex Adaptive System: Essays in Social Theory*, Amsterdam: Gordon and Breach, 1998.
- [4] W. Clark und B. Farley, „Simulation of self-organizing systems by digital computer,“ *Transactions of the IRE Professional Group on Information Theory*, Bd. 4, Nr. 4, p. 8, 1954.
- [5] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, T. Guy und B. Eric, *Self-Organization in Biological Systems*, Woodstock: Princeton University Press, 2003.
- [6] M. Köhler-Bußmeier, „Koordinierte Selbstorganisation und selbstorganisierte Koordination,“ Hamburg, 2010.
- [7] S. Kauffman, *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*, Oxford : Oxford University Press, 1996.
- [8] G. Schuh, *Produktionsplanung und -steuerung*, Aachen: Springer Verlag, 2006.
- [9] P. D. H. Kagermann, P. D. W. Wahlster und D. J. Helbig, „Umsetzungsempfehlungen für das

- Zukunftsprojekt Industrie 4.0,“ Frankfurt/Main, 2013.
- [10] T. Bauernhansl und M. t. Hompel, Industrie 4.0 in Produktion, Automatisierung und Logistik, Wiesbaden: Springer, 2014.
- [11] R. Anderl, R. Dumitrescu, M. Eigner, C. Ganz, A. S. Huber, J. S. Michels, T. Rückert, S. Tian, R. Stark und P. Zhi, Industrie 4.0 genzenlos, U. Sandler, Hrsg., Heidelberg: Springer Vieweg, 2016.
- [12] C. Klüver, J. Klüver und S. Jörn, Modellierung Komplexer Prozesse durch naturanaloge Verfahren, 2 Hrsg., Wiesbaden: Spinger Verlag, 2012.
- [13] Z. Hammami, W. Mouelhi und L. B. Said, „A self-adaptive neural-agent-based decision support system for solving dynamic,“ in *Intelligent Systems and Knowledge Engineering*, Tunis, 2015.
- [14] S. Poslad, „Specifying protocols for multi-agent systems interaction,“ in *ACM Transactions on Autonomous and Adaptive Systems*, New York, 2007.
- [15] D. M. Grieves, „Florida Institute of Technologie,“ 2014. [Online]. Available: http://innovate.fit.edu/plm/documents/doc_mgr/912/1411.0_Digital_Twin_White_Paper_Dr_Grieves.pdf.
- [16] G. Zäpfel und R. Braune, Moderne Heuristiken der Produktionsplanung, München: Vahlen, 2005.
- [17] J. Košturiak und M. Gregor, Simulation von Produktionssystemen, Wien : Springer Verlag, 1995.
- [18] I. N. Bronstein, K. A. Semendjajew, M. Gerhard und H. Mühlig, Taschenbuch der Mathematik, Frankfurt am Main: Harri Deutsch, 2016.
- [19] S. Völker, Reduktion von Simulationsmodellen zur simulationsbasierten Optimierung in der Termin- und Kapazitätsplanung, Frankfurt am Main: Peter Lang Verlag, 2003.
- [20] M. E. G. Jonathan, Y. Hujun, W. A. Wright und N. M. Allinson, „Self-Organization, Emergence and Multi-Agent Systems,“ in *Neural Networks and Brain, 2005. ICNN&B '05. International Conference on Volume: 3*, Beijing, 2005.
- [21] S. Ried, „The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver,“ in *Transaction on computers, Vol. C29 No 12*, 1980.
- [22] H. Corsten und R. Gössinger, Produktionswirtschaft, 13. Hrsg., München: Oldenbourg-Verlag, 2012.
- [23] W. Mendenhall, R. J. Beaver und B. M. Beaver, Introduction to Probability and Statistics, Belmont, CA, USA: Cengage Learning, 2008.
- [24] W. Dilger, „Multiagentensysteme,“ Chemnitz, 2005.
- [25] C. Hewitt, „A universal modular ACTOR formalism for artificial intelligence,“ in *IJCAI'73 Proceedings of the 3rd international joint conference on Artificial intelligence*, Stanford, 1973.
- [26] P. Kirmse, „Plattform Industrie 4.0,“ 2017. [Online]. Available: <http://www.plattform-i40.de/I40/Redaktion/DE/Anwendungsbeispiele/137-wandelbare-r8-manufaktur/beitrag-wandelbare-r8-manufaktur.html>. [Zugriff am 19.11.2017].
- [27] A. Roth, Einführung und Umsetzung von Industrie 4.0, Heidelberg: Springer Gabler, 2016.
- [28] J. Soder, Handbuch Industrie 4.0, 2 Hrsg., Bd. 1, B. Vogel-Heuser, T. Bauernhansl und M. ten Hompel, Hrsg., Springer, 2017.
- [29] S. Schukraft, S. Grundstein, B. Scholz-Reiter und F. Michael, „Evaluation Approach for the Identification of Promising Methods to Couple Central Planning and Autonomous Control,“ in *International Journal of Computer Integrated Manufacturing*, 2015.

Open Simulation Software - Development and Application

Tom Warnke^{*}, Frank Krüger, Adelinde M. Uhrmacher

Institute for Computer Science, University of Rostock, Albert-Einstein-Straße 22, 18059 Rostock

^{*}*tom.warnke@uni-rostock.de*

Abstract. Developing and applying software for simulation-based research in open science must satisfy two opposing requirements. On the one hand, simulation software needs to evolve continuously to adapt to technical advances and changing requirements. On the other hand, fixed versions of specific software artifacts must be made available persistently to make simulation results reproducible. These requirements are especially challenging for complex software with many constraints, such as the build environment or dependencies on other software. In this article we show how existing technologies and concepts can be leveraged to tackle these challenges. In particular, we discuss how open source software, build tools, automatic dependency management, and persistent artifact storage contribute to effective and credible simulation-based research.

Introduction

A 2012 study about the retraction of articles in the life sciences revealed that the majority (67%) was retracted for reasons of scientific misconduct (i.e. plagiarism or scientific fraud) [1]. From the remaining, the reasons for the retraction were traced back to laboratory errors (~56%), analytical errors (~19%), and irreproducibility (~16%), as a later analysis showed [2]. The reproducibility of research receives increasing attention in the computational sciences as well, as exemplified by the ACM's artifact review and badging [3]. In many research areas, the efforts to increase the reproducibility of research overlap with the paradigm shift towards open science.

The principle of open science, publishing not only the results but also making raw data and analysis workflows publicly available, does not prevent scientific errors but allows the scientific insights to be validated by the community and thus strengthens the quality of science in general. The particular level of realization

thereby depends on the research discipline. Research domains as the biomedical sciences, for instance, heavily rely on the investigation of physical artifacts such as biological materials, which cannot be published directly. The computational sciences, in contrast, rely almost entirely on digital artifacts such as source code and simulation output. While this gives excellent opportunities for providing the entire scientific workflow to the scientific community, from the first idea to the final analyses, it also raises issues with respect to joining the different requirements. To support and encourage the adoption of open science principles as well as the reproducibility of research, guidelines and best practices for researchers in the computational sciences in general [4] or in particular in modeling and simulation [5] have been proposed.

In the following, we summarize some of the best practices and guidelines that have been defined in the literature, in particular for modeling and simulation research. We describe how we implement those guidelines while developing, distributing, and using simulation software. We also discuss why we believe that our approach contributes to high-quality, reproducible simulation-based research.

1 Open Artifacts for Reproducible Research

In their article, Stodden and Miguez [4] establish six best practice principles for dealing with research artifacts in the computational sciences. They propose to publish all data and source code under open licenses in public accessible repositories to enable other researchers to reproduce scientific findings. Moreover, they recommend the usage of workflow tracking by means of version control in order to provide provenance information during the actual research. Finally, like for scientific literature, citation for external software and data is suggested. While the recommendations of Stodden and Miguez are at a rather high level, Fehr et al. [6]

give some more detailed instructions with respect to the documentation of scientific open source software. In particular, they describe a list of files that should accompany the publication of scientific software. Besides a README file, additional files with, for instance, installation and usage instructions are listed. In order to document the dependencies for external software and libraries, the inclusion of a DEPENDENCIES file is recommended. Metadata about the software, version and date of release are supposed to be documented in additional file, named CODE. Like Stodden and Miguez [4], Fehr et al. [6] propose the usage of digital object identifiers (DOI), as used for scientific literature, to refer to research artifacts archived in public repositories. Peng [7] raises software “bugs” as an additional point of interest. While he considers them in the context of wrong and questionable results, issues of quality play a central role in the development of software in general. Today, it is not clear how users of published artifacts are informed about changes to the software that do not change the desired behavior, such as bugfixes or performance improvements.

While the above literature focuses on research software artifacts in computational science in general, here we concentrate on simulation software. Taylor et al. [5] describe various aspects of open science in the simulation domain. To this end, they provide a checklist to follow when conducting and publishing simulation based studies. Dalle et al. [8, 9] also provide guidelines for developing high quality modeling and simulation software with a focus on separation of concerns and reuse. In the following, we provide a list of requirements, which is compiled from the above literature but specialized to the development, distribution, and application of simulation software.

Separation of concerns: Separating simulation software and simulation model enables reusing simulation algorithms for several models as well as executing a model with several algorithms. The resulting loose coupling of components makes the software easier to develop, extend, and test.

Version tracking of simulators: Simulation algorithms typically implement some kind of semantics. Different versions of the algorithms, regardless of optimizations and improvements, should produce the same results when applied to the same model.

Version tracking of models: Keeping track of the

development of simulation models allows retracing model refinements and extensions.

Version tracking of experiments: Simulation experiments reveal the behavior of a model in a specific experimental setting, or may check that the model shows a certain behavior. Parallel to simulation models, inspecting the history of experiments allows insights into the development of the model behavior.

Data artifacts: In general, persisting the output data of simulation experiments is not necessary, as the experiment can be repeated. Input data that is used in validation or calibration experiments, on the other hand, needs to be made available. Aggregated results of computationally expensive simulation experiments, for example, can be persisted as well.

Explicit Dependencies: By separating simulation algorithm, model, and experiment, the dependencies between them become explicit. Dependencies on other types of artifacts, such as input data or software libraries, and requirements on the runtime environment need to be made explicit as well. Additionally, a (possibly automatic) method to supply the dependencies and prepare the runtime environment should be provided.

Publication: Simulation algorithms, models, and experiments as well as their dependencies must be published and persisted with a unique identifier (e.g., a DOI or an ID in a model repository [10]). Given some identifiers, acquiring the corresponding simulation experiments and reproducing them should be straightforward. In general, the publication of research artifacts should follow the FAIR principles [11] in order to allow researchers to access them.

2 Open simulation software in practice

We now show how the above requirements are implemented in our group, the Modeling and Simulation Group (<https://mosi.informatik.uni-rostock.de>) at the Department of Computer science of the University of Rostock. Our research focuses on developing novel

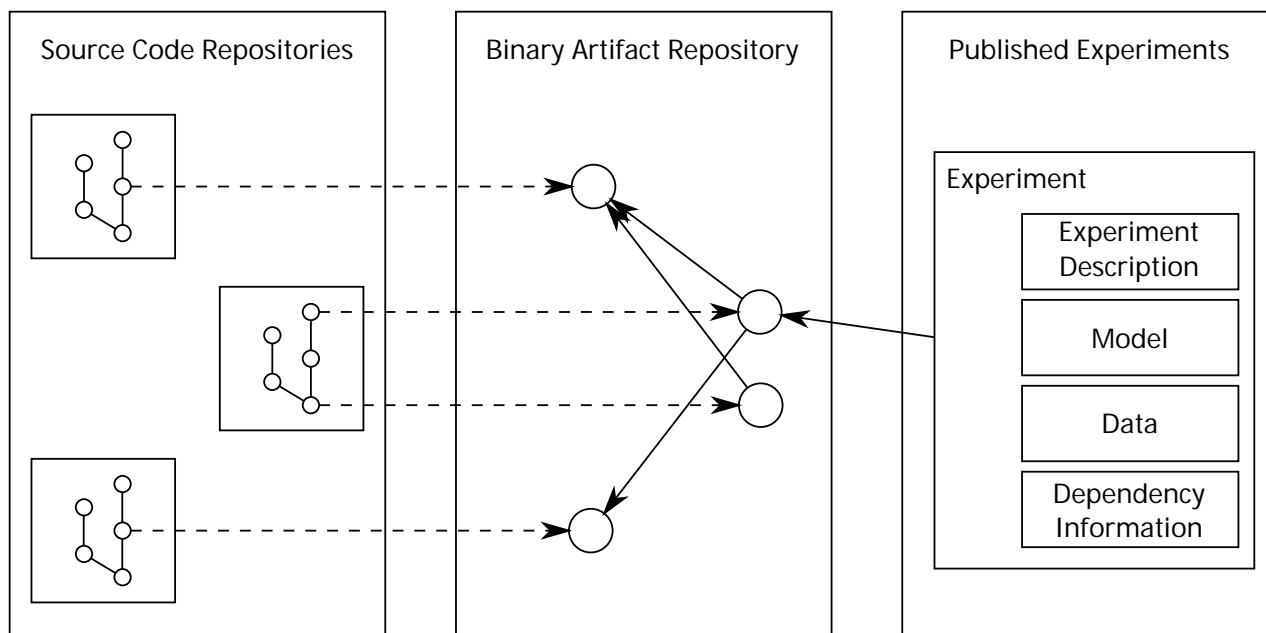


Figure 1: The three components of making simulation experiments reproducible and credible.

methods for modeling and simulation, while also supporting domain scientists in conducting simulation studies in their fields. Thus, we are constantly extending existing software components as well as developing new ones. Our current active software projects are written in Java, Scala, and C++, and comprise several thousand lines of code. The requirements from the previous section are implemented by combining the software components with an appropriate infrastructure.

Three aspects can be distinguished: the development of simulation software as open source projects, the distribution of the software in versioned binary artifacts, and its application in concrete published simulation experiments. Figure 1 gives an overview of the three aspects. The simulation software is developed in public open source repositories (left), which allows everyone to inspect and retrace the history of the software. From specific versions of the source code, persistent binary artifacts are produced and stored in a public repository (center). Those binary artifacts maintain information about the dependencies among them. Simulation experiments (right) are published as downloadable packages including data and models, enabling everyone to run simulation experiments and reproduce their results. Published simulation experiments also contain information on their dependencies on binary artifacts, which can in turn depend on other binary artifacts. This makes

it possible to collect all transitive dependencies to execute a simulation experiment.

2.1 Open source simulation software

One of the cornerstones of reproducible simulation research is that the used software is available and accessible [4]. Whereas the reproducibility of specific simulation results requires a specific version of the software, inspecting the history of the software can be valuable as well. Moreover, providing access to software versions that were developed after the initial publication, as for instance bugfixes or performance improvements, allow other researchers to utilize the changes for their simulation experiments. By maintaining a public source code repository with version control, a full log of who made what changes at which time is publicly available. This increases the trustworthiness of the software and, in turn, the results produced by the software.

We are maintaining a number of public open source simulation software projects in institutional repositories (<https://git.informatik.uni-rostock.de/mosi>). Even if no institutional infrastructure is available, hosting code for a project is straightforward due to the availability of repository hosting services, such as GitHub (<https://github.com>), GitLab (<https://gitlab.com>), or Bitbucket (<https://bitbucket.org>). These services offer free

hosting for open source projects, small teams, and/or academic research.

To further increase the trustworthiness of an open source simulation software project, we follow best practices for state-of-the-art software projects. A continuous integration service makes sure that the public code always passes all defined unit and integration tests in an isolated environment using Docker containers. We use Git for version control, which facilitates structuring the history of the project, making bugfixes or the development of new features explicit. This allows for easily retracing the development of the project over time. As a distributed version control system, Git also decreases the dependency on a single repository service.

A related topic that we can only shortly discuss in this article is the licensing of published code [12]. We use the term “open source” to refer to software that is published for other people to freely use and modify. However, the plethora of different licenses for open source software as well as competing terms such as “free software” or “libre software” clearly show that this is a non-trivial issue. Resources such as <https://choosealicense.com> can help to make the choice for a specific license. Our software is licensed under the Apache License Version 2.0, which is a permissive license that also allows the commercial use of the software. This way, users of our software are constrained in how they use the software in the minimal reasonable way.

2.2 Build tools and dependency management

Whereas public, version-controlled source code repositories enable inspecting and retracing the ongoing and past development of software, reproducing specific results requires the exact versions of the software that yielded the results. The versions of the software must be kept available and it must be executable for the public. This is challenging because of the diversity of operating systems and build and run environments (compilers, interpreters, virtual machines) and particularly because all dependencies of the software must be available in the originally used version as well. Consequently, simulation results are often hard to reproduce even if the software used to obtain them is available as source code.

Our software exploits Apache Maven (<https://maven.apache.org>) to manage dependencies and build and run the software. Maven is a standard tool for software management in the domain of Java and other programming languages that run

on the Java Virtual Machine (JVM). By design, JVM applications are platform-independent and can run on any computer with an installed JVM. Maven itself is a JVM application that, based on a project-specific configuration file, is able to automatically resolve and download (transitive) dependencies in the correct versions from on-line repositories, compile the software, and run it. The default on-line repository is Maven Central (<https://search.maven.org>), which as of April 2018 hosts more than 230,000 unique artifacts.

We publish binary artifacts of our simulation software to Maven Central via the Open Source Software Repository Hosting (<http://central.sonatype.org/pages/ossrh-guide.html>) program. To be allowed to publish artifacts this way, the software must satisfy a few requirements. Most importantly, the source code of the software must be attached. This way, Maven users can not only use the artifacts to build their own software, but also inspect the source code of the version they are using.

2.3 Artifact persistence

The simulation software itself can be persisted in on-line software artifact repositories such as Maven Central (<https://search.maven.org>). However, individual simulation models, simulation experiments, or data sets for simulation should be persisted differently. Whereas the software is necessary to execute simulations, the models, experiments, and data contain knowledge and insights from the application domain of a particular simulation study. Often they are also connected to a particular publication. Thus, they should be documented and made accessible in a way that resembles scientific publication rather than software (see [13] for a detailed discussion on software citation).

Some research areas have established databases to collect simulation models. For example, the BioModels Database hosts hundreds of curated systems biology models [10]. Similarly, the OpenABM Computational Model Library contains documented agent-based models [14]. Those model repositories assign persistent identifiers to each model, making it possible to reference them.

So far, no standard repositories for simulation experiments exist. In principle, similar methods as for models can be used to persist simulation experiments. Running persisted simulation experiments, however, is non-trivial, for the reasons described in the previous section. Using our tool Simulation Experiment Speci-

fication on a Scala Layer (SESSL) [15], which wraps a domain-specific language for specifying and executing simulation experiments, complete reproducible simulation experiments can be archived. The SESSL experiment specification contains all salient aspects of experiment execution, such as simulation algorithm to use, stop and replication conditions, or what simulation output to record and save. SESSL experiments are Maven projects, and Maven automatically downloads all required dependencies, compiles the code to an executable form, and runs the simulation experiment.

We provide a quickstart project (<https://git.informatik.uni-rostock.de/mosi/sessl-quickstart>) that can serve as a starting point for defining SESSL experiments. The quickstart project uses the Maven wrapper (<https://github.com/takari/maven-wrapper>) and contains start scripts for Windows and Unix systems. This way, users can start experiments with a single click without needing to install Maven first (or even know about Maven working in the background).

3 Next steps

First steps to improve the reproducibility of research in the computational sciences have already been taken. The development towards open science has sparked different initiatives, such as the model databases in specific simulation-based research domains. These encourage publishing models, enforce documentation standards, enable peer-reviewing models, and allow referencing models via unique identifiers. To further improve the reproducibility of simulation-based research, similar concepts should be applied to simulation experiments. As for models, this requires declarative descriptions. Whether experiment descriptions can be standardized similar to model description is questionable, as a method to describe simulation experiments must be very powerful to allow for the large diversity of simulation experiments. Nevertheless, public archives of executable simulation experiments, allowing to reference them via unique identifiers, will be crucial for reproducible, trustworthy, open research.

With artifacts such as models, experiments, data, and software made explicit, knowledge about the relations and interdependencies of those artifacts becomes valuable as well. Capturing provenance information for simulation studies can help to structure this knowledge by relating artifacts and processes within and across

simulation studies [16]. This way, not only the individual components of a simulation study can be made available, but also their connections to other research results, implementing the paradigm of open science.

The deployment of software updates to end users needs careful consideration. On the one hand, the results of a simulation experiment might only be reproducible with specific versions of software (for example when comparing the runtime performance of different simulation tools). On the other hand, receiving bugfixes and performance improvements is important or at least beneficial for end users. If an experimental result can not be reproduced after a bugfix, the corresponding published results might be flawed as well. Currently, our SESSL experiments allow users to manually adapt the version of the experiments' dependencies via Maven. A more sophisticated support for managing and updating the dependencies would arguably require a migration away from Maven, giving up its advantages. In particular, with more dependencies the risk of conflicts between transitive dependencies increases (this problem is sometimes called "dependency hell"). This limits the flexibility of changing the version of dependencies, for example to use an updated version of a simulation algorithm. Modern package managers such as Nix (<https://nixos.org/nix>) might be an interesting alternative.

4 Conclusions

Based on an analysis of requirements for software for simulation-based research, this article shows

- how a continuous, open pipeline from simulation software development via deployment to application by the end user (applying best practices and industry standards) can be implemented, and
- how persistent, reproducible, platform-independent simulation experiment artifacts can be provided.

The deployment of our software and the automatic management of dependencies in simulation experiments requires that all software is JVM-compatible. For other ecosystems that need a platform-specific compiler, for example C/C++-based software, other methods to distribute and run simulation experiments are necessary. One alternative approach is the use of Docker containers [17].

Currently, we package simulation experiments with the models and data to run the experiment. In the future, it would be desirable to have experiments, models, and data persisted in databases or repositories with a unique identifier. A simulation experiment would then refer to these artifacts via such identifiers, making the relations between them more explicit. Scientific publications could in turn reference the experiment, which would make obtaining the experiment and reproducing its results straightforward. This way, simulation experiments will become first-class citizens in the world of open science.

Acknowledgement

This research was supported by the DFG (German Science Foundation) Collaborative Research Centre 1270 ELAINE.

References

- [1] Fang FC, Steen RG, Casadevall A. Misconduct accounts for the majority of retracted scientific publications. *Proceedings of the National Academy of Sciences*. 2012;109(42):17028–17033.
- [2] Casadevall A, Steen RG, Fang FC. Sources of error in the retracted scientific literature. *The FASEB Journal*. 2014;28(9):3847–3855.
- [3] Boisvert RF. Incentivizing Reproducibility. *Commun ACM*. 2016;59(10):5–5.
- [4] Stodden V, Miguez S. Best practices for computational science: Software infrastructure and environments for reproducible and extensible research. *Journal of Open Research Software*. 2013;2(1).
- [5] Taylor SJE, Anagnostou A, Fabiyi A, Currie C, Monks T, Barbera R, Becker B. Open science: Approaches and benefits for modeling simulation. In: *2017 Winter Simulation Conference (WSC)*. 2017; pp. 535–549.
- [6] Fehr J, Heiland J, Himpe C, Saak J. Best practices for replicability, reproducibility and reusability of computer-based experiments exemplified by model reduction software. *AIMS Mathematics*. 2016; 1(3):261–281.
- [7] Peng RD. Reproducible Research in Computational Science. *Science*. 2011;334(6060):1226–1227.
- [8] Dalle O, Ribault J, Himmelspach J. Design considerations for M&S software. In: *Winter Simulation Conference*. Winter Simulation Conference. 2009; pp. 944–955.
- [9] Himmelspach J. Toward a collection of principles, techniques, and elements of modeling and simulation software. In: *Advances in System Simulation, 2009. SIMUL'09. First International Conference on*. IEEE. 2009; pp. 56–61.
- [10] Le Novère N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B, Snoep JL, Hucka M. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Research*. 2006; 34(suppl_1):D689–D691.
- [11] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo CT, Finkers R, Gonzalez-Beltran A, Gray AJ, Groth P, Goble C, Grethe JS, Heringa J, 't Hoen PA, Hoofstede R, Kuhn T, Kok R, Kok J, Lusher SJ, Martone ME, Mons A, Packer AL, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone SA, Schultes E, Sengstag T, Slater T, Strawn G, Swertz MA, Thompson M, van der Lei J, van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J, Mons B. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*. 2016;3:160018.
- [12] González AG. Open science: open source licenses in scientific research. *North Carolina Journal of Law and Technology*. 2006;7(2):321–366.
- [13] Smith AM, Katz DS, Niemeyer KEa. Software citation principles. *PeerJ Computer Science*. 2016;2:e86.
- [14] Rollins ND, Barton CM, Bergin S, Janssen MA, Lee A. A computational model library for publishing model documentation and code. *Environmental Modelling & Software*. 2014;61:59–64.
- [15] Ewald R, Uhrmacher AM. SESSL: A domain-specific language for simulation experiments. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*. 2014;24(2):11.
- [16] Ruschinski A, Uhrmacher AM. Provenance in Modeling and Simulation Studies - Bridging Gaps. In: *Winter Simulation Conference (WSC 2017)*. Electronic ISSN: 1558-4305: IEEE. 2017; pp. 872–883.
- [17] Boettiger C. An Introduction to Docker for Reproducible Research. *SIGOPS Oper Syst Rev*. 2015; 49(1):71–79.

Generierung von Steuerungen für Gelenkarmroboter mit simulationsbasiertem Reinforcement-Learning

Georg Kunert, Thorsten Pawletta

Hochschule Wismar – University of Applied Science, Forschungsgruppe CEA
georg.kunert@cea-wismar.de

Abstract. Es wird untersucht, wie mit Hilfe des Q-Learning-Verfahrens, einem speziellen Reinforcement-Learning-Ansatz, simulationsbasiert eine Robotersteuerung für eine Pick-und-Place-Anwendung erlernt werden kann. Weiterhin wird ein Postoptimierungsansatz zur Verbesserung einer erlernten Strategie vorgestellt. Anschließend wird gezeigt, wie die postoptimierte Strategie unter Nutzung des Simulation-Based-Control-Ansatzes automatisiert in eine ausführbare Steuerung transformiert werden kann.

Einleitung

Die konventionelle roboterorientierte Programmierung ist auf einen speziellen Robotertyp und ein explizit zu lösendes Problem fokussiert. Die Planung der Bewegungen und Aktionen sind genau auf diesen Typ und das Problem ausgerichtet und es wird ein maßgeschneidertes Programm entwickelt. Typische Programmieransätze sind nach [1]: (i) die offline Programmierung mit einer roboterspezifischen Programmiersprache (RP), (ii) Teach-In Verfahren, (iii) Master-Slave Verfahren und (iv) Playback Verfahren. Basierend auf den unterschiedlichen RP existieren herstellerspezifische und herstellerübergreifende Computer-Aided-Robotics (CAR) Systeme, wie z.B. KUKA.Sim oder EASY-ROB. Diese unterstützen eine offline Steuerungsentwicklung in einer virtuellen Umgebung.

Unter Leitung des „Robot Operating System“ (ROS) Konsortiums erfolgte mit ROS-Industrial [2] die Spezifizierung eines Schichtenmodells und eines Interface-Standards für Roboter-Controller. ROS-Industrial ist eine herstellerübergreifende Open-Source Entwicklung, basierend auf dem Master-Slave Verfahren. Das Schichtenmodell unterstützt die Integration mit CAR-Systemen und die Anwendung unterschiedlicher abstrakter Programmierverfahren.

Ein in der Robotik etablierter Abstraktionsansatz ist

die aufgabenorientierte Programmierung. Es werden Aufgaben spezifiziert, die nicht mehr robotertypabhängig sind. Zur Ausführung einer aufgabenbasierten Steuerung wird ein Aufgabentransformator benötigt.

Bereits vor ROS-Industrial startete die Forschungsgruppe „Computational Engineering and Automation“ (CEA) der Hochschule Wismar die Entwicklung der freien „Robotic Control and Visualization“ (RCV) Toolbox für MATLAB¹ [3]. Sie stellt eine Menge abstrahierter Befehle in Analogie zu einer Robotersprache auf MATLAB-Ebene zur Verfügung und die Steuerungsentwicklung kann in einer virtuellen Umgebung erfolgen. Weiterhin wurde von der Forschungsgruppe CEA ein Framework, der Simulation-Based-Control (SBC) Ansatz [4], entwickelt. Mit dem SBC-Ansatz können in Verbindung mit ROS-Industrial oder der RCV-Toolbox Robotersteuerungen robotertypunabhängig, aufgabenorientiert und modellbasiert entwickelt werden.

In diesem Beitrag wird der Einsatz eines maschinellen Lernverfahrens im Kontext mit dem SBC-Ansatz zur Robotersteuerungsentwicklung untersucht. Mit Hilfe eines Reinforcement-Learning (RL) Verfahrens soll eine aufgabenorientierte Robotersteuerung für ein Montageproblem simulationsbasiert offline erlernt und unter Nutzung des SBC-Ansatzes in eine ausführbare Steuerung transformiert werden.

Zu Beginn werden die Grundlagen des RL und ein spezifischer Ansatz des Q-Learnings nach [5,6] kurz eingeführt und an einem Anwendungsbeispiel dargestellt. Danach wird am Anwendungsbeispiel gezeigt, wie eine erlernte Verhaltensstrategie durch eine Postoptimierung verbessert werden kann. Anschließend wird gezeigt, wie unter Nutzung des SBC-Ansatzes eine erlernte und postoptimierte Strategie automatisch in eine ausführbare Robotersteuerung transformiert werden kann. Abschließend

¹ RCV Tbx., www.cea-wismar.de, Stand 05/2018

wird ein Ausblick auf zukünftige Arbeiten gegeben, die zum Ziel haben, ein RL-Verfahren auf eine Klasse von Montageproblemen anzuwenden.

1 Reinforcement Learning

Das RL bildet nach [6] neben dem überwachten Lernen und unüberwachten Lernen eine eigenständige Gruppe im Bereich der maschinellen Lernverfahren.

Das Ziel des RL ist es, eine Verhaltensstrategie zu erlernen und anschließend auf deren Basis eine optimale Folge von Aktionen vorzuschlagen, um einen vorgegebenen Zielzustand zu erreichen. Der Vorteil dieses Lernansatzes ist, dass keine Vorkenntnisse in Form von Trainingsdaten und kein übergeordneter Lehrer benötigt werden. Ein RL-Verfahren trainiert sich auf Basis des Trial-and-Error-Prinzips selbst.

1.1 Grundlagen des RL

Abbildung 1 zeigt die Grundstruktur des RL. Das RL beinhaltet eine Umgebung und mindestens einen Agent.

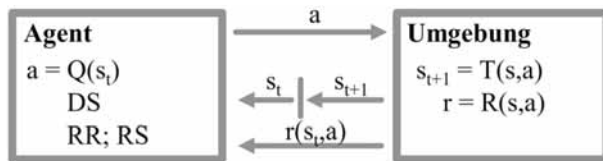


Abbildung 1: Grundstruktur des RL

Die Umgebung kann eine endliche Menge an Zuständen $s \in S$ annehmen. Die Umgebung kommuniziert ihren Anfangszustand s_t , auf den der Agent mit einer Aktion a reagiert. Die Umgebung sendet auf Grundlage der Aktion a eine Belohnung oder Bestrafung r und wechselt in einen Folgezustand s_{t+1} . Die Belohnung r ist ein Feedback der Umgebung auf die Aktion a . Dieses Wechselspiel zwischen Agent und Umgebung wird wiederholt, bis die Umgebung einen erfolgreichen oder nicht erfolgreichen Zielzustand $s_{\text{ziel}} \in S$ erreicht hat. Die Iteration bis zum Erreichen eines Zielzustandes wird im RL als Episode bezeichnet.

Die Umgebung generiert bei Markow-Prozessen erster Ordnung die Belohnungen r nach Gleichung 1 und berechnet den Folgezustand $s_{t+1} \in S$ nach Gleichung 2.

$$r = R(s_t, a) \quad (1)$$

$$s_{t+1} = T(s_t, a) \quad (2)$$

Das Belohnungsmodell R und das Zustandsübergangsmodell T sind Funktionen des aktuellen Zustands s_t der Umgebung und der aktuellen Aktion a des Agenten. Der Folgezustand kann identisch dem letzten Zustand sein. Die Belohnung r ist ein skalarer Wert. Oft wird durch die Umgebung mit Belohnungen $r \in \{-\infty, -1, 0, +1\}$ gearbeitet. Eine negativ unendliche Belohnung bedeutet, dass die Aktion a des Agenten nicht erlaubt war. Eine minus Eins bedeutet, dass die Umgebung einen Zielzustand erreicht hat, dieser aber nicht als Erfolg zu werten ist. Bei einer plus Eins ist ein erfolgreicher Zielzustand erreicht. Aktionen, die mit einer Null bewertet werden, sind erlaubt, führen aber nicht zu einem Zielzustand. Diese Zustände können als Zwischenzustände zum Erreichen eines Ziels interpretiert werden. Der Agent speichert alle von der Umgebung erhaltenen Zustände $s \in S$ in einer Datenstruktur DS (Abb. 5) und sammelt damit Wissen über die Umgebung. Dadurch können bereits erforschte Zustände identifiziert werden. Die in Abbildung 1 eingeführte Bezeichnung Q wird im Abschnitt 1.2 erläutert, RR und RS im Abschnitt 3.

Die Trainingsphase des RL besteht in der Ausführung einer Vielzahl von Episoden. Anfangs exploriert der Agent die Umgebung ausschließlich nach dem Trial-und-Error-Prinzip. Auf Basis des gesammelten Wissens werden in späteren Episoden, unter Verwendung eines Lernverfahrens, die Erfolge von Aktionen vom Agenten bewertet. Ein häufig eingesetztes Lernverfahren ist das Q-Learning. Ziel ist es, mit möglichst wenigen Zustandswechseln der Umgebung zu einem erfolgreichen Zielzustand zu gelangen. Das Ergebnis der Trainingsphase ist eine erlernte Verhaltensstrategie π des Agenten. Gemäß Gleichung 3, ist π eine Funktion, die auf Basis des aktuellen Zustands der Umgebung eine zielorientierte Aktion a zurückgibt.

$$a = \pi(s_t) \quad (3)$$

Während des Lernens befindet sich der Agent im sogenannten „Exploration-Exploitation-Dilemma“. Exploitation bezeichnet das Lernen auf Basis bereits erworbenen Wissens und Exploration das Erforschen unbekannter Strategien durch Ausprobieren neuer Aktionen.

Für ein erfolgreiches iteratives Lernen benötigen RL-Verfahren oft sehr viele Episoden. Die Rechenzeit hängt maßgeblich von der Komplexität des Zustandsraums und der Reaktionszeit der Umgebung ab. Aufgrund der Reaktionszeit ist bei Robotikanwendungen die Möglichkeit des Lernens an realen Umgebungen in der Regel nicht

praktikabel. Daher wird das Verhalten der Umgebung wird auf Basis eines Modells simuliert und das Lernen erfolgt offline. Alternativ kann mit dem simulierten Verhaltensmodell der Umgebung auch proaktiv während des Steuerungsbetriebs gelernt werden.

1.2 Q-Learning

Wie eingangs beschrieben ist das Ziel des RL das Erlernen einer möglichst optimalen Verhaltensstrategie π nach Gleichung 3, um – in der einfachsten Form – einen vorgegebenen Zielzustand zu erreichen. Das Q-Learning ist eine spezielle Ausprägung des RL zur Lösung des Optimierungsproblems. Es basiert auf der von [7] eingeführten Idee des Lernens auf Grundlage einer „memory matrix“ $W(a, s_t)$, beim Q-Learning Q-Matrix genannt. Die Methode des Q-Learnings existiert in verschiedenen Ausprägungen [5,6]. Nachfolgend wird der Ansatz des „value based learnings“ (VBL) kurz eingeführt, der nach [6] einen speziellen Ansatz des Q-Learnings darstellt

Beim VBL besitzt der Agent vorab kein internes Modell der Umgebung. Er erforscht anfangs mit rein zufälligen Aktionen $a \in A$ die Zustände $s \in S$ sowie Belohnungen r der Umgebung und erlernt auf Basis dieser Informationen iterativ sein internes Modell der Umgebung, die sogenannte Q-Funktion $Q(s_t, a)$. Die Q-Funktion beschreibt den zu erwartenden Nutzen, den Q-Wert, einer Aktion a im Zustand s_t . Die Q-Funktion wird im Agenten mit einer dynamisch wachsenden Matrix, genannt Q-Matrix, abgebildet. Jede Zeile der Q-Matrix repräsentiert einen erkundeten Zustand $s \in S$ der Umgebung und jede Spalte eine mögliche Aktion $a \in A$ des Agenten. Die Q-Matrix speichert somit das iterativ erlernte Wissen des Agenten.

Zum Beginn der Trainingsphase, d.h. beim Start der ersten Episode, ist die Q-Matrix eine $0 \times n$ Matrix. Die Spaltendimension n entspricht der Anzahl möglicher Aktionen. Für jeden neu erforschten Zustand $s \in S$ wird Q um eine Zeile erweitert und alle Elemente der neuen Zeile mit 0 initialisiert. Die Berechnung des Q-Wertes eines aktuellen Aktionspaares (s_t, a) kann unterschiedlich erfolgen. Gleichung 4 zeigt den einfachsten Ansatz.

$$Q(s_t, a) = r(s_t, a) + \gamma \cdot \max(Q(s_{t+1}, \forall a \in A)) \quad (4)$$

Wie im vorherigen Unterabschnitt beschrieben, antwortet die Umgebung dem Agenten auf eine Aktion a mit einem Folgezustand s_{t+1} und einer Belohnung r gemäß der Gleichungen 1 und 2. Der Nutzen $Q(s_t, a)$ ergibt sich nach Gleichung 4 aus der Summe der aktuellen Belohnung

des Zustands-Aktionspaares (s_t, a) und dem gewichteten maximalen Nutzen einer möglichen Folgeaktion. Letzterer wird aus der bisher aufgebauten Q-Matrix bestimmt. Dazu werden alle Spalten $\forall a \in A$ in der Zeile des Folgezustandes s_{t+1} der Q-Matrix analysiert. Der Gewichtungsfaktor γ wird als Diskontierungsparameter bezeichnet. Er liegt zwischen 0 und 1 und steuert den Einfluss zu erwartender zukünftiger Belohnungen auf die aktuelle Entscheidung.

Erweiterte Q-Learning-Verfahren führen zusätzlich einen Lernratenparameter α ein. Sie besitzen die Möglichkeit der Gewichtung zwischen bereits erlernten Nutzenwert $Q(s_t, a)$, der aktuellen Belohnung und der zu erwartender zukünftiger Belohnungen [5,8].

Unabhängig vom konkreten Q-Learning-Verfahren wird in der Trainingsphase eine Vielzahl von Episoden ausgeführt. Es wird mit einem zufälligen oder bekannten Anfangszustand $s \in S$ der Umgebung gestartet. Anschließend werden in jeder Episode zufällig („Exploration“) oder auf Basis bereits bekannter Q-Werte („Exploitation“) Aktionen $a \in A$ an der Umgebung ausprobiert, um das interne Modell des Agenten in Form der Q-Matrix zu verbessern. Eine Episode endet, wenn ein Zielzustand der Umgebung erreicht wurde. Dies erkennt der Agent am zurückgesendeten Belohnungswert. Mit einem zusätzlichen Parameter kann eine Exploration- Exploitation- Gewichtung erfolgen, so dass zum Beispiel mit zunehmender Anzahl an Episoden der Explorationanteil abnimmt.

Nach der Trainingsphase repräsentiert die Q-Matrix die erlernte Verhaltensstrategie π . Im Falle des Q-Learnings kann Gleichung 3 wie folgt konkretisiert werden.

$$\pi(s_t) = \operatorname{argmax}(Q(s_t, \forall a \in A)) \quad (5)$$

Der Agent bekommt von der Umgebung den Anfangszustand als aktuellen Zustand s_t . Nach Gleichung 5 sucht er in der Q-Matrix in der Zeile des aktuellen Zustands den maximalen Q-Wert und bestimmt das Argument der Spalte. Dieses repräsentiert die beste Aktion a im Zustand s_t . Die Aktion a wird ausgeführt, der darauf empfangene Folgezustand s_{t+1} wird wieder als aktueller Zustand s_t betrachtet und die nächste Aktion nach Gleichung 5 bestimmt. Die Iteration wird fortgesetzt bis der von der Umgebung gesendete Folgezustand s_{t+1} einem Zielzustand entspricht. Der Agent erkennt das Ende der Episode am Belohnungswert r . In diesem Fall bleiben Zeilen der Q-Matrix, die einen Zielzustand abbilden, mit Nullen besetzt, da der Agent keine Aktionen nachfolgend auf einen Zielzustand untersucht.

Der Q-Learning Ansatz basiert auf einem Backpropagation-Algorithmus. Es ist nicht sichergestellt, dass die gelernte Strategie optimal ist, d.h. die kürzeste Zustandssequenz der Umgebung von einem Startzustand zum Zielzustand gefunden wurde. Auf dieses Problem wird im Abschnitt 3 näher eingegangen. Zuvor wird im nächsten Abschnitt das Q-Learning an einem Beispiel verdeutlicht.

2 Anwendungsbeispiel

Als Anwendungsbeispiel wird das Spiel „Türme von Hanoi“ (TvH), eine Einspieleranwendung, gewählt. Diese entspricht einer typischen „Pick & Place“ Applikation in der Gelenkarmrobotik. Die Anwendung von RL auf Mehrspieleranwendungen ist ungleich komplexer. Die Regeln des Spiels sind simpel und werden im Folgenden erläutert.

Es gibt drei Stäbe. Auf einem Stab ist eine beliebige Anzahl von unterschiedlich großen Scheiben aufeinander gestapelt. Sie sind der Größe nach sortiert und beginnen unten mit der Größten. Ziel des Spieles ist es, die Scheiben auf einen anderen Stab umzustapeln, wobei je Spielzug immer nur eine Scheibe bewegt werden darf und nie eine größere auf eine kleinere Scheibe legen darf. Die Ablagemöglichkeiten sind durch die Stäbe definiert.

Im Sinne des RL bilden die Stäbe und Scheiben die Umgebung und der Spieler, später der Roboter, den Agenten. Die Zustände der Umgebung ergeben sich durch die Belegung der Stäbe mit den Scheiben. Die Bewegung von Scheiben sind Aktionen des Agenten, die zu Zustandsübergängen der Umgebung führen. In Abbildung 3 sind alle möglichen Zustände eines Spiels mit zwei Scheiben gezeigt.

Die Spielregeln sind im Zustandsübergangsmodell T der Umgebung abzubilden. Nach Gleichung 2 generiert T den auf eine Aktion a folgenden Zustand s_{t+1} . Wenn eine Aktion a nicht erlaubt war, wird als Folgezustand s_{t+1} durch T der aktuelle Zustand s_t zurückgegeben.

Abbildung 3 zeigt das Zustandsübergangsdiagramm für eine Umgebung mit zwei Scheiben. Die Nummerierung der Zustände entspricht der Nummerierung in Abbildung 2. Weiterhin ist in der Umgebung ein Belohnungsmodell R gemäß Gleichung 1 spezifiziert.

Aus den Spielregeln ergeben sich Belohnungen mit $r \in \{0, 1, -\infty\}$. Die Belohnung $r = 1$ steht für das Erreichen des Zielzustands als s_{t+1} nach Abbildung 2.

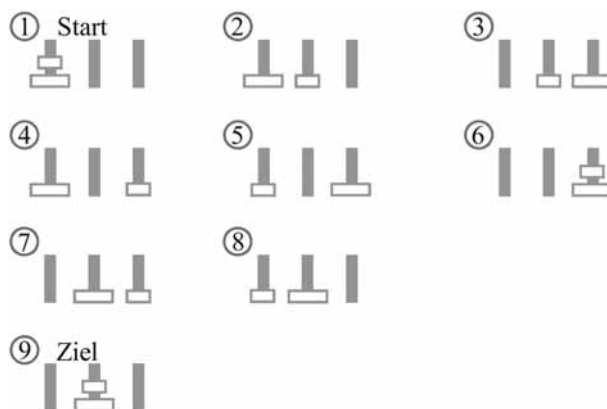


Abbildung 2: Alle Zustände von TvH mit zwei Scheiben.

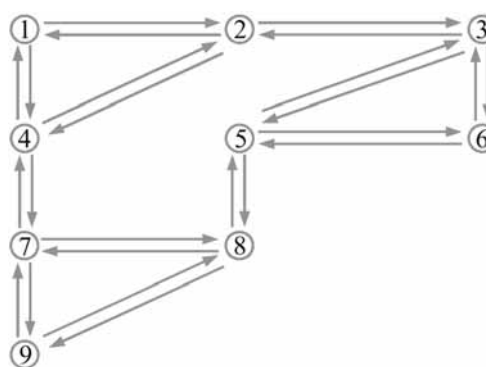


Abbildung 3: Zustandsübergangsdiagramm für TvH mit zwei Scheiben.

Unerlaubte Aktionen a erhalten die Belohnung $r = -\infty$. Die Belohnung $r = 0$ steht für das Erreichen eines Folgezustands s_{t+1} , der kein Zielzustand ist.

Eine Belohnung $r = -1$ wird hier nicht verwendet, da kein Zielzustand vorhanden ist, der als Misserfolg zu werten ist. Das Bewegen einer Scheibe auf einen anderen Stab stellt eine Aktion a dar. Wie in Abbildung 4 gezeigt, hat der Agent bei TvH stets die Auswahl aus sechs Aktionen. Die Beschränkung der Aktionsmenge A auf diese sechs Aktionen impliziert, dass: (i) eine aufgenommene Scheibe nur auf einen der beiden anderen Stäbe abgelegt werden kann und (ii) immer nur eine Scheibe aufgenommen werden darf.

Aus der in Abbildung 4 aufgezeigten Aktionsmenge A folgt, dass die Q-Matrix des Agenten die Spaltendimension sechs besitzt. Die Zeilenanzahl wächst dynamisch. Beim Start der Trainingsphase ist dem Agenten kein Zustand bekannt.

Bei TvH gilt nach Abbildung 2 ein fester Anfangszustand 1. Dieser wird als erster Zustand s_1 dem Agenten von der Umgebung kommuniziert.

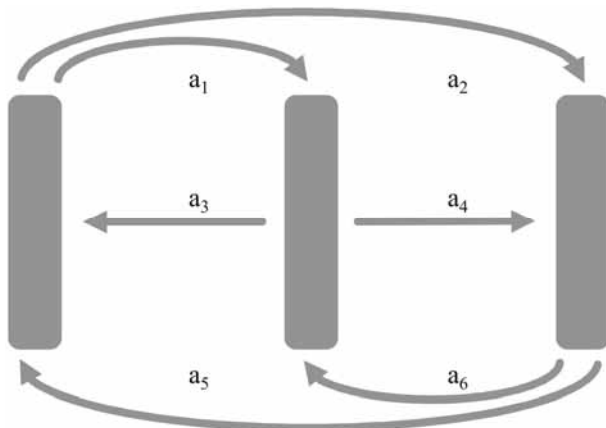


Abbildung 4: Aktionsmenge A des Agenten bei TvH

Jeder Zustand s_t ist charakterisiert durch die Positionen der zwei Scheiben auf den drei Stäben, die mit sechs Attributen abbildbar sind. Die Wertebelegungen der Zustandsattribute werden im Agenten in einer Datenstruktur DS gespeichert, wie in Abbildung 5 dargestellt. Die Attributwerte haben folgende Interpretation: 0 keine Scheibe; 1 kleine Scheibe; 2 große Scheibe. Das erste Element (Attribut) jeder Zeile beschreibt die Existenz beziehungsweise Größe der oberen Scheibe des ersten Stabes und das zweite die Existenz oder Größe der unteren Scheibe des ersten Stabes. Das dritte bis sechste Element jeder Zeile beschreiben analog die Zustände des zweiten und dritten Stabs.



Q-Matrix

	a_1	a_2	a_3	a_4	a_5	a_6
s_1	0	0	0	$-\infty$	0	$-\infty$
s_2	0	0	0	0	0	$-\infty$
s_3	0	0	0	0	0	0

Datenstruktur DS

s_1	1	2	0	0	0	0
s_2	0	2	0	1	0	0
s_3	0	0	2	0	0	1

Abbildung 5: Mögliche Entwicklung der Q-Matrix und der Datenstruktur DS des Agenten beim Start der Trainingsphase

Während der Trainingsphase wird jeder durch die Umgebung kommunizierte Folgezustand $s_{t+1} \in S$ im Agenten analysiert. Ist der Zustand in DS nicht enthalten, wird: (i) der Zustand in DS abgespeichert und (ii) die Q-Matrix um eine Zeile erweitert, in der alle Elemente Null sind. War der Zustand bereits bekannt oder wurde er neu aufgenommen, wird gemäß Gleichung 4 der Nutzen Q des letzten Zustands-Aktionspaares (s_t, a) berechnet. Dazu wird die von der Umgebung kommunizierte Belohnung r

ausgewertet. In diesem Beispiel wurde ein Diskontierungsfaktor von 0,8 gewählt. Wie in Abbildung 5 ersichtlich werden am Anfang der Trainingsphase oft nur neue Zustände aufgenommen und so die Q-Matrix mit Nullelementen aufgebaut. Das iterative Lernen der Nutzenwerte $Q(s_t, a)$ erfolgt in nachfolgenden Episoden.

Abbildung 6 zeigt beispielhaft in unterschiedlichen Episoden berechnete Q-Matrizen. Im Idealfall werden bereits in der ersten Episode alle Zustände S der Umgebung durch den Agenten erforscht. Gemäß Abbildung 6 hat die Q-Matrix dann die Zeilendimension neun. Die Q-Matrix nach Abschluss von Episode 1 zeigt weiterhin, dass das Zustandsaktionspaar (s_8, a_1) mit Nutzenwert = 1 zum Zielzustand führt (s. Abb. 2 und 4). Die Q-Matrix der Episode 2 zeigt einen möglichen Lernfortschritt. Aufgrund des Backpropagation-Algorithmus ist beim TvH die Lernphase abgeschlossen, wenn ein Element der ersten Zeile der Q-Matrix einen Wert größer Null besitzt. Dieser Fall ist in der Q-Matrix der Episode n dargestellt.

Episode 1

	a_1	a_2	a_3	a_4	a_5	a_6
s_1	0	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$
s_2	0	0	0	0	0	$-\infty$
s_3	0	0	0	0	0	$-\infty$
s_4	0	$-\infty$	$-\infty$	0	0	0
s_5	0	0	0	0	0	0
s_6	$-\infty$	0	0	0	0	0
s_7	$-\infty$	0	0	$-\infty$	0	0
s_8	1	0	$-\infty$	0	0	0
s_9	0	0	0	0	0	0

Episode 2

	a_1	a_2	a_3	a_4	a_5	a_6
s_1	0	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$
s_2	$-\infty$	0	0	0	0	$-\infty$
s_3	0	0	0	0	0	$-\infty$
s_4	0	$-\infty$	$-\infty$	0	0	0
s_5	0	0	$-\infty$	0	0	0,8
s_6	$-\infty$	0	$-\infty$	$-\infty$	0	0
s_7	$-\infty$	0	0	$-\infty$	0	0
s_8	1	0	$-\infty$	0	0	$-\infty$
s_9	0	0	0	0	0	0

$$Q(s_8, a_1) = r(s_8, a_1) + \gamma \cdot \max(Q(s_9, \forall a \in A))$$

$$1 = 1 + \gamma \cdot 0$$

$$Q(s_5, a_5) = r(s_5, a_5) + \gamma \cdot \max(Q(s_8, \forall a \in A))$$

$$0,8 = 0 + \gamma \cdot 1$$

Episode n

	a_1	a_2	a_3	a_4	a_5	a_6
s_1	0,41	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$
s_2	$-\infty$	0,51	0	0	$-\infty$	$-\infty$
s_3	$-\infty$	$-\infty$	0,64	0	0	$-\infty$
s_4	0	$-\infty$	$-\infty$	$-\infty$	0	0
s_5	0	0	$-\infty$	$-\infty$	$-\infty$	0,8
s_6	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	0
s_7	$-\infty$	$-\infty$	0	$-\infty$	0	1
s_8	1	0	$-\infty$	0	$-\infty$	$-\infty$
s_9	0	0	0	0	0	0

Abbildung 6: Mögliche Entwicklung der Q-Matrix während der Trainingsphase

Nach der Trainingsphase folgt aus der Q-Matrix gemäß Gleichung 5 die erlernte Verhaltensstrategie π . Das Beispiel in Abbildung 7 zeigt die Generierung von Aktionen auf Basis von π . U steht für die Umgebung und Ag für den Agenten. Das Beispiel bezieht sich auf die Zustände s in Abbildung 2, die Aktionen a in Abbildung 4 und die Q-Matrix der Episode n in Abbildung 6.

```

U:  st = s1
Ag:  π(s1) = a1
U:  st+1 = s2
Ag:  π(s2) = a2
U:  st+1 = s3
Ag:  π(s3) = a3
U:  st+1 = s5
Ag:  π(s5) = a5
U:  st+1 = s8
Ag:  π(s8) = a1
U:  st+1 = s9
Ag:  st+1 == Zielzustand → Abbruch
    
```

Abbildung 7: Beispiel zur Generierung von Aktionen auf Basis von π .

Aus dem Beispiel folgt die Sequenz von Zustands-Aktionspaaren: [(s₁,a₁), (s₂,a₂), (s₃,a₃), (s₅,a₅), (s₈,a₁), (s₉, Abbruch)]. Ein Vergleich mit dem Zustandsübergangsdigramm in Abbildung 3 zeigt, dass kein optimales π erlernt wurde, da nicht der kürzeste „Weg“ vom Start- zum Zielzustand gefunden wurde.

3 Postoptimierung der Verhaltensstrategie

Ziel der Postoptimierung ist die Suche nach einer kürzeren Sequenz von Zustands-Aktionspaaren, die möglichst den minimalen „Weg“ vom Start- zum Zielzustand abbildet. Dazu ist es sinnvoll, während der Trainingsphase, alle von der Umgebung kommunizierten Belohnungen r und Folgezustände s_{t+1} im Agenten zu speichern. Dafür werden im Agenten dynamisch zwei weitere Matrizen RR (Received-Reward) und RS (Received-nextState) aufgebaut (Abb. 1). Abbildung 8 zeigt die RS- und RR-Matrix des betrachteten TvH Beispiels. Deren Struktur entspricht der Q-Matrix, d.h. der Zeilenindex repräsentiert die durch den Agenten erforschten Zustände und die Spalten entsprechen der Aktionsmenge A. Bei RS wird wie im Beispiel von TvH von einem deterministischen Verhalten der Umgebung ausgegangen.

Die Postoptimierung basiert auf einer iterativen Neuberechnung der Q-Matrix unter Nutzung des Wissens in den Matrizen RR und RS. Abbildung 9 zeigt den Algorithmus in Pseudocode und Abbildung 10 die resultierende Q-Matrix nach der ersten, zweiten und letzten (neunten) Iteration.

RR-Matrix

	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆
s ₁	0	0	-∞	-∞	-∞	-∞
s ₂	-∞	0	0	0	-∞	-∞
s ₃	-∞	-∞	0	0	0	-∞
s ₄	0	-∞	-∞	-∞	0	0
s ₅	0	0	-∞	-∞	-∞	0
s ₆	-∞	-∞	-∞	-∞	0	0
s ₇	-∞	-∞	0	-∞	0	1
s ₈	1	0	-∞	0	-∞	-∞
s ₉	0	0	0	0	0	0

RS-Matrix

	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆
s ₁	2	4	1	1	1	1
s ₂	2	3	1	4	2	2
s ₃	3	3	5	6	2	3
s ₄	7	4	4	4	1	2
s ₅	3	6	5	5	5	8
s ₆	6	6	6	6	5	3
s ₇	7	7	4	7	8	9
s ₈	9	7	8	5	8	8
s ₉	0	0	0	0	0	0

Abbildung 8: RR- und RS-Matrix nach der Trainingsphase

Q-Matrix alle Elemente Null setzen

for i = 1 to |S|

for t = 1 to |S|

for a = 1 to |A|

$$Q(s_t, a) = RR(s_t, a) + \gamma \cdot \max(Q(RS(s_t, a), \forall a \in A))$$

end for

end for

end for

Normierung der Q-Matrixwerte zwischen [-∞, 1]

Abbildung 9: Pseudocode der Postoptimierung

Q-Matrix – 1. Iteration

	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆
s ₁	0	0	-∞	-∞	-∞	-∞
s ₂	-∞	0	0	0	-∞	-∞
s ₃	-∞	-∞	0	0	0	-∞
s ₄	0	-∞	-∞	-∞	0	0
s ₅	0	0	-∞	-∞	-∞	0
s ₆	-∞	-∞	-∞	-∞	0	0
s ₇	-∞	-∞	0	-∞	0	1
s ₈	1	0,8	-∞	0	-∞	-∞
s ₉	0	0	0	0	0	0

Q-Matrix – 2. Iteration

	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆
s ₁	0	0	-∞	-∞	-∞	-∞
s ₂	-∞	0	0	0	-∞	-∞
s ₃	-∞	-∞	0	0	0	-∞
s ₄	0,8	-∞	-∞	-∞	0	0
s ₅	0	0	-∞	-∞	-∞	0,8
s ₆	-∞	-∞	-∞	-∞	0,6	0
s ₇	-∞	-∞	0,6	-∞	0,8	1
s ₈	1	0,8	-∞	0,6	-∞	-∞
s ₉	0	0	0	0	0	0

Q-Matrix – 9. Iteration

	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆
s ₁	0,51	0,6	-∞	-∞	-∞	-∞
s ₂	-∞	0,5	0,51	0,6	-∞	-∞
s ₃	-∞	-∞	0,64	0,5	0,5	-∞
s ₄	0,8	-∞	-∞	-∞	0,5	0,51
s ₅	0,51	0,5	-∞	-∞	-∞	0,8
s ₆	-∞	-∞	-∞	-∞	0,6	0,51
s ₇	-∞	-∞	0,64	-∞	0,8	1
s ₈	1	0,8	-∞	0,6	-∞	-∞
s ₉	0	0	0	0	0	0

Abbildung 10: Q-Matrix nach erster, zweiter und neunter Iteration

Zu Beginn der Postoptimierung werden alle Elemente der Q-Matrix auf Null gesetzt. Die Formel zur Neuberechnung der Nutzenwerte in Abbildung 9 entspricht vom Aufbau der Gleichung 4. Die Belohnung eines Zustands-Aktionspaares (s_t, a) ergibt sich aus der RR-Matrix. Der Folgezustand s_{t+1} wird aus der RS-Matrix bestimmt. Die innerste Schleife iteriert über alle Aktionen eines Zustandes und die mittlere Schleife über alle bekannten Zustände. Damit erfolgt eine einmalige Neuberechnung aller Elemente der Q-Matrix. Die äußere Schleife setzt den Backpropagation-Ansatz um. Die Nutzenwerte werden dadurch vom Zielzustand zum Anfangszustand entwickelt, wie in Abbildung 10 auszugsweise gezeigt. Die Wiederholung der Berechnung aller Q-Matrixelemente ergibt sich demzufolge aus der Anzahl bekannter Zustände (Zeilendimension von Q-Matrix). Abschließend kann optional eine Normierung der Q-Matrixwerte erfolgen.

Beim TvH Beispiel wurden während der Trainingsphase alle neun Zustände erforscht. Das Ergebnis der Postoptimierung ist die Q-Matrix nach der neunten Iteration, wie in Abbildung 10 dargestellt. Analog zur Generierung von Aktionen auf Basis der erlernten Verhaltensstrategie π in Abbildung 7, folgt unter Anwendung der postoptimierten Q-Matrix folgende kürzere Sequenz von Zustands-Aktionspaaren:

$[(s_1, a_2), (s_4, a_1), (s_7, a_6), (s_9, \text{Abbruch})]$.

4 Generierung einer Robotersteuerung

Auf Basis einer vorab bestimmten Verhaltensstrategie π soll automatisch eine ausführbare Steuerungsspezifikation generiert werden. Dabei wird auf die Steuerungsstruktur des Simulation-Based-Control (SBC) Ansatzes in [4] aufgebaut, wie in Abbildung 11 gezeigt.

Das Steuerungsproblem wird aufgabenorientiert im Steuerungs- und Prozessmodell abgebildet. Bei einfachen Problemen wie dem TvH Beispiel kann ein separates Prozessmodell entfallen. Das Interface-Modell (IM) realisiert eine robotertypunabhängige Aufgabentransformation und wird mit der „Robotic Control and Visualization (RCV) Toolbox for MATLAB“ [3] umgesetzt. Die robotertypabhängige Schicht bildet gemäß [3] ein Interpreter auf dem Controller des jeweiligen Roboters.

Die TvH Anwendung ist mit drei Aufgaben: (i) *move(eulerK)*, (ii) *pick()* und (iii) *place()* spezifizierbar. Die Aufgabe *move* hat als Parameter einen Vektor mit

den sechs Euler-Koordinaten der anzufahrenden Position. Die anderen beiden Aufgaben dienen zum Aufnehmen und Ablegen einer Scheibe. Die drei Aufgaben sind als wiederverwendbare Softwarekomponenten implementiert. Aufgrund der Einfachheit der Aufgaben wurde das IM als integraler Bestandteil jeder Aufgabe realisiert.

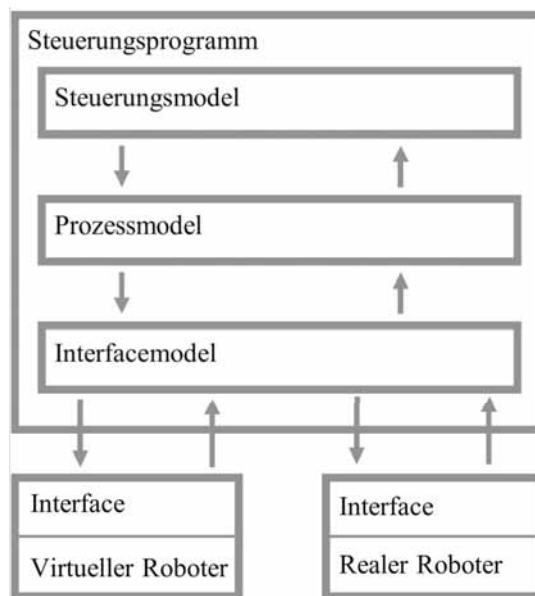


Abbildung 11: Prinzipieller Steuerungsaufbau nach dem SBC-Ansatz

Zur automatischen Generierung der Steuerungsspezifikation muss mit der Verhaltensstrategie π eine Zustands-Aktionsfolge $[(s_t, \exists a \in A), (s_{t+1}, \exists a \in A), \dots, (s_{\text{Ziel}}, \text{Abbruch})]$ berechnet werden. Dies erfolgt wie im Abschnitt 2 (Abb. 7) erläutert unter Verwendung der simulierten Umgebung. Aus der Zustands-Aktionsfolge wird anschließend die aufgabenorientierte Steuerungsspezifikation abgeleitet.

Die TvH Anwendung ist eine typische „Pick & Place“ Anwendung. Jedes Zustands-Aktionspaar $(s_t, \exists a \in A)$ kann durch die gleiche Aufgabenfolge:

$move(EulerK) \rightarrow pick() \rightarrow move(EulerK) \rightarrow place()$

realisiert werden. Der Parameter *EulerK* kodiert die x- und y-Koordinate eines Stabes und die z-Koordinate zum Aufnehmen beziehungsweise Ablegen der Scheibe. Die x- und y-Koordinaten einer Aufgabenfolge ergeben sich aus der auszuführenden Aktion *a* (Abb. 4) und die z-Koordinate aus dem Zustand s_t (Abb. 2). Die Winkelpositionen sind konstant.

Im Abschnitt 3 wurde für zwei Scheiben die postoptimierte Zustands-Aktionsfolge: $[(s_1, a_2), (s_4, a_1), (s_7, a_5), (s_9, \text{Abbruch})]$ berechnet. Bei der Steuerungsgenerierung

wird demgemäß ein Steuerungsmodell mit drei Aufgabenfolgen: $move(\dots) \rightarrow pick() \rightarrow move(\dots) \rightarrow place()$ erzeugt. Zur Ausführung der generierten Steuerung müssen dieser die Koordinaten der Stäbe, die konstanten Winkel und die Parameter der Scheibengeometrie übergeben werden.

Die praktische Umsetzung erfolgte im Labor der Forschungsgruppe CEA mit einem Gelenkarmroboter vom Typ KUKA Agilus, wie in Abbildung 12 gezeigt. Zur technischen Vereinfachung wurden die Scheiben durch nummerierte Würfel ersetzt. Das Lernprogramm wurde für eine beliebige Anzahl an Scheiben implementiert. Auf einem handelsüblichen PC benötigte die Trainingsphase für vier Scheiben zirka zwei Sekunden.

5 Zusammenfassung

Es wurde anhand eines RL-Verfahrens gezeigt, dass eine aufgabenorientierte Steuerung für einen Gelenkarmroboter simulationsbasiert offline erlernt und nach dem SBC-Ansatz in Betrieb genommen werden kann. Weiterhin wurde gezeigt, wie die Lösung eines klassischen Lernverfahrens durch eine Postoptimierung verbessert werden kann. Das betrachtete Anwendungsbeispiel ist eine typische „Pick & Place“ Applikation.



Abbildung 12: Versuchsaufbau im Labor der Forschungsgruppe CEA

In weiteren Arbeiten soll untersucht werden, ob das implementierte RL-Verfahren und die Steuerungsgenerierung direkt auf ähnliche Problemstellungen und damit auf Problemklassen angewendet werden kann. Weiterhin sollen zukünftig Applikationen mit Mensch-Roboter-Kollaboration untersucht werden. Hier kann oft kein vollständiges Simulationsmodell der Umgebung erstellt werden und es muss teilweise online gelernt werden. Es ist zu untersuchen, ob nach der Trainingsphase unter Verwendung eines teilweise simulierten Umgebungsmodells auch proaktiv gelernt werden kann. Des Weiteren sollen im Kontext von Anwendungen mit Mensch-Roboter oder Roboter-Roboter-Kollaboration RL-Verfahren mit mehreren Agenten untersucht werden.

Literatur

- [1] Weber, W. *Industrieroboter – Methoden der Steuerung und Regelung*. Carl Hanser Verlag, München, 2008.
- [2] ROS.org. wiki.ros.org/Industrial, Stand 03/2018
- [3] Deatcu, C., Freymann, B., Schmidt, A., Pawletta, T. MATLAB/Simulink Based Rapid Control Prototyping for Multivendor Robot Applications. *SNE – Simulation News Europe*. 2015; 25(2): 69-78. doi: 10.11128/sne.25.2.1029.
- [4] Freymann, B., Pawletta, S., Schmidt, A., Pawletta, T. Design, Simulation and Optimization of Task-Oriented Multi-Robot Applications with MATLAB/Stateflow. *SNE – Simulation Notes Europe*. 2016; 26(2): 83-90. doi: 10.11128/sne.26.2.1033.
- [5] Sutton, R.S., Barton, A.G. *Reinforcement Learning: An Introduction*. 2nd Edition. Cambridge/MA: MIT Press; 2012. 334 p.
- [6] Akhtar, S.M.F. *Practical Reinforcement Learning*. 1st Edition. Birmingham/UK: Packt Publishing Ltd.; 2017. 320 p.
- [7] Watkins, C.J.C.H. *Learning from Delayed Rewards* Ph.D. thesis, Cambridge Univ. / UK; 1989. 241 p.
- [8] Kramer, O. *Computational Intelligence*. 1st Edition. Berlin, Heidelberg/DE: Springer Pub.; 2009. 158 p.

A Comparison of Simulation Tools for Multibody Models used for Anatomic Joints

Ruth Leskovar^{1*}, Andreas Körner¹, Felix Breiteneker¹

¹Institute for Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; *ruth.leskovar@tuwien.ac.at

This contribution gives an overview of three simulation tools providing multibody modelling libraries. First, the software tools are presented and one case study gives an insight about the implementation process. After, the human knee joint, as biomechanical system, is considered and two multibody models of the knee joint are investigated. This show facilities the softwares offer. Furthermore, the opportunity to embed the multibody model in a closed simulation loop is discussed.

Introduction

This contribution deals with the development of multibody models in various software frameworks. The considered multibody models address biomechanical models for human knee joints.

In general two main approaches exist for modelling biomechanical systems, namely

- Modelling with partial differential equations,
- Multibody modelling.

Their applications are slightly different due to their diverse mathematical description as it is explained in [1]. In biomechanics, partial differential equations are used describing the influence of small motions and movements to calculate resulting strains and stresses in small bone parts of the human body. Multibody modelling is commonly used to describe the kinematics under gross movements and it is possible to simulate the impact of loads in whole bones or even joints. In the following, multibody models are investigated only.

The design and development of mathematical models describing the biomechanical interactions in human joints became important and are an useful tool for the analysis of the interactions taking place during movements and undertaking loads. The human knee joint is an attractive research focus due to its complex structure and vulnerability. Especially, in the last decade, the integration of ligaments and increase of degrees of freedoms in knee joint models dominates the point of interest as it can be seen in [2] and [1].

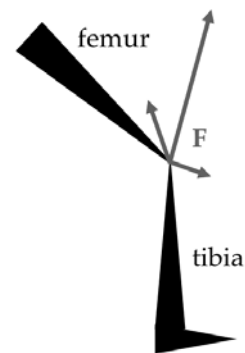


Figure 1: Human leg and force vector acting on the knee joint during walking.

The necessity of biomechanical models covers a wide variety of application fields. Insights derived from joint models are applied in clinical practice, e.g. in therapy of injuries or diseases as it is illustrated in [3]. Even the impact of full knee replacements on the biomechanic interactions can be analysed by multibody models as it is done by [4]. This illustrates the usage of joint models in the development of medical devices. Knee implants, but whole leg prostheses can be enhanced by joint models. In the last years, prostheses improved a lot concerning their features and wearing comfort. This ensures more safety for prosthesis carriers and additionally more flexibility in the daily routine. Due to high requirements on the human leg, three main functions dominate in the research of leg prostheses:

- Stabilisation,
- Carrying loads and weights,
- Dynamic functions, like adapting to different walking scenarios.

The dynamic functions of the human leg allow the adaptation of the human walk to different grounds and various movements, e.g. walking in the wood or climbing stairs. In current research, leg prostheses satisfy this essential need with an included actuator changing the

damping in the prosthesis acting on different sensor signals. The sensor signals measure the acting force on the knee in order to regulate the gait cycle smoothly. Figure 1 illustrates force acting on the human knee joint during walking. More details about microprocessing actuated knees can be read in [5]. This technology is used by OttoBock for example, a well known manufacturer of prostheses [6].

The aim of this work is the design of a feedback simulation loop for multibody models. An environment with powerful control tools facilitates the design of this loop. Therefore, two software frameworks supporting multibody modelling and control tools are presented in order to specify advantages and disadvantages. This requires an already validated multibody model of the human knee joint forming the base for the multibody models in additional simulation environments.

1 Concepts of Multibody Modelling

Multibody modelling is a physical modelling approach. The development of the model is based on the structure and the physical interactions of the investigated system. Hence, it is possible to define a mathematical model proceeding from the mechanical and physical principles without knowing the underlying formulas.

A multibody model consists of bodies, either flexible or rigid, which are linked together by different types of joints. The joints differ by the degree of freedom and the type of movement they allow. This includes a spherical joint containing three degrees of freedom or a revolute joint with one degree of freedom respectively. The properties of the bodies must be defined by size, mass, density and other physical characteristics. The body properties and the interconnections between the bodies formulate the model structure. This allows quick development of physical models on the one hand and analysing complex structures on the other hand. Multibody modelling theory is explained in more detail in [7].

In each multibody model, the equation of motion between bodies is derived by the Newton-Euler equation and the Lagrange formalism. It is given by

$$M\ddot{x} + J_x^T \lambda = F. \quad (1)$$

Hereby, the vector x holds the system coordinates of the system, meaning the variables describing the position of a depicted body in a system. M represents the mass matrix of the system, J_x is the Jacobian of the system coordinates and λ contains the Lagrange multipliers. The right hand side F implies the external force vector on the system. The derivation of the formula and

a more detailed explanation can be seen in [8]. With this equation (1) it is possible to describe the dynamics in a biomechanical system and the resulting kinematics under different types of loads. Furthermore, this gives the possibility deriving underlying equations of complex system.

Having the time-dependence in the equation gives the possibility to consider a multibody model as a dynamic system. This allows to interconnect multibody models easily to system simulation approaches, as e.g. feedback loops which will be discussed later.

2 Simulation tools for Multibody modelling

Various software tools are available providing multibody modelling libraries. In this section three, Simscape, MapleSim and Adams are introduced. One case study, the double pendulum, is implemented in Simscape and MapleSim in order to illustrate common features and distinctions. This illustrates the development of multibody models and facilitates the switch to more complex and biomechanical systems.

Simscape Multibody. Simscape Multibody is an embedded library in the environment of Simulink. This gives the possibility to use the powerful control tools of Simulink directly in models. Furthermore, MATLAB variables and functions can be added. Simscape itself offers more libraries, as e.g. Simscape Driveline or Simscape Electronics. Elements of these libraries can be combined in a multibody model as well.

MapleSim. MapleSim is developed by Maplesoft for physical modelling. The implemented structure is based on Modelica. This gives the possibility to use Modelica code to extend own models. Additionally, having the model structure implemented in Modelica is accompanied by the mathematical description in formulas of the system. MapleSim itself covers various libraries for physical systems including one for multibody elements.

Adams. Adams is a software developed by MSC Software for analysing mechanical systems. The integrated solvers are based on Fortran and C++. It offers a wide variety of interconnections to other softwares, as e.g. other products of MSC Software as the finite element analysis MARC. Besides, Adams gives the possibility of linking to MATLAB and Simulink which makes it easy to use a Adams model in a developed Simulink model. Likewise, the opposite direc-

tion is possible, a Simulink model can be included in Adams.

3 Case Study - The double pendulum

The development of a multibody model starts defining the reference frame for the model. Further, external forces as e.g. the gravity are added. In MapleSim this is realised in the properties of the model, in Simscape blocks exist for these settings. Per default, in both frameworks the gravity is applied.

Giving the double pendulum, two rigid bodies and two revolute joints are considered. The rotational axis of the revolute joint in Simscape is the z-axis. Rotating around another axis requires to add coordinate transforms. In MapleSim, the rotational axis can be chosen freely. Physical properties as the mass must be defined for the rigid bodies. Simscape asks for the shape of the body. MapleSim calculates the geometry depending on the frames which are added to the body. These frames define the distances along each direction.

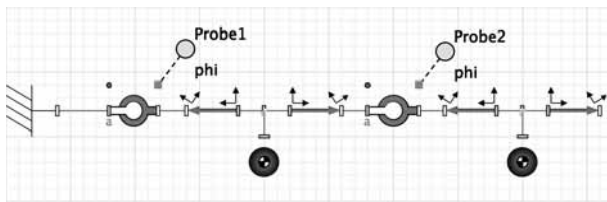


Figure 2: Block diagram of the double pendulum in MapleSim.

In figure 2 and 3 block diagrams of the multibody models for the double pendulum in MapleSim and Simscape respectively, are depicted. Both models contain elements in order to reproduce the same structure.

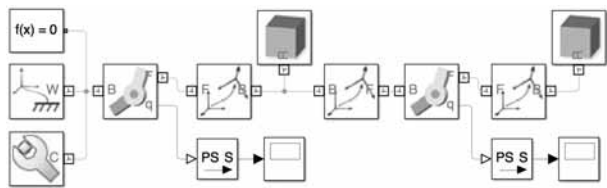


Figure 3: Block diagram of the double pendulum in Simscape.

Figure 4 show the angles of the first revolute joint over time for both simulation frameworks. The results for the simulation are not fitting exactly due to some different definitions for the bodies. The bodies in MapleSim have cylindrical shape. It is possible to determine the

translational offset in each direction but not the shape directly. In Simscape, the shape is easy to determine by choosing different types of geometries. But, choosing a cylindrical shape requires to add more rigid transforms to bring the body in the right direction.

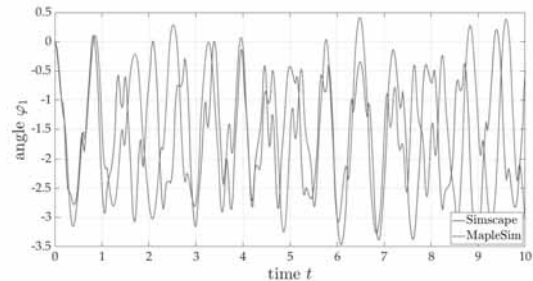


Figure 4: Angle of the first revolute joint in both frameworks.

4 Multibody model for the human knee joint

In order to establish multibody models in various simulation frameworks, an already established and validated model is used as reference for the following models. This model is implemented in Adams and was found on the platform <https://simtk.org>. There, researchers share biomechanical models and corresponding geometry files. This facilitates the development of new models. In the following, a multibody model of the human knee joint is investigated. To fully understand the complex structure of a mathematical model for a joint some basic anatomic knowledge is required.

4.1 Basic anatomics of the human knee joint

The human knee joint is in addition to the shoulder joint the most complex joint in the human body. The complexity comes from the different compositions of components which are interacting there. The human knee contains three main bones, the femur, the tibia and the patella. The composition of these bones is held together by ligaments and tendons. The femur and the tibia are connected by the crucial and the collateral ligaments. The ligaments itself are composed of two fiber bundles, acting on different sites on the bones. The ligaments stabilise the knee reacting on external forces.

The femur and the tibia itself contain cartilage on their surface and between them, the menisci are loaded. They support the damping and optimise the friction. The patella is located in the tendon of the quadriceps. This tendon trends over the knee joint connecting the femur and tibia. The force of the quadriceps is acting

along that tendon and activates each walking scenario in our daily routine.



Figure 5: Anatomical structure of the human knee joint given in [9].

The anatomical structure of the human knee joint can be seen in detail in figure 5 and is explained in [9].

4.2 Adams model of the knee joint

The model which is used for further analysis and for other simulation environments is based on the work of Guess et al. [10], [11], [12]. The model simulates the flexion of the right knee of a 77 year old man. All three main bones, the femur, the tibia and the patella are included. Furthermore, the patellar tendon and the ligaments are incorporated. The geometries of the bones are derived by magnetic resonance images of a cadaver knee. In Adams, the bones are modelled as rigid bodies. The tendon is modelled as a bundle of four linear springs. The position where the tendon is connected to the femur and the tibia are derived from magnetic resonance imaging as well and from testing the cadaver knee in a knee simulator. The stiffness and damping parameter of the tendon are detected from the testing. Modelling the ligaments is more complex because the fibers change their reaction depending on their strain. The strain-stress curve of ligaments shows a non-linear behaviour first, namely the toe region. This is a result from crimping all fibers under stress [10]. When all fibers are stretched, the ligament behaves again as a linear spring with the stiffness parameter k . The force f resulting from this behaviour can be calculated with the formula derived by [13], [14]

$$f(\varepsilon) = \begin{cases} \frac{1}{4\varepsilon_l} k \varepsilon^2, & 0 \leq \varepsilon \leq 2\varepsilon_l \\ k(\varepsilon - \varepsilon_l), & \varepsilon > 2\varepsilon_l \\ 0, & \varepsilon < 0 \end{cases}, \quad (2)$$

whereby the strain ε is calculated by

$$\varepsilon = \frac{l - l_0}{l_0}. \quad (3)$$

The current length of the ligament l is calculated during the simulation directly in Adams. The zero-load length l_0 is the length of the ligament when all fibers become taut. This parameter varies from ligament to ligament. The derivation of this parameter is studied in [11]. The parameter ε_l is an additional spring parameter and is assumed to $\varepsilon_l = 0.03$.

Contact forces between femur and patella and between femur and tibia are implemented following the Hertz contact law

$$F = k\delta^n + B(\delta)\dot{\delta} \quad (4)$$

with the contact force F , a spring constant k , the interpenetration δ and the damping coefficient B [12]. An external force applies a body-moving time-varying force to the tibia in the posterior direction.

The law for the contact force and parameters for the zero-load length of ligaments are studied with calibration of the multibody model with a model described by partial differential equations. This is a common used technique for calibration of new multibody models.

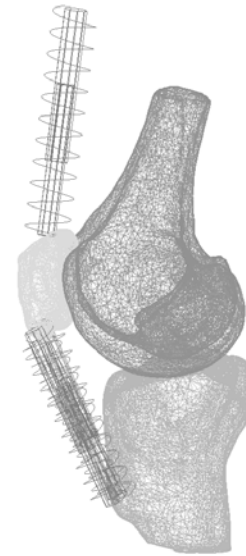


Figure 6: Visualisation of the knee model in Adams with three bones and the patellar tendon.

In figure 6 the multibody model of the knee developed in Adams can be seen. Only the three bones and the tendon is visible due to the fact that the ligaments are implemented as forces. No block diagram is visible during the development of the model. The bodies are visible with their corresponding geometry immediately

after creating them. No joints are included in this model but of course joints are part of the elements available in Adams.

4.3 Simscape model of the knee joint

The knee model designed by Guess et al. forms the starting-point for the build-up of the multibody model in Simscape. All used geometries and physical properties are assumed from Guess et al. as described above. The block diagram of the model can be seen in figure 7. The first three blocks define the solver properties, the world frame represents the global reference frame for the model and further properties as e.g. acting gravity.

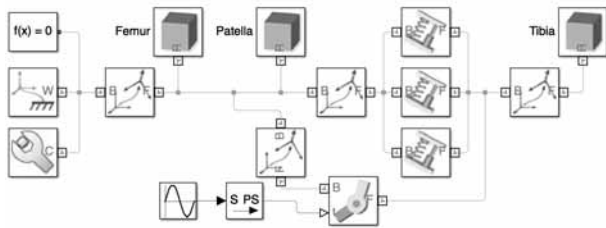


Figure 7: Structure of the knee model in Simscape containing three bones, one joint and the patellar tendon.

The bones are defined by their mass and moments of inertia. The patellar tendon is included by three linear spring and damper forces with determined length of each bundle and parameters for stiffness and damping. The patellar tendon connects the patella and the tibia. The femur and the tibia are connected via one revolute joint. A sine wave is acting on the revolute joint, representing the force on the knee joint during normal walk. For the implementation, it was required to introduce rigid transforms. The first translational transform is needed to erase the offset between the world frame and the femur which occurred from the geometries of the bones. The revolute joint in Simscape has one revolute primitive over the z-axis. To rotate over the x-axis two rotational transforms are used.

The ligaments are missing in the model. Their realisation could be implemented by using a MATLAB function block or non-linear spring elements from the Driveline library in Simscape. Since the non-linear spring elements are 1D, but multibody elements of Simscape are 3D, the spring can only be added as force to joints. Additionally, it is not possible to attach elements on specific points on bodies as it is in Adams. This aggravates the exact implementation of the ligaments. Despite these restrictions, the ligaments will be included in next steps although their effect will be slightly different to their implementation in Adams.

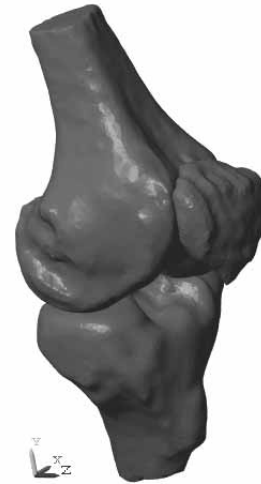


Figure 8: View of the model in the Mechanics Explorer.

The resulting model in the Mechanics Explorer of Simscape can be seen in figure 8.

5 Feedback Simulation Loop for Multibody Models

The build model in Simscape can be easily embedded in a feedback simulation loop. This requires an input u and an output y of the model and an individual behaviour of the system. Therefore, the system can be described as a dynamic system. This presumes that the dynamics of the system can be described by a first order differential equation as in

$$\dot{x} = f(t, u(t), x(t), p), \quad x(0) = x_0, \quad (5)$$

whereby the vector x comprises the states of the system and the vector u represents the input of the system. The change of the states is dependent from the time t , the input, previous states and some parameters p . A block diagram of a dynamical system is depicted in figure 9.

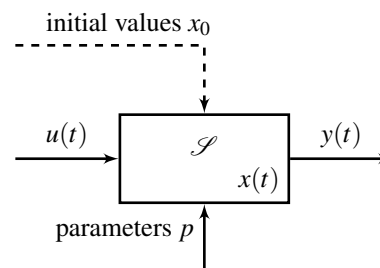


Figure 9: Block diagram for a dynamical system.

Having a multibody model gives the opportunity to describe the motions between the bodies with the second order differential equation 1. For the input vector u an acting force is chosen which is reacting during the human walk. The output of the system is considered as the angle between the femur and the tibia.

Assuming this allows to build up a new model which simulates the human gait cycle. This gives the opportunity to use the knee model in further analysis of the human knee during walk. Further, using different geometries and physical properties gives the possibility to study the behaviour of various knee implants or even knee prostheses in the human gait cycle. This can improve the calibration of the damping actuation. Additionally, more complex movements as walking stairs or climbing can be analysed.

The Simscape model is easily embedded in a control circle due to the Simulink environment. As mentioned above, Adams gives the possibility to export the models as plant and use in Simulink as a subsystem which allows a similar handling. At this point, a PID controller is used, more complex control designs will be investigated in future.

6 Conclusion and Outlook

The implementation of biological systems as multibody models is a quick but valid possibility for analysis of biomechanical interactions. A simplified model is implemented in Simscape which will be extended in further work. The crucial and collateral ligaments will be added and further it will be analysed if an implementation without the revolute joint, similar to the formulation in Adams, is possible. The same procedure will be exercised using MapleSim in order to compare and benchmarking both frameworks in their multibody modelling tools concerning biomechanics. It is clear demonstrated that Adams is a powerful tool in biomechanical modelling.

Nevertheless, concerning this work control tools in all considered frameworks are important to embed the established model in a closed simulation loop. Since the human knee joint displays a non-linear behaviour, non-linear control theory is needed to get satisfying results. Simulink offers a wide variety of powerful tools for control designs which can be used directly in Simscape. Nevertheless, it is not yet clear whether MapleSim is as suitable as simulation tool for this approach. Therefore, multibody models with the same level of complexity will be embedded in an equivalent control circle to study their behaviour in the closed loop. The summary of pros and cons of both frameworks will concern both, the multibody modelling as well the control tools.

References

- [1] Machado M, Flores P, Claro JC, Ambrósio J, Silva M, Completo A, Lankarani HM. Development of a planar multibody model of the human knee joint. *Nonlinear Dynamics*. 2010;60(3):459–478.
- [2] Xu H, Bloswick D, Merryweather A. An improved OpenSim gait model with multiple degrees of freedom knee joint and knee ligaments. *Computer Methods in Biomechanics and Biomedical Engineering*. 2015; 18(11):1217–1224.
- [3] Bersini S, Sansone V, Frigo CA. A dynamic multibody model of the physiological knee to predict internal loads during movement in gravitational field. *Computer Methods in Biomechanics and Biomedical Engineering*. 2016;19(5):571–579.
- [4] Piazza SJ, Delp SL. Three-Dimensional Dynamic Simulation of Total Knee Replacement Motion During a Step-Up Task. *Journal of Biomechanical Engineering*. 2001;123(6):599–606.
- [5] Berry D. Microprocessor Prosthetic Knees. *Physical Medicine and Rehabilitation Clinics of North America*. 2006;17(1):91 – 113. New Advances in Prosthetics and Orthotics.
- [6] OttoBock. Genium. *Tech. rep.*, Otto Bock HealthCare GmbH. 2011.
- [7] Rill G, Schaeffer T. *Grundlagen und Methodik der Mehrkörpersimulation*. Springer Vieweg, 3rd ed. 2017.
- [8] Ambrósio J, Quental C, Pilarczyk B, Folgado J, Monteiro J. Multibody biomechanical models of the upper limb. *Procedia IUTAM*. 2011;2:4 – 17. IUTAM Symposium on Human Body Dynamics.
- [9] Schabus R, Bosina E. *Das Knie - Diagnostik, Therapie und Rehabilitation*. Springer-Verlag Wien New York. 2007.
- [10] Guess TM, Thiagarajan G, Kia M, Mishra M. A subject specific multibody model of the knee with menisci. *Medical Engineering and Physics*. 2010;32(5):505–515.
- [11] Bloemker KH, Guess TM, Maletsky L, Dodd K. Computational Knee Ligament Modeling Using Experimentally Determined Zero-Load Lengths. *The Open Biomedical Engineering Journal*. 2012;6:33–41.
- [12] Guess TM, Liu H, Bhashyam S, Thiagarajan G. A multibody knee model with discrete cartilage prediction of tibio-femoral contact mechanics. *Computer Methods in Biomechanics and Biomedical Engineering*. 2013; 16(3):256–270.
- [13] Wismans J, Veldpaus F, Janssen J, Huson A, Struben P. A three-dimensional mathematical model of the knee-joint. *Journal of Biomechanics*. 1980;13(8).
- [14] Blankevoort L, Huiskes R. Ligament-Bone Interaction in a Three-Dimensional Model of the Knee. *Journal of Biomechanical Engineering*. 1991;113(3):263.

Copyright Information / Regulations ARGESIM

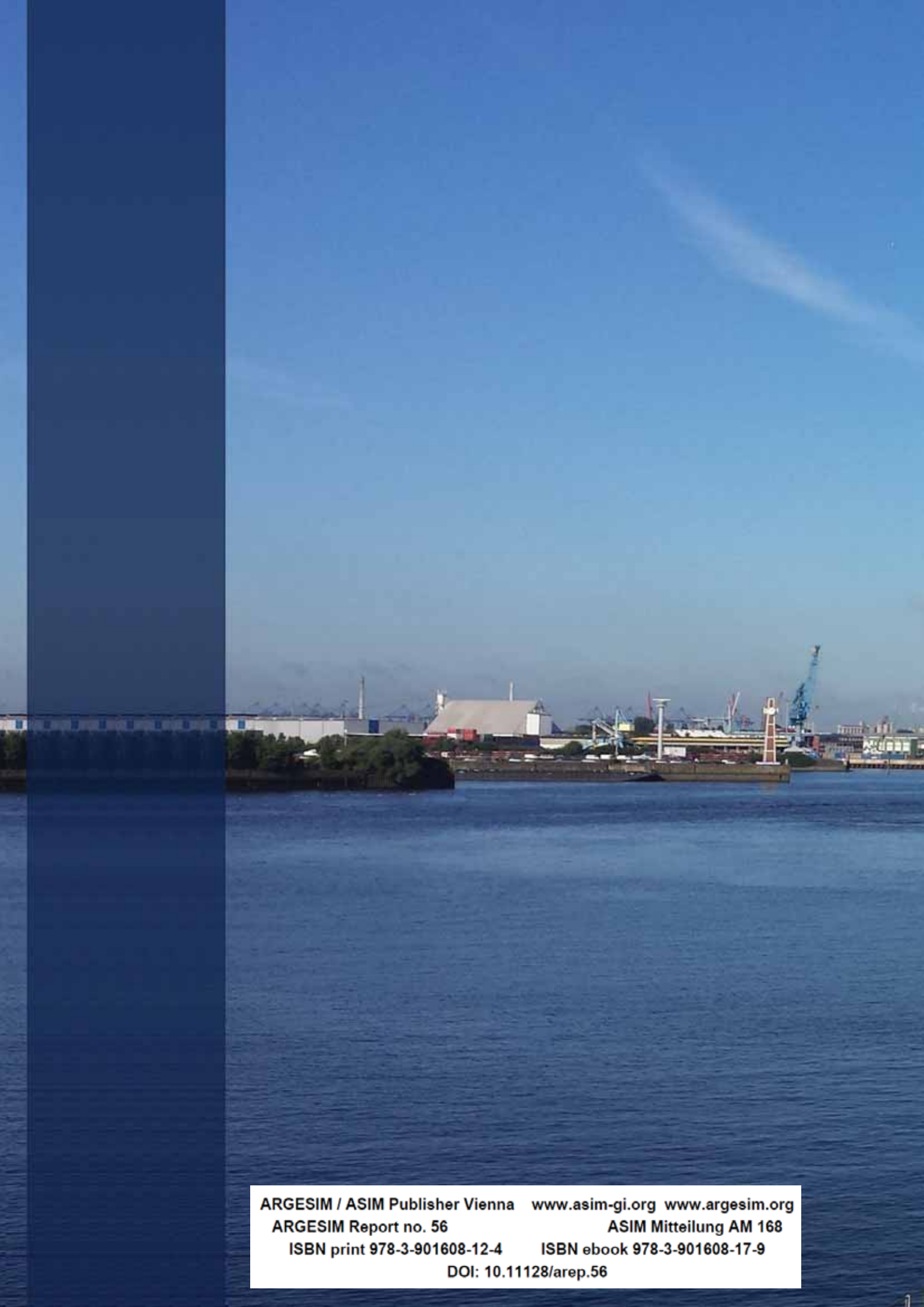
ARGESIM is a non-profit scientific society generally aiming for dissemination of information on system simulation - from research via development to applications of system simulation. ARGESIM's primary publication is the journal SNE – Simulation Notes Europe with open access to all contributions; generally, the authors retain the copyright of their SNE contributions. This copyright regulation holds also for *ARGESIM Reports* and *ARGESIM Advances in Simulation* publishing conference volumes for ASIM, MTHMOD, and EUROSIM (in consideration of copyright regulations for related conference publications) and monographs on system simulation (mainly PhD theses).

About ARGESIM

ARGESIM is a non-profit society 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). *ARGESIM Publisher* organizes publishing activities, with ISBN root 978-3-901608-xx-y and DOI root 10.11128/xx...x.

ARGESIM's activities are:

- 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 → www.argesim.org/benchmarks/
- Publication of the series ARGESIM Reports (for monographs in system simulation, and proceedings of simulation conferences and workshops) → www.argesim.org/publications/
- Publication of the special series FBS Simulation - Advances in Simulation / Fortschrittsberichte Simulation (monographs in co-operation with ASIM, the German Simulation Society)
- Organisation of the Conference Series MATHMOD Vienna (triennial, in co-operation with EUROSIM, ASIM, and TU Wien) → www.mathmod.at
- Organisation of Summerschools and Seminars on Mod&Sim
- Administration and support of ASIM (German Simulation Society → www.asim-gi.org) and of EUROSIM (Federation of European Simulation Societies → www.eurosim.info)



ARGESIM / ASIM Publisher Vienna www.asim-gi.org www.argesim.org
ARGESIM Report no. 56 ASIM Mitteilung AM 168
ISBN print 978-3-901608-12-4 ISBN ebook 978-3-901608-17-9
DOI: 10.11128/arep.56